Lecture Notes in Computer Science 4292

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

George Bebis Richard Boyle
Bahram Parvin Darko Koracin
Paolo Remagnino Ara Nefian
Gopi Meenakshisundaram Valerio Pascucci
Jiri Zara Jose Molineros
Holger Theisel Thomas Malzbender (Eds.)

Advances in Visual Computing

Second International Symposium, ISVC 2006
Lake Tahoe, NV, USA, November 6-8, 2006
Proceedings, Part II

Springer

Volume Editors

George Bebis
University of Nevada, Reno, USA, E-mail: bebis@cse.unr.edu

Richard Boyle
NASA Ames Research Center, CA, USA, E-mail: Richard.Boyle@nasa.gov

Bahram Parvin
Lawrence Berkeley National Laboratory, CA, USA, E-mail: parvin@hpcrd.lbl.gov

Darko Koracin
Desert Research Institute, Reno, NV, USA, E-mail: darko@dri.edu

Paolo Remagnino
DIRC, Kingston University, UK, E-mail: P.Remagnino@kingston.ac.uk

Ara Nefian
Intel, Santa Clara, CA, USA, E-mail: ara.nefian@intel.com

Gopi Meenakshisundaram
University of California at Irvine, CA, USA, E-mail: gopi@ics.uci.edu

Valerio Pascucci
Lawerence Livermore National Laboratory, USA, E-mail: pascucci1@llnl.gov

Jiri Zara
Czech Technical University in Prague, E-mail: zara@fel.cvut.cz

Jose Molineros
Rockwell Scientific, CA, USA, E-mail: jmolineros@rwsc.com

Holger Theisel
Max-Planck Institut für Informatik, Germany, E-mail: theisel@mpi-sb.mpg.de

Thomas Malzbender
Hewlett Packard Labs, Palo Alto, CA, USA, E-mail: malzbend@hpl.hp.com

Library of Congress Control Number: 2006935880

CR Subject Classification (1998): I.4, I.5, I.2.10, I.3.5, I.2.6, F.2.2

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-540-48626-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-48626-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11919629 06/3142 5 4 3 2 1 0

Preface

It is with great pleasure that we welcome you all to the proceedings of the 2nd International Symposium on Visual Computing (ISVC 2006) held in Lake Tahoe. Following a successful meeting last year, we witnessed a much stronger and more productive event this year. ISVC offers a common umbrella for the four main areas of visual computing including vision, graphics, visualization, and virtual reality. Its goal is to provide a forum for researchers, scientists, engineers and practitioners throughout the world to present their latest research findings, ideas, developments and applications in the broader area of visual computing.

This year, the program consisted of 13 oral sessions, one poster session, ten special tracks, and six keynote presentations. The response to the call for papers was very strong. We received more than twice the papers received last year. Specifically, we received over 280 submissions for the main symposium from which we accepted 65 papers for oral presentation (23% acceptance) and 56 papers for poster presentation (20% acceptance). Special track papers were solicited separately through the Organizing and Program Committees of each track. A total of 57 papers were accepted for presentation in the special tracks.

All papers were reviewed with an emphasis on potential to contribute to the state of the art in the field. Selection criteria included accuracy and originality of ideas, clarity and significance of results, and presentation quality. The review process was quite rigorous, involving two to three independent blind reviews followed by several days of discussion. During the discussion period we tried to correct anomalies and errors that might have existed in the initial reviews. Despite our efforts, we recognize that some papers worthy of inclusion may have not been included in the program. We offer our sincere apologies to authors whose contributions might have been overlooked.

We wish to thank everybody who submitted their work to ISVC 2006 for review. It was because of their contributions that we succeeded in having a technical program of high scientific quality. In particular, we would like to thank the ISVC 2006 area Chairs, the organizing institutions (UNR, DRI, LBNL, and NASA Ames), our industrial sponsors (Intel, DigitalPersona, Equinox, Ford, Siemens, Hewlett Packard, NVIDIA, MERL, UtopiaCompression), the international Program Committee, the special track organizers and their Program Committees, the keynote speakers, the reviewers, and especially the authors that contributed their work to the symposium. In particular, we would like to thank Siemens who kindly offered the best paper award this year.

We sincerely hope that the proceedings of ISVC 2006 will offer opportunities for professional growth.

August 2006 ISVC 2006 Steering Committee and Area Chairs

Organization

ISVC 2006 Steering Committee

George Bebis, University of Nevada, Reno, USA
Richard Boyle, NASA Ames Research Center, USA
Bahram Parvin, Lawrence Berkeley National Laboratory, USA
Darko Koracin, Desert Research Institute, USA

ISVC 2006 Area Chairs

Computer Vision
Ara Nefian, Intel, USA
Paolo Remagnino, DIRC, Kingston University London, UK

Computer Graphics
Gopi Meenakshisundaram, University of California-Irvine, USA
Valerio Pascucci, Lawrence Livermore National Laboratory, USA

Virtual Reality
Jiri Zara, Czech Technical University in Prague, Czech Republic
Jose Molineros, Rockwell Scientific, USA

Visualization
Holger Theisel, Max-Planck-Institut für Informatik, Germany
Tom Malzbender, Hewlett Packard Labs, USA

Publicity/Website
Ali Erol, eTreppid Technologies, USA

Local Arrangements
Kostas Veropoulos, Desert Research Institute, USA

Publications
Junxian Wang, UtopiaCompression, USA

ISVC 2006 Keynote Speakers

Carolina Cruz-Neira, University of Louisiana at Lafayette, USA
Eli Peli The Schepens, Harvard Medical School, USA
Daniel DeMenthon, National Science Foundation, USA
Chris Johnson, University of Utah, USA

Dr. Karel Zuiderveld, Vital Images, USA
Mark Nixon, University of Southampton, UK

ISVC 2006 International Program Committee

(Area 1) Computer Vision

J. K. Aggarwal , University of Texas, Austin, USA
Ioannis Pavlidis, University of Houston, USA
Mubarak Shah, University of Central Florida, USA
George Bebis, University of Nevada, Reno, USA
Hammoud, Delphi Corporation, USA
Salil Prabhakar, DigitalPersona Inc., USA
GianLuca Foresti, University of Udine, Italy
Andrea Salgian, The College of New Jersey, USA
Carlo Regazzoni, University of Genoa, Italy
Tieniu Tan, Chinese Academy of Sciences, China
Mircea Nicolescu, University of Nevada, Reno, USA
Stefanos Kollias, National Technical University of Athens, Greece
Bogdan Georgescu, Siemens, USA
James Davis, Ohio State University, USA
Davide Maltoni, University of Bologna, Italy
Alessandro Verri, University of Genova, Italy
Eam Khwang Teoh, Nanyang Technological University, Singapore
Sergio Velastin, Kingston University London, UK
Nikos Paragios, Ecole Centrale de Paris, France
Nikolaos Bourbakis, ITRI Wright State University, USA
Antonis Argyros, University of Crete , Greece
Rahul Singh, San Francisco State University, USA
Zehang Sun, eTreppid Technologies, USA
Bahram Parvin, Lawrence Berkeley National Laboratory, USA
Alexei Skourikhine, Los Alamos National Lab, USA
Theodoros Katsaounis, University of Crete, Greece
Anders Heyden, Lund University, Sweden
Yoshinori Kuno, Saitama University, Japan
Gang Qian, Arizona State University, USA
Vijayan Asari, Old Dominion University, USA
Kyungnam Kim, IPIX, USA
How Lung Eng, Institute for Infocomm Research, Singapore
George Kamberov, Stevens Institute of Technology, USA
Guoliang Fan, Oklahoma State University, USA
Andrea Cavallaro, Queen Mary, University of London, UK
Larry Davis, University of Maryland, USA
Yunqian Ma, Honyewell Labs, USA
Gerald Schaefer, Nottingham Trent University, UK
Goh Wooi Boon, Nanyang Technological University, Singapore

Wei-Yun Yau, Institute for Infocomm Research, Singapore
Jochen Triesch, University of California-San Diego, USA
Michael Webster, University of Nevada, Reno, USA
Jeff Mulligan, NASA Ames Research Center, USA
Stefano Tubaro, DEI, Politecnico di Milano, Italy
Augusto Sarti, DEI, Politecnico di Milano, Italy
James Ferryman, Reading University, UK
Murat Kunt, EPFL, Switzerland
Justus Piater, Université de Liège, Belgium
Ioannis Pitas, Aristotle University of Thessaloniki, Greece
Larry Wolff, Equinox Corporation, USA
Fatih Porikli, MERL, USA
Euripides Petrakis, Technical University of Crete, Greece
Barbara Lynn O'Kane, US Army Night Vision Lab, USA
Besma Abidi, University of Tennessee, USA
Alberto Broggi, Università di Parma, Italy
Gerard Medioni, University of Southern California, USA
Peggy Agouris, University of Maine, USA
Rama Chellappa, University of Maryland, USA
Bob Fisher, University of Edinburgh, UK
Song Wang, University of South Carolina, USA
Peter Sturm, INRIA Rhône-Alpes, France
Mark Nixon, University of Southampton, UK
Ioannis Kakadiaris, University of Houston, USA
David Nister, University of Kentucky, USA
Majid Mirmehdi, Bristol University, UK
Hammadi Nait-Charif, Bournemouth University, UK
Steve Maybank, Birkbeck College, UK
Seong-Whan Lee, Korea University, Korea
Gerda Kamberova, Hofstra University, USA
Aly A. Farag, University of Louisville, USA
Dimitris Samaras, Stony Brook University, USA
Ahmed El-Gammal, University of New Jersey, USA
Christian Debrunner, Colorado School of Mines, USA
Ping Peng, Tulane University, USA
Mohammed Yeasin, University of Memphis, USA
Reinhard Klette, Auckland University, New Zeland
Kokichi Sugihara, University of Tokyo, Japan
Yunhong Wang, Chinese Academy of Sciences, China
Anders Heyden, Malmö University, Sweden
Kenneth Wong, University of Hong Kong, Hong Kong
Kenneth Tobin, Oak Ridge National Laboratory, USA
George Anagnostopoulos, Florida Institute of Technology, USA
Tanveer Syeda-Mahmood, IBM Almaden, USA
David Thirde, Reading University, UK

George Papadourakis, Technological Education Institute, Greece
Sylvain Peyronnet, LRDE/EPITA, France
Alice O'Toole, University of Texas-Dallas, USA
Chandrika Kamath, Lawrence Livermore National Lab, USA
Gabriel Tsechpenakis, Rutgers University, USA
Tony Xiang, Queen Mary, University of London, UK
Stan Birchfield, Clemson University, USA
Ron Miller, Ford Motor Company, USA
Anthony Maeder, CSIRO ICT Centre, Australia
George Kartsounis, Agricultural University of Athens, Greece
Xiangjian He, University of Technology, Australia
Klimis Ntalianis, National Technical University of Athens, Greece
Chunrong Yuan, Fraunhofer Inst. for Applied Info Tech., Germany
Wenjing Li, STI Medical Systems, USA

(Area 2) Computer Graphics

John Dingliana, Trinity College, Ireland
Hanspeter Bieri, University of Bern, Switzerland
Anders Kugler, NVIDIA, USA
Cesar Mendoza, Universidad Rey Juan Carlos, Spain
Li-Yi Wei, Stanford University, USA
Chung-Yen Su, National Taiwan Normal University, Taiwan
Georg Umlauf, University of Kaiserslautern, Germany
Paolo Cignoni, ISTI - CNR, Italy
Gladimir Baranoski, University of Waterloo, Canada
Hammadi Nait-Charif, University of Dundee, Scotland
Tao Ju, Washington University in St. Louis, USA
Lijun Yin, Binghamton University, USA
Valentin Brimkov, State University of New York, USA
Tom Malzbender, Hewlett Packard Labs, USA
Dimitris Samaras, Stony Brook University, USA
Ioannis Kakadiaris, University of Houston, USA
Ralph Martin, Cardiff University, UK
Shimin Hu, Tsinghua University, China
Alvar Vinacua, Universitat Politècnica de Catalunya, Spain
Jian Huang, University of Tennessee, USA
Hyeong-Seok Ko, Seoul National University, Korea
Jorg Peters, University of Florida, USA
James Klosowski, IBM T.J. Watson Research Center, USA
Lakhmi Jain, University of South Australia, Australia
Manuel Oliveira, Univ. Fed. do Rio Grande do Sul, Brazil
Jorn Loviscach, University of Applied Sciences, Bremen, Germany
Miguel Otaduy, ETH-Zurich, Switzerland
Nicholas Bilalis, Technical University of Crete, Greece
Reneta Barneva, State University of New York, USA

Philippe Palanque, University of Paul Sabatier, France
David Ebert, Purdue University, USA
Ik Soo Lim, University of Wales, UK
Ross Brown, Queensland University of Technology, Australia
Alexander Belyaev, Max-Planck-Institut für Informatik, Germany
Alexei Sourin, Nanyang Technological University, Singapore
Ming Wan, Boeing, USA
Irene Cheng, University of Alberta, Canada
Min-Hyung Choi, University of Colorado at Denver, USA
Jim Cremer, University of Iowa, USA
Andre Hinkenjan, Bonn-Rhein-Sieg University of Applied Sciences, Germany
Han-Wei Shen, Ohio State University, USA
Holly Rushmeier, Yale University, USA
Issei Fujishiro, Tohoku University, Japan
John C Hart, University of Illinois at Urbana-Champaign, USA
Kelly Gaither, University of Texas at Austin, USA
Leila De Floriani, University of Maryland, USA
Rachael Brady, Duke University, USA
Raghu Machiraju, Ohio State University, USA
Arik Shamir, The Interdisciplinary Center, Herzliya, Israel
Claudio Silva, University of Utah, USA
Jim Ahrens, Lawrence Livermore National Laboratory, USA
Ken Joy, University of California, Davis USA
Renato Pajarola, University of Zurich, Switzerland

(Area 3) Virtual Reality

Anders Heyden, Lund University, Sweden
Alvar Vinacua, Universitat Politècnica de Catalunya, Spain
Miguel Otaduy, ETH-Zurich, Switzerland
Fred Harris, University of Nevada, Reno, USA
Nicholas Bilalis, Technical University of Crete, Greece
Alexei Sourin, Nanyang Technological University, Singapore
Ming Wan, Boeing, USA
Irene Cheng, University of Alberta, Canada
Richard Boyle, NASA Ames Research Center, USA
Cesar Mendoza, Universidad Rey Juan Carlos, Spain
Reinhold Behringer, Leeds Metropolitan University UK
Jos Remo Ferreira Brega, UNIVEM, PPGCC, Brazil
Hans Hagen, University of Kaiserslautern, Germany
Robert van Liere, CWI, Netherlands
Min-Hyung Choi, University of Colorado at Denver, USA
Cagatay Basdogan, Koç University, Turkey
Jim Cremer, University of Iowa, USA
Joe LaViola, Brown University, USA
Simon Richir, University of Angers, France

Manohar Srikanth, Indian Institute of Science, India
Nickolas Sapidis, Aegean University, Greece
Nigel John, University of Wales Bangor, UK
Ildeberto Rodello, UNIVEM, PPGCC, Brazil
Alan Craig, NCSA University of Illinois at Urbana-Champaign, USA
George Kartsounis, Agricultural University of Athens, Greece
Andre Hinkenjan, Bonn-Rhein-Sieg University of Applied Sciences, Germany
Joerg Meyer, University of California Irvine, USA
Roberto Ranon, University of Udine, Italy
Thomas Varsamidis, University of Wales, UK
Sabine Coquillart, INRIA, France
Greg Schmidt, Naval Research Laboratory, USA
Chunrong Yuan, Fraunhofer Inst. for Applied Info Tech., Germany

(Area 4) Visualization

J. Edward Swan II, The Naval Research Laboratory, USA
James Klosowski, IBM T.J. Watson Research Center, USA
Paolo Cignoni, ISTI - CNR, Italy
Nicholas Bilalis, Technical University of Crete, Greece
Darko Koracin, Desert Research Institute, USA
Fred Harris, University of Nevada, Reno, USA
Olaf Thiele, University of Mannheim, Germany
Robert Rabin, University of Wisconsin, Madison, USA
David Ebert, Purdue University, USA
Helwig Hauser, VRVis Research Center, Austria
Robert Moorhead, Mississippi State University, USA
Klaus Mueller, SUNY Stony Brook, USA
Theresa-Marie Rhyne, North Carolina State University, USA
Mark Apperley, University of Waikato, New Zealand
Alfred Inselberg, Tel Aviv University, Israel
Nabil Adam, Rutgers University, USA
Brian Wylie, Sandia National Laboratory, USA
Alexei Sourin, Nanyang Technological University, Singapore
Mao Lin Huang, University of Technology, Australia
Anthony Maeder, CSIRO ICT Centre, Australia
Jos Roerdink, University of Groningen, Netherlands
Jose Malpica, Alcala University, Spain
Yoshitaka Masutani, The University of Tokyo Hospital, Japan
Pavel Slavik, Czech Technical University in Prague, Czech Republic
Kwan-Liu Ma, University of California-Davis, USA
Ming Wan, Boeing, USA
Irene Cheng, University of Alberta, Canada
Jack Snoeyink, University of North Carolina, USA
Heidrun Schumann, Rostock University, Germany

Ross Brown, Queensland University of Technology, Australia
Robert van Liere, CWI, Netherlands

ISVC 2006 Special Tracks

1. Intelligent Environments: Algorithms and Applications

Organizers
Paolo Remagnino, DIRC, Kingston University, UK
How-Lung Eng, IIR, Singapore
Guoliang Fan, Oklahoma State University, USA
Yunqian Ma, Honeywell Labs, USA
Monique Thonnat, INRIA, France

2. Multimodal Data Understanding and Visualization for Industrial Applications

Organizers
Fatih Porikli, MERL, USA
Andrea Cavallaro, Queen Mary, University of London, UK

Program Committee
Rama Chellapa, University of Maryland, USA
Yuri Ivanov, MERL, USA
Swarup Medasani, HRL, USA
Ron Miller, Ford Motor Company, USA
Chris Wren, MERL, USA

3. Pattern Analysis and Recognition Applications in Biometrics

Organizers
Ali Erol, University of Nevada, Reno, USA
Salil Prabhakar, DigitalPersona, USA
Mark Nixon, University of Southampton, UK
Arun Abraham Ross, West Virginia University, USA

4. Biomedical Image Analysis

Organizers
Tao Ju, Washington University, USA
Ioannis Kakadiaris, University of Houston, USA
Shi Pengcheng, Hong Kong University of Science and Technology, China
Tomas Gustavsson, Chalmers University of Technology, Sweden

5. Understanding and Imitating Nature: Analysis, Interpretation, Rendering and Inspiration of Biological Forms

Organizers
Paolo Remagnino, DIRC, Kingston University, UK
Richard Boyle, NASA Ames, USA
Paul Wilkin, The Royal Botanic Gardens, UK
Jonathan Clark, University of Surrey, UK
Sarah Barman, Kingston University, UK

6. Visual Computing and Biological Vision

Organizers
Jeff Mulligan, NASA Ames, USA
Michael Webster, University of Nevada, Reno, USA
Alice O'Toole, University of Texas at Dallas, USA

7. 4D Medical Data Modeling, Visualization and Measurement

Organizers
Irene Cheng, University of Alberta, Canada
Randy Goebel, University of Alberta, Canada
Lijun Yin, State University of New York, USA

Program Committee
Walter Bischof, University of Alberta, Canada
Pierre Boulanger, University of Alberta, Canada
Paul Major, University of Alberta, Canada
Jana Rieger, Misericordia Community Hospital, Canada
Brian Maraj, University of Alberta, Canada
Carol Boliek, University of Alberta, Canada

8. Discrete and Computational Geometry and Their Applications in Visual Computing

Organizers
Valentin Brimkov, State University of New York, USA
Reneta Barneva, State University of New York, USA

Program Committee
Eric Andres, Université de Poitiers, France
David Coeurjolly, Université Claude Bernand Lyon, France
Isabelle Debled-Rennesson, IUFM de Lorraine, France
Guillaume Damiand, Université de Poitiers, France
Christophe Fiorio, Ecole Polytechnique Universitaire de Montpellier, France
Atushi Imiya, Chyba University, Japan
Reinhard Klette, Auckland University, New Zealand

9. Soft Computing in Image Processing and Computer Vision

Organizers
Gerald Schaefer, Nottingham Trent University, UK
Muhammad Sarfraz, King Fahd University of Petroleum and Minerals, Saudi Arabia
Lars Nolle, Nottingham Trent University, UK

10. Energy Minimization Approaches in Image Processing and Computer Vision

Organizers
Jose M. Bioucas-Dias, Instituto Superior Tecnico Torre Norte, Portugal
Antonin Chambolle, CMAP Ecole Polytechnique, France
Jerome Darbon, EPITA Research and Development Laboratory, France

Additional Reviewers

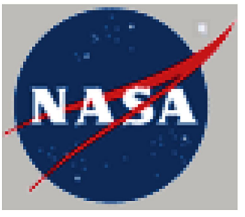
Steve Callahan
Emanuele Santos
John Schreiner
Louis Bavoil
Linh Ha
Huy T. Vo
Erik Anderson
Raphael Brger
Oliver Wang
Max Louwerse
Mike Harville
Bruce Culbertson
Harlyn Baker
Alireza Tavakkoli
Leandro Loss
Gholamreza Amayeh
Kostas Veropoulos
Junxian Wang
Ali Erol

Organizing Institutions and Sponsors





















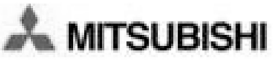




Table of Contents – Part II

Multiple Description Coding for Robust Video Transmission over Wireless Ad-Hoc Networks 1
Joohee Kim

Emotion-Based Textile Indexing Using Colors, Texture and Patterns . . . 9
Soo-jeong Kim, Eun Yi Kim, Karpjoo Jeong, Jee-in Kim

Affine Camera for 3-D Retinal Surface Reconstruction 19
Thitiporn Chanwimaluang, Guoliang Fan

The Diagnostic Application of Brain Image Processing and Analysis System for Ischemic Stroke 31
Tzyh-Chyang Chang, Jiann-Der Lee, Chung-Hsien Huang, Tony Wu, Chi-Jen Chen, Shwu-Jiuan Wu

Development of Early Tunnel Fire Detection Algorithm Using the Image Processing 39
Dongil Han, Byoungmoo Lee

Simulation of Artificial Winds Using a Hardware Illumination Technique 49
Namkyung Lee, Nakhoon Baek, Kwan Woo Ryu

Learning for Multi-view 3D Tracking in the Context of Particle Filters 59
Juergen Gall, Bodo Rosenhahn, Thomas Brox, Hans-Peter Seidel

Improving Brightness for a Multi-projector Display Considering Image Content 70
Hee-Won Lee, Byung-Uk Lee

VirtualQWERTY: Textual Communication in Virtual Reality 79
Jaewoo Ahn, Kyungha Min

Networked Heterogeneous Camera System for High Resolution Face Images 88
Sofiane Yous, Abdelaziz Khiat, Masatsugu Kidode, Tsukasa Ogasawara

A New Method for Approximating Optimal Parameterization of Polynomial Curves 98
Fenghua Guo, Caiming Zhang

Interpolation by Piecewise Quadric Polynomial to Scattered Data Points 106
Shanshan Gao, Caiming Zhang, Li Zhong

Detection and Localization of the Top Object in the Stack of Objects 116
Hernsoo Hahn, Youngjoon Han

Recognition of 3D Object Using Attributed Relation Graph of Silhouette's Extended Convex Hull 126
Hernsoo Hahn, Youngjoon Han

Image Retrieval by Local Contrast Patterns and Color 136
M.K. Bashar, N. Ohnishi

A VR Game Platform Built Upon Wireless Sensor Network 146
Doo-seop Eom, Jungshik Jang, Taeyoung Kim, JungHyun Han

Feature Extraction and Selection for Recognizing Humans by Their Gait 156
Jang-Hee Yoo, Mark S. Nixon

Rectification of Illumination in Images Used for Shape from Focus 166
Mannan S.M., Aamir Saeed Malik, Humaira Nisar, Tae-Sun Choi

Bilateral Edge Detection on a Virtual Hexagonal Structure 176
Xiangjian He, Wenjing Jia, Namho Hur, Qiang Wu, Jinwoong Kim, Tom Hintz

Issues and Implementation of C^1 and C^2 Natural Neighbor Interpolation 186
T. Bobach, M. Bertram, G. Umlauf

Iris Recognition Using a Low Level of Details 196
Jaemin Kim, Seongwon Cho, Daewhan Kim, Sun-Tae Chung

Dynamic Reconstruction of Complex Planar Objects on Irregular Isothetic Grids 205
Antoine Vacavant, David Coeurjolly, Laure Tougne

Layout of Multiple Views for Volume Visualization: A User Study 215
Daniel Lewis, Steve Haroz, Kwan-Liu Ma

Video Indexing and Retrieval in Compressed Domain Using Fuzzy-Categorization 227
Hui Fang, Rami Qahwaji, Jianmin Jiang

Computing Homology for Surfaces with Generalized Maps: Application to 3D Images 235
Guillaume Damiand, Samuel Peltier, Laurent Fuchs

Dynamic Texture Analysis and Synthesis Using Tensor Decomposition 245
Roberto Costantini, Luciano Sbaiz, Sabine Süsstrunk

Color Pair Clustering for Texture Detection 255
Lech Szumilas, Allan Hanbury

CPU-GPU Multithreaded Programming Model: Application to the Path Tracing with Next Event Estimation Algorithm 265
Christophe Cassagnabère, François Rousselle, Christophe Renaud

Real-Time and Robust Monocular SLAM Using Predictive Multi-resolution Descriptors 276
Denis Chekhlov, Mark Pupilli, Walterio Mayol-Cuevas, Andrew Calway

A Faster Graph-Based Segmentation Algorithm with Statistical Region Merge 286
Ahmed Fahad, Tim Morris

Sensor Fusion Based Obstacle Detection/Classification for Active Pedestrian Protection System 294
Ho Gi Jung, Yun Hee Lee, Pal Joo Yoon, In Yong Hwang, Jaihie Kim

Combinatorial Pyramids and Discrete Geometry for Energy-Minimizing Segmentation 306
Martin Braure de Calignon, Luc Brun, Jacques-Olivier Lachaud

Fast Dense Stereo Matching Using Adaptive Window in Hierarchical Framework 316
SangUn Yoon, Dongbo Min, Kwanghoon Sohn

A New Photographing Apparatus for Skin Maps of Human Face Rendering 326
Haedong Kim, Howook Jang, Inho Lee

Investigating the Dynamics of Facial Expression 334
Jane Reilly, John Ghent, John McDonald

GLOBAL Topology Preservation in Isosurface Extraction of Volumetric Data 344
Xingqiang Yang, Caiming Zhang

Real-Time Model-Based SLAM Using Line Segments 354
Andrew P. Gee, Walterio Mayol-Cuevas

Feature Correspondences from Multiple Views of Coplanar Ellipses 364
C. Barat, J.F. Menudet, H. Louhichi, T. Fournel

Evaluation of Subpixel Tracking Algorithms 374
Johan Skoglund, Michael Felsberg

Adaptive Real-Time Rendering for Large-Scale Molecular Models 383
Jun Lee, Sungjun Park, Jee-In Kim

An Efficient Algorithm for Connected Attribute Thinnings and Thickenings 393
David Lesage, Jérôme Darbon, Ceyhun Burak Akgül

Creating Multi-layered 3D Images Using Reversible Jump MCMC Algorithms 405
Sergio Hernandez-Marin, Andrew M. Wallace, Gavin J. Gibson

A Multi-Modal Interface for Road Planning Tasks Using Vision, Haptics and Sound 417
Matt Newcomb, Chris Harding

Venation Pattern Analysis of Leaf Images 427
James Clarke, Sarah Barman, Paolo Remagnino, Ken Bailey, Don Kirkup, Simon Mayo, Paul Wilkin

A Mobile Low-Cost Motion Capture System Based on Accelerometers 437
Jan-Phillip Tiesel, Jörn Loviscach

Fusing Individual Algorithms and Humans Improves Face Recognition Accuracy 447
Alice J. O'Toole, Fang Jiang, Hervé Abdi, P. Jonathon Phillips

A Method for the Automatic Analysis of Colour Category Pixel Shifts During Dichromatic Vision 457
Mike Bennett, Aaron Quigley

Rendering Dynamic Real-World Scenes Using Image Spheres 467
Stephan Behrendt

Arithmetic Discrete Parabolas 480
I. Debled-Rennesson, E. Domenjoud, D. Jamet

Retinal Spot Lesion Detection Using Adaptive Multiscale Morphological Processing 490
Xin Zhang, Guoliang Fan

Iterative Closest SIFT Formulation for Robust Feature Matching 502
Rafael Lemuz-López, Miguel Arias-Estrada

Invertible Polygonalization of 3D Planar Digital Curves and Application to Volume Data Reconstruction 514
Martine Dexet, David Cœurjolly, Eric Andres

Lateral and Depth Calibration of PMD-Distance Sensors 524
Marvin Lindner, Andreas Kolb

Autonomous Vehicle Video Aided Navigation – Coupling INS and Video Approaches 534
Chris Baker, Chris Debrunner, Sean Gooding, William Hoff, William Severson

A Method of Improving Cloud Predictions for Real-Time Weather Forecasting and Visualization 544
Ramesh Vellore, Darko Koračin, Melanie Wetzel

An Efficient Hardware Architecture for Full-Search Variable Block Size Motion Estimation in H.264/AVC 554
Seung-Man Pyen, Kyeong-Yuk Min, Jong-Wha Chong, Satoshi Goto

A Domain Reduction Algorithm for Incremental Projective Reconstruction 564
Rafael Lemuz-López, Miguel Arias-Estrada

An Automated Procedure for Word Balloon Placement in Cinema Comics 576
Bong-Kyung Chun, Dong-Sung Ryu, Won-Il Hwang, Hwan-Gue Cho

Segmentation of Three Dimensional Cell Culture Models from a Single Focal Plane 586
Hang Chang, Bahram Parvin

A Semi-automatic 3D Reconstruction Algorithm for Telepresence 596
Michel Sarkis, Klaus Diepold

Motion Detection Using an Improved Colour Model 607
Horst Wildenauer, Philipp Blauensteiner, Allan Hanbury, Martin Kampel

Combining Pixelization and Dimensional Stacking 617
John T. Langton, Astrid A. Prinz, Timothy J. Hickey

Detection and Characterization of Abnormal Vascular Patterns in Automated Cervical Image Analysis 627
Wenjing Li, Allen Poirson

Towards a Modular Network-Distributed Mixed-Reality Learning Space System 637
Timothy J. Rogers, Bedřich Beneš, Gary R. Bertoline

Real-Time Multi-view 3D Object Tracking in Cluttered Scenes 647
Huan Jin, Gang Qian, Stjepan Rajko

Visualizing Natural Stereo Images in Short Distance: A New Approach 657
Antonia Lucinelma Pessoa Albuquerque, Sergio Pinheiro, Rubens Melo

Next Best View Algorithms for Interior and Exterior Model Acquisition 668
Bradley D. Null, Eric D. Sinzinger

An Experiential Approach to Interacting with Biological Information 678
Naureen Moon, Bibek Dev Bhattarai, Rahul Singh

Convex Shapes and Convergence Speed of Discrete Tangent Estimators 688
Jacques-Olivier Lachaud, François de Vieilleville

Towards Obtaining an Ideal Real Time Panoramic Video 698
Harsh Dhand, Lakshmi Pavan Daggubati

Object Recognition Using Local Descriptors: A Comparison 709
Andrea Salgian

Improving Spatiotemporal Inpainting with Layer Appearance Models 718
Thommen Korah, Christopher Rasmussen

Edge Detection Using a Complex-Valued Directional Vector Representation 731
Sung Bae Kim, Rae-Hong Park

Analysis and Design of Graphical Password Techniques 741
Xiaoyuan Suo, Ying Zhu, G. Scott Owen

Viewing Scenes Occluded by Smoke 750
Arturo Donate, Eraldo Ribeiro

A Non-photorealistic Rendering of Seurat's Pointillism 760
Hui-Lin Yang, Chuan-Kai Yang

Semantically Relevant Image Retrieval by Combining Image and Linguistic Analysis 770
Tony Lam, Rahul Singh

A New Algorithm for Solid Texture Synthesis 780
Jia-Wei Chiou, Chuan-Kai Yang

What Can We Learn from Biological Vision Studies for Human Motion Segmentation? 790
Cheng Chen, Guoliang Fan

3D Geometry from Uncalibrated Images 802
George Kamberov, Gerda Kamberova, O. Chum, Š. Obdržálek, D. Martinec, J. Kostková, T. Pajdla, J. Matas, R. Šára

Hierarchical Image Database Navigation on a Hue Sphere 814
Gerald Schaefer, Simon Ruszala

Image-Based Informatics for Preclinical Biomedical Research 824
Kenneth W. Tobin, Deniz Aykac, V. Priya Govindasamy, Shaun S. Gleason, Jens Gregor, Thomas P. Karnowski, Jeffery R. Price, Jonathan Wall

Segmentation-Based Registration of Organs in Intraoperative Video Sequences 835
James Goddard, Timothy Gee, Hengliang Wang, Alexander M. Gorbach

On Asymmetric Classifier Training for Detector Cascades 843
Timothy F. Gee

Active Stabilization of Images Acquired on a Walking Robotic Platform 851
Xander Twombly, Richard Boyle, Silvano Colombano

Motion Estimation with Edge Continuity Constraint for Crowd Scene Analysis 861
B. Zhan, P. Remagnino, S.A. Velastin, N. Monekosso, L-Q. Xu

Perceptual Grouping Based on Iterative Multi-scale Tensor Voting 870
Leandro Loss, George Bebis, Mircea Nicolescu, Alexei Skourikhine

Fast Prediction Mode Decision Algorithm for H.264 Based on Hierarchical Mode Classification Framework 882
Cheng-dong Shen, Si-kun Li

Camera Self-calibration in Scheimpflug Condition for Air Flow Investigation 891
Hanene Louhichi, Thierry Fournel, Jean Marc Lavest, Habib Ben Aissia

Author Index 901

Table of Contents – Part I

Activity Recognition Via Classification Constrained Diffusion Maps 1
Yunqian Ma, S.B. Damelin, O. Masoud, N. Papanikolopoulos

Generating and Updating Textures for a Large-Scale Environment 9
Jinhui Hu, Suya You, Ulrich Neumann

Planar Surface Detection in Image Pairs Using Homographic Constraints 19
Qiang He, Chee-hung Henry Chu

Robust Quality-Scalable Transmission of JPEG2000 Images over Wireless Channels Using LDPC Codes 28
Abdullah Al Muhit, Teong Chee Chuah

A Novelty Detection Approach for Foreground Region Detection in Videos with Quasi-stationary Backgrounds 40
Alireza Tavakkoli, Mircea Nicolescu, George Bebis

Procedural Image Processing for Visualization 50
Xiaoru Yuan, Baoquan Chen

Tracking of Individuals in Very Long Video Sequences 60
P. Fihl, R. Corlin, S. Park, T.B. Moeslund, M.M. Trivedi

A Natural Interface for Sign Language Mathematics 70
Nicoletta Adamo-Villani, Bedřich Beneš, Matt Brisbin, Bryce Hyland

A Novel Gait Recognition Method Via Fusing Shape and Kinematics Features 80
Yanmei Chai, Qing Wang, Jingping Jia, Rongchun Zhao

Illumination Normalization for Color Face Images 90
Faisal R. Al-Osaimi, Mohammed Bennamoun, Ajmal Mian

Real-Time Detection of Out-of-Plane Objects in Stereo Vision 102
Weiguang Guan, Patricia Monger

Stereo Imaging with Uncalibrated Camera 112
Xiaokun Li, Chiman Kwan, Baoxin Li

Global Hand Pose Estimation by Multiple Camera Ellipse Tracking 122
Jorge Usabiaga, Ali Erol, George Bebis, Richard Boyle, Xander Twombly

Vision-Based Self-localization of Autonomous Guided Vehicle Using Landmarks of Colored Pentagons 133
Y.S. Kim, J.C. Kim, E.J. Park, Joonwhoan Lee

An Automated System for Contact Lens Inspection 141
A.I. Bazin, T. Cole, B. Kett, M.S. Nixon

Efficient Motion Search in Large Motion Capture Databases 151
Yi Lin

Real-Time Rendering of Light Shafts on GPU 161
Shuyi Chen, Sheng Li, Guoping Wang

Learning the Stylistic Similarity Between Human Motions 170
Yu-Ren Chien, Jing-Sin Liu

Effects of Layer Partitioning in Collaborative 3D Visualizations 180
Lars Winkler Pettersson, Andreas Kjellin, Mats Lind, Stefan Seipel

GPU-Based Active Contour Segmentation Using Gradient Vector Flow 191
Zhiyu He, Falko Kuester

Active Single Landmark Based Global Localization of Autonomous Mobile Robots 202
Abdul Bais, Robert Sablatnig, Jason Gu, Stefan Mahlknecht

Iterative Estimation of 3D Transformations for Object Alignment 212
Tao Wang, Anup Basu

Temporal Alignment of Time Varying MRI Datasets for High Resolution Medical Visualization 222
Meghna Singh, Anup Basu, Mrinal Mandal

Physically Interacting with Four Dimensions 232
Hui Zhang, Andrew J. Hanson

Low Level Moving-Feature Extraction Via Heat Flow Analogy 243
Cem Direkoğlu, Mark S. Nixon

Shape Tracking and Registration for 4D Visualization of MRI and Structure 253
Irene Cheng, Sharmin Nilufar, Anup Basu, Randy Goebel

History Trees as Descriptors of Macromolecular Structures 263
Deniz Sarioz, T. Yung Kong, Gabor T. Herman

Fusing Features in Direct Volume Rendered Images 273
Yingcai Wu, Huamin Qu, Hong Zhou, Ming-Yuen Chan

Binocular Uncalibrated Photometric Stereo 283
Hui Kong, Pengfei Xu, Eam Khwang Teoh

Empirical Evaluation of a Visual Interface for Exploring Message Boards 293
Beomjin Kim, Philip Johnson, Jason Baker

Direct Estimation of the Stereo Geometry from Monocular Normal Flows 303
Ding Yuan, Ronald Chung

Singular Value Decomposition-Based Illumination Compensation in Video 313
Ki-Youn Lee, Rae-Hong Park

Facial Expression Transformations for Expression-Invariant Face Recognition 323
Hyung-Soo Lee, Daijin Kim

A High-Speed Parallel Architecture for Stereo Matching 334
Sungchan Park, Hong Jeong

Light Simulation in a Distributed Driving Simulator 343
Stefan Lietsch, Henning Zabel, Martin Eikermann, Veit Wittenberg, Jan Berssenbrügge

Self-adaptive RBF Neural Networks for Face Recognition 353
S. Gharai, S. Thakur, S. Lahiri, J.K. Sing, D.K. Basu, M. Nasipuri, M. Kundu

An Improved Representation of Junctions Through Asymmetric Tensor Diffusion 363
Shawn Arseneau, Jeremy R. Cooperstock

Accurate Extraction of Reciprocal Space Information from Transmission Electron Microscopy Images 373
Edward Rosten, Susan Cox

GPU Accelerated Isosurface Extraction on Tetrahedral Grids 383
Luc Buatois, Guillaume Caumon, Bruno Lévy

Enhancing Information on Large Scenes by Mixing Renderings 393
Vincent Boyer, Dominique Sobczyk

Auto-focusing in Extreme Zoom Surveillance: A System Approach with Application to Faces 401
Yi Yao, Besma Abidi, Michael Tousek, Mongi Abidi

Trifocal Transfer Based Novel View Synthesis for Micromanipulation 411
Julien Bert, Sounkalo Dembélé, Nadine Lefort-Piat

Simulation of Diabetic Retinopathy Neovascularization in Color Digital Fundus Images 421
Xinyu Xu, Baoxin Li, Jose F. Florez, Helen K. Li

Mesh Optimisation Using Edge Information in Feature-Based Surface Reconstruction 434
Jun Liu, Roger Hubbold

Finite Sample Bias of Robust Scale Estimators in Computer Vision Problems 445
Reza Hoseinnezhad, Alireza Bab-Hadiashar, David Suter

Flexible Segmentation and Smoothing of DT-MRI Fields Through a Customizable Structure Tensor 455
Thomas Schultz, Bernhard Burgeth, Joachim Weickert

Using Visualizations to Support Design and Debugging in Virtual Reality 465
Cara Winterbottom, Edwin Blake, James Gain

Strategies for Part-Based Shape Analysis Using Skeletons 475
Wooi-Boon Goh

Automatic Learning of Articulated Skeletons from 3D Marker Trajectories 485
Edilson de Aguiar, Christian Theobalt, Hans-Peter Seidel

Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking 495
Thomas Coogan, George Awad, Junwei Han, Alistair Sutherland

Physically-Based Real-Time Diffraction Using Spherical Harmonics 505
Clifford Lindsay, Emmanuel Agu

3D Segmentation of Mammospheres for Localization Studies 518
Ju Han, Hang Chang, Qing Yang, Mary Helen Barcellos-Hoff, Bahram Parvin

Viewpoint Selection for Angiographic Volume 528
Ming-Yuen Chan, Huamin Qu, Yingcai Wu, Hong Zhou

Recognizing Action Primitives in Complex Actions Using Hidden Markov Models 538
V. Krüger

Polyhedrization of Discrete Convex Volumes 548
Valentin E. Brimkov, Reneta Barneva

Automatic Camera Calibration and Scene Reconstruction with Scale-Invariant Features 558
Jun Liu, Roger Hubbold

Surface Fitting to Curves with Energy Control 569
Wen-Ke Wang, Hui Zhang, Jun-Hai Yong, Jia-Guang Sun

Blob Tracking with Adaptive Feature Selection and Accurate Scale Determination 579
Jingping Jia, David Feng, Yanmei Chai, Rongchun Zhao, Zheru Chi

Self-Calibration with Two Views Using the Scale-Invariant Feature Transform 589
Jae-Ho Yun, Rae-Hong Park

Improved Face Recognition Using Extended Modular Principal Component Analysis 599
Changhan Park, Inho Paek, Joonki Paik

Shape Reconstruction by Line Voting in Discrete Space 608
Kosuke Sato, Atsushi Imiya, Tomoya Sakai

Characterization of the Closest Discrete Approximation of a Line in the 3-Dimensional Space 618
J.-L. Toutant

Margin Maximizing Discriminant Analysis for Multi-shot Based Object Recognition 628
Hui Kong, Eam Khwang Teoh, Pengfei Xu

A Novel 3D Statistical Shape Model for Segmentation of Medical Images 638
Zheen Zhao, Eam Khwang Teoh

Scale Consistent Image Completion 648
Michal Holtzman-Gazit, Irad Yavneh

EXDRAP: An Extended Dead Reckoning Architectural Pattern for the Development of Web-Based DVE Applications 660
Nerssi Nasiri Amini, Mostafa Haghjoo

Optimal Parameterizations of Bézier Surfaces 672
Yi-Jun Yang, Jun-Hai Yong, Hui Zhang, Jean-Claude Paul, Jiaguang Sun

Constrained Delaunay Triangulation Using Delaunay Visibility 682
Yi-Jun Yang, Hui Zhang, Jun-Hai Yong, Wei Zeng, Jean-Claude Paul, Jiaguang Sun

Immersing Tele-operators in Collaborative Augmented Reality 692
Jane Hwang, Namgyu Kim, Gerard J. Kim

GrayCut - Object Segmentation in IR-Images 702
Christian Ruwwe, Udo Zölzer

Unsupervised Clustering of Shapes 712
Mohammad Reza Daliri, Vincent Torre

Markerless Pose Tracking for Augmented Reality 721
Chunrong Yuan

Lip Detection Using Confidence-Based Adaptive Thresholding 731
Jin Young Kim, Seung You Na, Ronald Cole

Optic Flow Integration at Multiple Spatial Frequencies - Neural Mechanism and Algorithm 741
Cornelia Beck, Pierre Bayerl, Heiko Neumann

A Critical Appraisal of the Box Counting Method to Assess the Fractal Dimension of Tree Crowns 751
D. Da Silva, F. Boudon, C. Godin, O. Puech, C. Smith, H. Sinoquet

3D Surface Reconstruction and Registration for Image Guided Medialization Laryngoplasty 761
Ge Jin, Sang-Joon Lee, James K. Hahn, Steven Bielamowicz, Rajat Mittal, Raymond Walsh

Vision-Based User Interfaces for Health Applications: A Survey 771
Alexandra Branzan Albu

Multiple Hypothesis Target Tracking Using Merge and Split of Graph's Nodes 783
Yunqian Ma, Qian Yu, Isaac Cohen

Understanding 3D Emotions Through Compact Anthropometric Autoregressive Models 793
Charlotte Ghys, Nikos Paragios, Bénédicte Bascle

Graph-Based Multi-resolution Temporal-Based Face Reconstruction 803
Charlotte Ghys, Nikos Paragios, Bénédicte Bascle

Web-Based Interface for the Visualization of Microarray Data 813
B. Vanteru, J. Shaik, M. Yeasin

3D and Texture Modelling of Precolombian Objects 822
Jorge Hernández, Flavio Prieto

Segmentation of Triangular Meshes Using Multi-scale Normal Variation 831
Kyungha Min, Moon-Ryul Jung

Integration of Multiple Methods for Class and Specific Object Recognition 841
Al Mansur, Md. Altab Hossain, Yoshinori Kuno

An Efficient Photon Mapping Algorithm for Rendering Light-Emitting Fluids 850
Kyungha Min

Face Recognition Using 2D and 3D Multimodal Local Features 860
Ajmal Mian, Mohammed Bennamoun, Robyn Owens

Adaptive Background Generation for Video Object Segmentation 871
Taekyung Kim, Joonki Paik

Omnidirectional Camera Calibration and 3D Reconstruction by Contour Matching 881
Yongho Hwang, Jaeman Lee, Hyunki Hong

Real-Time GPU-Based Simulation of Dynamic Terrain 891
Anthony S. Aquilio, Jeremy C. Brooks, Ying Zhu, G. Scott Owen

High-Resolution Video from Series of Still Photographs 901
Ge Jin, James K. Hahn

Author Index 911

Multiple Description Coding for Robust Video Transmission over Wireless Ad-Hoc Networks

Joohee Kim*

School of Information and Communication Engineering
Inha University, Incheon, 402-751, Korea
joohee@inha.ac.kr

Abstract. This paper presents a quality-scalable multiple description coding method for the H.264/AVC for error-resilient video transmission in wireless ad-hoc networks. In the proposed system, each video frame is divided into two regions according to the motion activities in the scene and the more important region is encoded into two independently decodable and equally important descriptions. The two descriptions are transmitted over different paths to a receiver in order to alleviate the effect of unstable channel conditions of wireless ad-hoc networks. If the two descriptions are received error-free, then a high quality video is reconstructed. If one description is lost, then the other description can still be decoded to produce a basic video quality, and furthermore, the correctly received description is used to estimate the lost information of the corrupted description. The proposed MD coder maintains an adequate video quality as long as both description are not simultaneously lost. Simulation results show that the proposed MD coding combined with multi-path transport system is largely immune to packet losses, and therefore, can be a promising solution for robust video communications in wireless ad-hoc networks.

1 Introduction

With the development of mobile communication technologies, it is expected that there will be increasing demand for video streaming services over wireless ad hoc networks. Possible applications of video streaming over ad-hoc networks include video conferencing, highway automation, broadband wireless Internet access, and search-and-rescue missions. However, video streaming over wireless ad-hoc networks has many technical challenges. Since ad-hoc networks are deployed instantly in situations where infrastructure is unavailable, the network topology is frequently changing due to node mobility and links are continuously established and broken. Therefore, the availability and quality of a link fluctuates and transmission error is more frequent than that in single-hop wireless networks.

* This works was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2005-204-D00018).

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 1–8, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The H.264/AVC video coding standard [1] provides a high coding efficiency and has a high degree of flexibility for operation in a variety of network conditions because many restrictions concerning the data structure (e.g., picture ordering, slice structure, and macroblock ordering) are removed. Flexible macroblock ordering (FMO), one of the new error-resiliency features provided in H.264/AVC, specifies a pattern that assigns the macroblocks in a picture to one of the slice groups. Each slice group is independently coded and transmitted. Recently, a simple two-layer scalable coder based on H.264/AVC has been proposed that allows H.264/AVC to achieve a quality scalable coding by exploiting the flexible macroblock ordering [2].

Multiple description coding (MDC) has emerged as a promising technique to enhance the error resilience of a video transport system [3]. An MD coder generates two or more independently decodable and equally important descriptions (or bitstreams) so that each description alone provides low but acceptable quality and multiple descriptions lead to higher quality. This MD coding approach is especially effective for delay-constrained video transmission in wireless ad-hoc networks where the error rate is high and the network conditions change frequently due to potential node mobility. The benefits of using MDC in video streaming can be further amplified when MDC is combined with multi-path transport [4,5]. In these approaches, each description is explicitly transmitted over an independent path to a receiver. MDC combined with multi-path transport can achieve higher throughput and increase tolerance to packet loss and delay due to network congestion because the probability of having all the paths simultaneously congested is low. In wireless ad-hoc networks, multiple paths can be supported by practical multi-path routing protocols such as DSR and TORA [6].

In this paper, we propose a scalable multiple description coding method for the H.264/AVC standard. The proposed MD coder divides each video frame into two regions according to the motion activities of the scene and encodes the blocks belonging to the more important region into two independent descriptions. The two descriptions are transmitted over different paths to a receiver in order to alleviate the effect of unstable channel conditions of wireless ad-hoc networks. If the two descriptions are received error-free, then a high quality video is reconstructed. If one description is lost, then the other description can still be decoded to produce a basic video quality, and furthermore, the correctly received description is used to estimate the lost information of the corrupted description.

The rest of the paper is organized as follows. In Section 2, we introduce the proposed multiple description video coding algorithm combined with multi-path transport and decoder-side error concealment. The simulation results are given in Section 3 and Section 4 concludes the paper.

2 Proposed Multiple Description Video Coding

We propose a multiple description video coding method that extends the quality-scalable H.264/AVC video coding algorithm proposed in [2]. The proposed

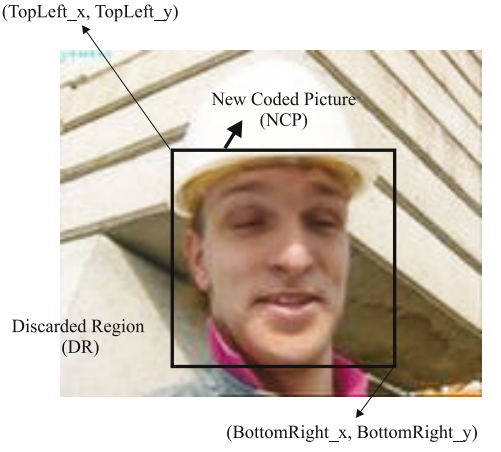


Fig. 1. Division of the original picture into a rectangular New Coded Picture (NCP) region and a Discarded Region (DR)

method achieves quality scalable coding by exploiting the block ordering scalability of H.264/AVC and improves the error resiliency of H.264/AVC in wireless ad-hoc networks by coding the video into multiple independently decodable descriptions. In addition, an error concealment algorithm is employed at the decoder to improve the reconstructed video quality when one of the two descriptions is lost during transmission.

In the quality-scalable H.264/AVC video coding algorithm [2], the blocks containing a large amount of motion constitute the base layer of the coded sequence and provides a basic video quality. The blocks with low motion are considered to be less important and coded for enhancement. In each frame, only the blocks belonging to the base layer are coded and the enhancement layer blocks are substituted with the collocated blocks of the previous reference frame. Therefore, the moving parts of the sequence are coded with a higher number of frames and less moving or static parts are coded with a lower frame rate.

The proposed MD coding extends the quality-scalable H.264/AVC algorithm to generate multiple independent descriptions for the blocks in the base layer. The detailed coding operation of the proposed method consists of the following steps. In the first step, motion vectors of the macroblocks (MBs) are estimated and the resulting information is used to determine the position and size of the coded region. The position and size of the New Coded Picture (NCP) region is defined by two points `(TopLeft_x, TopLeft_y)` and `(BottomRight_x, BottomRight_y)` as shown in Figure 1. First, the values `(TopLeft_x, TopLeft_y)` and `(BottomRight_x, BottomRight_y)` of the rectangle are initialized to zero. Then, the maximum of the two motion components of each MB (MV_x, MV_y) is compared to the motion activity threshold V_{th}. If

$$max(MV_x, MV_y) \geq V_{th}, \tag{1}$$

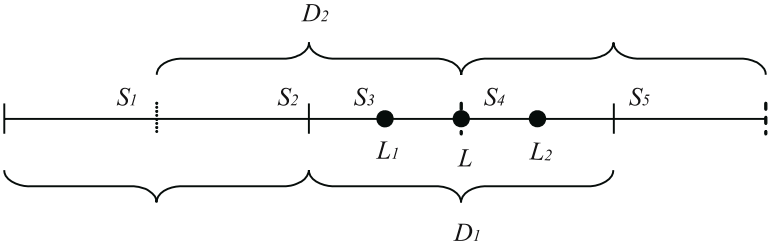


Fig. 2. The decision regions of the two coarse quantizers and the central quantizer defined in MSDQ [7]

then the values of the two corners are adjusted to include the current MB into the NCP region. Otherwise, the rectangle remains unchanged. Macroblocks are processed in the raster scan order and the two slices are determined at the end.

The blocks belonging to the NCP region, which are considered to be more important, are encoded using MDSQ [8] and two independent descriptions are generated. The blocks in the discarded region are not coded and replaced with the collocating blocks of the previous reference frame. In MDSQ, two quantizers whose decision regions are offset by one-half of a quantization interval from each other are used. Figure 2 shows an example of the decision regions of the two coarse quantizers and the central quantizer defined in MDSQ. Let D_1 and D_2 be the two descriptions that have been generated by two coarse quantizers. When both descriptions are received, we can decide whether L_1 or L_2 is the correct reconstruction level. On the other hand, when only one of the descriptions is correctly received, we cannot decide the actual (fine) quantization interval. For example, when only D_1 is available, the reconstruction level is decided to be the midpoint L of the quantization interval $[S_2, S_5]$ (Figure 2) and this coarse quantization makes the single-description reconstruction quality low.

In order to improve the video quality in the presence of lost description, we employ an error concealment algorithm that exploits the spatio-temporal smoothness property in video signals as in [7]. When one of the descriptions is lost during transmission, only one description that contains the coarsely quantized MBs is available at the decoder. To enhance the single-description reconstruction quality at the decoder, we refine the coarsely quantized coefficients using an algorithm that finds the optimal coefficient adjustment for each coarsely quantized transform coefficient by exploiting the smoothness property of video signals [7]. In this approach, our objective is to find the best reconstruction level for each coefficient so as to minimize a pre-defined spatio-temporal smoothness measure. The adjustment of the coefficient, δ, should be restricted to $\delta < \Delta$ (Figure 2), where

$$\Delta = \frac{S_5 - S_2}{2} = S_5 - L = L - S_2. \tag{2}$$

The refinement process for an $N \times N$ block can be represented in matrix form as

$$\boldsymbol{f} = \boldsymbol{T}(\hat{\boldsymbol{a}} + \boldsymbol{\Lambda}) \tag{3}$$

where $\boldsymbol{f}$ is a vector composed of the original sample values, $\hat{\boldsymbol{a}}$ and $\boldsymbol{\Lambda}$ are vectors containing the transform coefficients from the received description and the coefficient adjustments, respectively. $\boldsymbol{T} = [t_0, t_1, \cdots, t_{N^2-1}]^T$ and t_k is the k^{th} transform basis vector. All vectors are arranged in a row-major order. The optimal solution for Equation (3) with respect to $\boldsymbol{\Lambda}$ is given by

$$\boldsymbol{\Lambda}_o = \boldsymbol{f}_p - \boldsymbol{T}\hat{\boldsymbol{a}}, \tag{4}$$

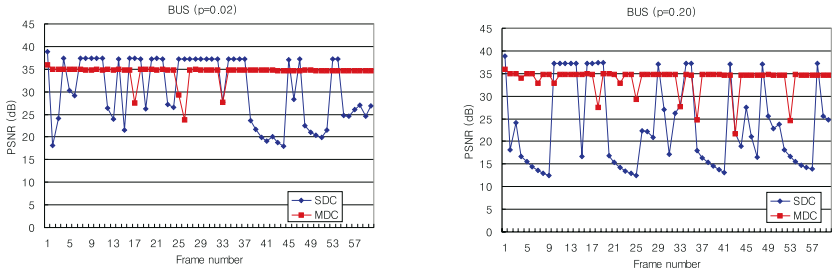
where $\boldsymbol{f}_p$ is the temporal prediction for an $N \times N$ block.

The two descriptions generated by the proposed MD coding method are transmitted over different paths to a receiver to increase the capacity and reliability of data transmissions. Since the probability of all paths being down is relatively small because of the statistical independence of the packet loss events over different paths in the multi-path transport system, the probability of losing both descriptions is relatively small. The proposed MD coder is quality-scalable because the MBs in the more important slice (in the NCP region) are coded with a higher frame rate for base quality, while the less important MBs (in the discarded region) are coded with a lower frame rate for enhancement. The motion activity threshold value V_{th} controls the base layer quality and the coding bit rate. The proposed method is also error-resilient because it generates two independently decodable descriptions for the MBs belonging to the more important region, transmits them over different paths to a receiver, and therefore, maintains satisfactory video quality most of the time by exploiting inter-description redundancy.

3 Performance Evaluation

A number of experiments were performed to examine the effectiveness of the proposed MD coding with multi-path transport versus the layered single description (SD) coding [2] with single-path transport in error-prone networks. We used the CIF (352×388 pixels/frame for luminance and 176×144 pixels/frame for chrominance components) sequence Bus encoded at 30 frames/s in the experiments. The SD coding and the proposed MD coding algorithms were individually implemented on top of the H.264/AVC JM 10.1 reference software.

For the SD coder, the input video frames are divided into two regions according to the motion activities in the scene and the MBs belonging to the NCP region are coded into one description and transmitted over a single path. The encoding bit rate is set to 1.5 Mb/s. In the proposed system, the input video frames are divided into two regions according to the motion activities in the scene and the MBs in the NCP region are encoded using MDSQ to generate two descriptions. For an intra MB, the transform coefficients are first quantized using the default quantizer, and then, two descriptions are generated by looking up the index assignment table proposed in [8]. For an inter MB, motion-compensation is performed based on the reconstruction generated using two descriptions, and the transform coefficients of the residual signal are coded into two descriptions using the same method as in intra MBs. For independent decoding of each description, the header and motion information of all MBs in a frame are duplicated in both



(a) (b)

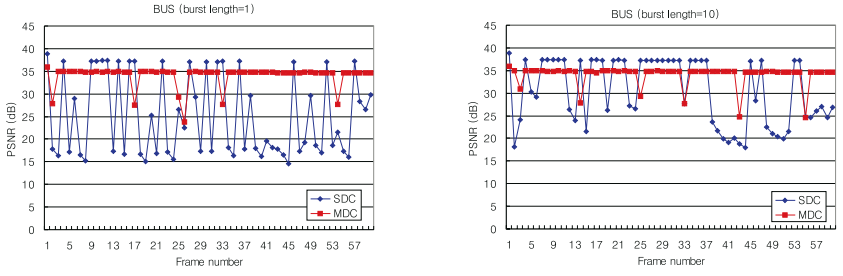
Fig. 3. Performance comparison between the SD coding with single-path transport and the proposed MD coding with multi-path transport (a) at 5% mean packet loss rate, and (b) at 20% mean packet loss rate

descriptions. The combined bit rates for the two descriptions is set to 1.5 Mb/s. For the proposed multi-path transport system, we assume that two independent paths are already set up in advance.

Error concealment is performed at the decoder when packets are lost. In the SD coder, the lost blocks in the NCP region are copied from the previous frame. In the proposed MD coder, if one of the descriptions is lost during transmission, then the coarsely quantized transform coefficients in the NCP region of the lost description are refined using the error concealment method introduced in Section 2. If both descriptions are lost, then the lost blocks are copied from the previous frame.

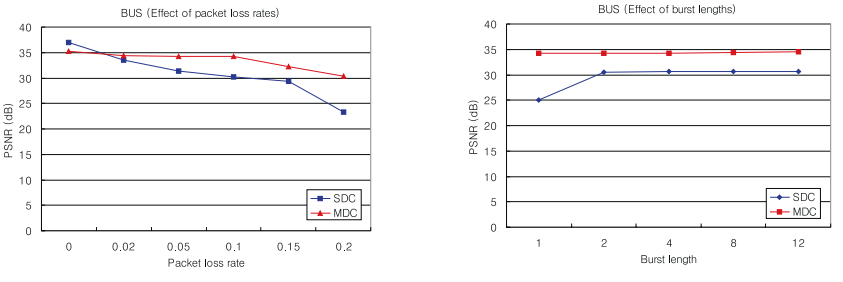
We compare the performance of the SD coder with single-path transport and the proposed MD coder with multi-path transport under various network conditions. Specifically, we examine the effects of packet loss rates and burst lengths for the two schemes. For the multi-path transport system, we assume that the two paths used are disjoint to each other. We can obtain two disjoint paths by using various heuristic algorithms as in [10].

1. *Effect of packet loss rates:*
   Figures 3 (a) and (b) illustrate the PSNR values of the two schemes at different mean packet loss rates. The burst length was set to 5 for these experiments. The results show that the SD coding with single-path transport is very vulnerable to packet losses and the effect of loss propagates to future frames. However, the proposed MD coding with multi-path transport recovers fast from packet losses and maintains a reasonable video quality as long as both descriptions are not lost simultaneously.
2. *Effect of error burst lengths*:
   Figures 4 (a) and (b) compare the performance of the two systems at different burst lengths when the average packet loss rate was set to 10%. We observe that the reconstructed video quality of the single-path transport system fluctuates abruptly, especially when the burst length is short. On the other



(a) (b)

Fig. 4. Performance comparison between the SD coding with single-path transport and the proposed MD coding with multi-path transport (a) when burst length = 1, and (b) when burst length = 10



(a) (b)

Fig. 5. Performance comparison between the SD coding with single-path transport and the proposed MD coding with multi-path transport (a) at different packet loss rates, and (b) with different burst lengths

hand, the simulation results show that the proposed MD coding with multi-path transport is not sensitive to burst lengths.

Figure 5 compares the two schemes when the average packet loss rate and burst length are varied. When there is no loss, the performance of the proposed MD coding is lower than that of the SD coding because of the redundancy introduced in generating multiple descriptions. However, as the packet loss rate increases the proposed MD coding with multi-path transport outperforms the SD coding with single-path transport because the proposed scheme is largely immune to packet losses and burst lengths as long as both descriptions are not lost simultaneously.

4 Conclusion

This paper presents a scalable multiple description coding for the H.264/AVC standard that enables robust video transmission in wireless ad-hoc networks.

The proposed method divides the input video frames into two regions according to the motion activities in the scene and encodes the MBs belonging to the more important region into two independently decodable descriptions. Each description is transmitted over an independent path to a receiver. If both descriptions are received without errors, then a high quality video is produced. If one description is lost, then the other correctly received description can still be decoded and can be used to estimate the information contained in the lost description. Simulation results show that the proposed MD coding is largely immune to packet losses because the reconstructed video quality is maintained as long as both streams are not simultaneously lost. As a results, the proposed video transmission system consisting of H.264/AVC-based scalable multiple description coding, multi-path transport, and decoder-side error concealment enables reliable video delivery over wireless ad-hoc networks.

References

1. ITU-T and ISO/IEC JTC1, Advanced video coding for generic audiovisual services, ITU-T Recommendation H.264-ISO/IEC 14496-10 AVC, 2003.
2. S. Benierbah and M. Khamadja: A new technique for quality scalable video coding with H.264, IEEE Trans. on Ciruits and Syst. Video Tech., vol. 19, pp. 877-908, June 2004.
3. Y. Wang, A. R. Reibman, and S. Lin: Multiple description coding for video delivery, Proceedings of The IEEE, vol. 93, no. 1, pp. 57-70, 2005.
4. J. G. Apostolopoulos: Reliable video communications over lossy packet networks using multiple state encoding and path diversity, in Proc. Visual Communications and Image Processing (VCIP '01), pp. 392-409, Jan. 2001.
5. N. Gogate and D. M. Chung and S. S. Panwar and Y. Wang: Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding, IEEE Trans. on Circuits Syst. Video Technol., vol. 12, no. 9, pp. 777-792, Sept. 2004.
6. E. M. Royaer and C.-K. Toh: A review of current routing protocols for ad-hoc mobile wireless networks, IEEE Personal Comm., pp. 46-55, Apr. 1999.
7. Y.-C. Lee, Y. Altunbasak, and R. M. Mersereau: A coordinated multiple description scalar quantization and error concealment algorithm for error-resilient video streaming over lossy channels, Proc. Intl. Conf. on Image Processing, pp. 99-103, 2002.
8. V. A. Vaishampayan: Design of multiple description scalar quantizers, IEEE Trans. on Info. Theory, vol. 39, no. 3, pp. 821-834, May 1993.
9. S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi: Video transport over ad-hoc networks: Multistream coding with multipath transport, IEEE Trans. on Sel. Areas in commm. vol. 21, no. 10, pp. 1721-1737, Sec. 2003.
10. A. C. Begen, Y. Altunbasak, O. Ergun, and M. M. Ammar: Multi-path selection for multiple description encoded video streaming, EURASIP signal processing: image comm., vol. 20, pp. 39-60, Jan. 2005.
11. U. Horn, K. Stuhlmuller, M. Link, and B. Girod: Robust Internet video transmission based on scalable coding and unequal error protection, Signal Processing: Image Communication, vol. 15, pp. 77-94, Sept. 1999.

Emotion-Based Textile Indexing Using Colors, Texture and Patterns

Soo-jeong Kim[1], Eun Yi Kim[2], Karpjoo Jeong[2], and Jee-in Kim[2,*]

[1] Department of Computer Engineering, Konkuk University, Seoul, Korea
cryolite@konkuk.ac.kr
[2] CAESIT, Konkuk University, Seoul, Korea
{eykim, jeongk, jnkm}@konkuk.ac.kr

Abstract. We propose a textile indexing system which can classify textile images based on human emotions. The emotions can be regarded as emotional reactions of human beings when they view specific textile images. The evaluation system starts with extracting features of textile images such as colors, texture and patterns using various image processing techniques. The proposed system utilizes both fuzzy rules and neural networks. The fuzzy rules are determined for six emotional features which can be formulated with respect to color and texture. On the other hand, the neural network is used for recognizing patterns which can be used in classifying textile images based on the 4 other emotional features. For the machine learning component of the system, we selected 70 subjects so that they could view and annotate 160 textile images using ten pairs of emotional features. The fuzzy rule based component of the system uses color features and texture features in order to predict six pairs of emotional features such as (warm, cold), (gay, sober), (cheerful, dismal), (light, dark), (strong, weak), and (hard, soft). The neural-network based component of the system can predict four pairs of emotional features such as (natural, unnatural), (dynamic, static), (unstable, stable) and (gaudy, plain). Our experimental results showed that the proposed system was effective for predicting human emotions based on textile images and improving the accuracy of indexing the textile images based on emotional features.

1 Introduction

Human affect and emotion had been studied and applied to various areas such as user interface development [1-5], psychological study [6-8,15], content-based retrieval system [11,12], ubiquitous computing [13,14], and so on. In particular, human emotion has been a vital ingredient in designing visual aspects of textiles, fashion, and homepages [9]. However, due to high ambiguity in perceiving/interpreting visual information at the affective level, it is quite difficult to predict human emotions directly from visual information. Therefore, it is an important issue to find out relations between human emotions and visual information.

In many researches related to textiles, emotional features have been defined in various ways. Kobayashi proposed emotional features for the field of fashion [6,18].

* Corresponding author.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 9 – 18, 2006.
© Springer-Verlag Berlin Heidelberg 2006

While Soen developed emotional expressions [7], which include 13 pairs of adjectives of positives and negatives such as (like, dislike), (beautiful, ugly), (natural, unnatural), (dynamic, static), (warm, cold), (gay, sober), (cheerful, dismal), (unstable, stable), (light, dark), (strong, weak), (gaudy, plain), (soft, hard), and (light, heavy). We used the emotional features developed by Soen, because they are used in many previous researches and can present general emotional features. Color, texture, and pattern (shape) are representative features of visual information. For recognizing emotional features, Soen and Kim used regression equations with color and texture [7, 9]. But, these models are not sufficient, because nonlinear relationships could exist between physical features and emotional features in real world. To resolve the problem, Um developed fuzzy rules and a neural network based evaluation system for converting physical features into emotional features [8]. The system performed well, but it is inappropriate to apply the results to more general cases because the sizes of the sample data used in the studies were not sufficient.

Therefore, in our previous work [16], we studied relationships between Soen's emotional features and 160 textiles collected in real commercial areas for extending general cases. Only 6 emotional features[1] could be distinguished by color and texture. So we could evaluate them with a fuzzy-based evaluator. However, it can be applied only to the emotions which can be distinguished by color and texture.

In this paper, we extended our previous research in order to investigate correlations among emotional features and patterns. The additional 4 emotional features such as {(natural, unnatural), (dynamic, static), (unstable, stable), (gaudy, plain)}, which could not be classified by the fuzzy rule based system, were examined. So a neural network system was constructed to extract emotional features. We performed a series of experiments in order to evaluate the proposed system which utilized both the fuzzy rule based system and the neural network based system.

The experimental results demonstrated that our proposed system could classify emotional features according to individual physical features and their values. And then we could file human emotional features affected by the physical features individually based on these results. In that way, we could show a potential that our system could be used for indexing textile images based on emotional features. Hopefully, the system could be used in commercial applications in the fields of textile design, fashion design, and commodity design.

The rest of the paper is organized as follows. Section 2 covers our physical feature extraction component. The proposed evaluation system is composed of a fuzzy rule based evaluation system and a neural network based system. Section 3 describes how the proposed system operates. The performance of the proposed system was evaluated with sample textile images. Section 5 shows the experimental results.

2 Physical Feature Extraction

In order to collect information about the textile images and their corresponding emotional features, we selected 70 subjects. We asked them for manually annotating

[1] They are emotional feature sets such as (warm, cold), (gay, sober), (cheerful, dismal), (light, dark), (strong, weak), and (hard, soft).

textile images with emotional features which they felt from the images. The textile images were collected from Dongdaemoon textile shopping malls in Seoul, Korea. Then, we analyzed and tried to find the relationships between the physical features of textile images and their corresponding emotional features from the data collected through these works.

In our previous research, we used only color and texture in order to identify the emotional features of textile images [16]. However, the indexing power of these features was limited to some emotional features. Since we want to classify other features such as (gaudy, plain), we studied patterns of textile images and their corresponding emotional features. So, we additionally used the pattern feature in order to find more emotional features from textile images. For recognizing the pattern feature from textile images, we developed a physical feature extraction system using a neural network.

Our physical feature extraction system is composed of the color and texture extraction component and the pattern extraction component. The RGB color space of images is transformed to the corresponding HSI color space [19]. The HSI color space is closer to human perception than the RGB color space [7, 8]. In addition to the color properties, we use the texture property as a physical feature of images. In order to represent the degree of sparseness of images, we use GLRLM (Gray Level Run Length Matrix) [19], which is defined as follows:

$$GLRLM(m,n) = CARD[\{(i,j) \in \underset{MN}{\wedge} \mid I(i,j) = m, \tau(m,\theta) = n\}],$$

where $\wedge MN$ is M×N matrix and I(i, j) denotes the pixel value at the site (i, j). And $\tau(m,\theta)$ is the run-length of a pixel in the direction of an angle, θ, and CARD means the cardinality of the set. In order to remove effects of the direction, GLRLMs were averaged in the direction of 0°, 45°, 90°, 135° and the run lengths were normalized so that the sum becomes one. The texture value C is defined as a scalar, SRE (Short Run Emphasis), using the following equation.

$$C = \sum_{0}^{m} \sum_{0}^{n} \frac{GLRLM(m,n)}{n^2}$$

We also utilize patterns in extracting features of images. We propose to use edge features, not only because they are effective for indexing images [17], but also because they present patterns of images which are not affected by intensity values of images.



Fig. 1. Pattern Extraction Component

The pattern extraction component operates as shown in Fig 1. First, the edges were detected using a mask. The image is divided into sub-images. In the next step, the system summarizes white pixels of each sub-image and chooses a sub-image which has the maximum count of white pixels. The selected sub-image is used as an input for the neural network which would evaluate pattern properties of images.

3 Evaluation System

In our previous work [16], we reported that color and texture features could be used in indexing textile images with some emotional features such as (strong, weak), (sober, gay), (light, dark), (dismal, cheerful), (warm, cold), and (soft, hard). And we developed fuzzy rules according to color and texture.

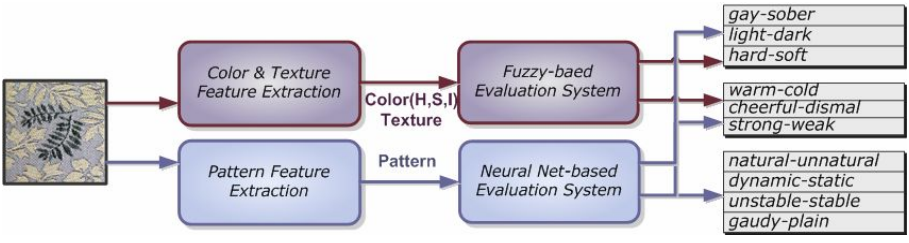


Fig. 2. Evaluation System

In this paper, we extend our previous work by adding a neural network based evaluation system. As shown in Fig. 2, the color features and the texture features of textile images are processed by the fuzzy rule based system, while the pattern features are evaluated by the neural network based system.

3.1 Fuzzy Rule Based Evaluation

The fuzzy rule based system could distinguish six pairs of emotional features expressed as opposite adjectives such as (strong, weak), (sober, gay), (light, dark), (dismal, cheerful), (warm, cold), and (soft, hard). The texture features and the color features were used in the experiments.

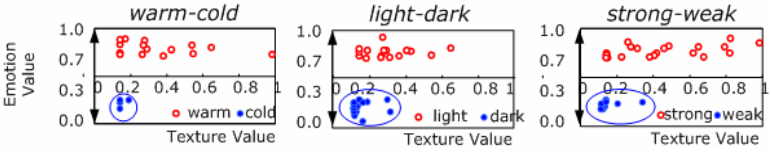


Fig. 3. Distribution of texture values according to emotional values

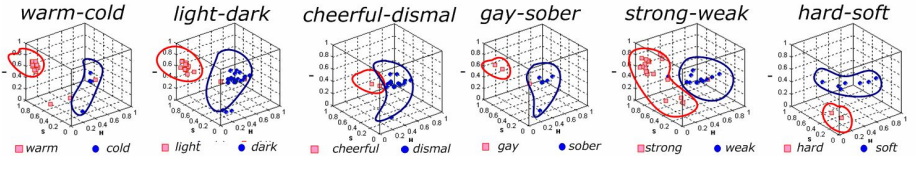


Fig. 4. Relationships between the color features and the emotional feature pairs

Fig 3 shows the relationships between the texture features and their corresponding emotional features. As shown in Fig 3, emotional feature pairs such as (warm, cold), (light, dark), and (strong, week) can be distinguished by the texture features from its adverse emotional features. Fig 4 shows the relationships between the color features and the respective pairs of emotional features. As can be seen in Fig 4, the color features can give hints for predicting the six pairs of emotional features included in textile images. Most of emotion groups can be partitioned by the color properties, for example, the textiles including the 'sober' emotion are clustered into the partition with low saturation values, while ones including emotion 'gay' are clustered into the partition with very small hue values.

For the conversion of the physical features such as color and texture to the corresponding emotional features, we developed a fuzzy rule based system [20, 21]. To formulate fuzzy rules of the system, an appropriate fuzzy partition is performed, since the performance of the derived rules is sensitive to the partition of a pattern space. In this paper, we use the following symmetric triangular membership function.

$$\mu_1^{\,K}(x) = \max\left\{1 - \left|x - a_1^{\,K}\right| / b^K, 0\right\}, \qquad i = 1,2,...,K$$

Where

$$a_i^k = \frac{(i-1)}{K} + \frac{1.5}{2K}, i = 1,2,...K; \qquad b^k = \frac{1.5}{2K}$$

Here, $a_1^{\,K}$ is the center where the grade of membership is equal to 1 and b^K is the spread of the membership function. In our work, K is fixed to 5.

The following 14 control rules in Table 1 are used. The 11 rules are generated for the color properties and the other 3 rules are generated for the texture properties. Based on the results from our previous study [16], we first applied the rules of the texture feature are applied and the color based rules are applied after the first step.

Table 1. Fuzzy rules for properties

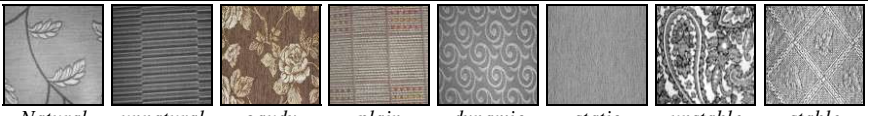
property	*Rule*	
color	R_1	If *H(x)* is very small, and *I(x)* below middle, it is a warm
	R_2	If *S2(x)* is very small, *I(x)* is high, it is a cold
	R_3	If *H(x)* is very small, and *I(x)* is middle, it is a gay
	R_4	If *S2(x)* is small , and *I(x)* is high or *I(x)* is very small , it is a sober
	R_5	If *H(x)* is very small , and *I(x)* above middle, it is a cheerful
	R_6	If *S2(x)* below middle, it is a dismal
	R_7	If *H(x)* is very small, and *I(x)* is middle, it is a light
	R_8	If *S2(x)* is very small, *I(x)* is high or *I(x)* is very small, it is a dark
	R_9	If *I(x)* below middle ,it is a strong
	R_{10}	If *S(x)* is very small, *I(x)* above middle, it is a weak
	R_{11}	If *S(x)* is very small, and *I(x)*above middle, it is a soft
texture	R_1	If *C(x)* is very small, it is a cold.
	R_2	If *C(x)* below small, it is a dark.
	R_3	If *C(x)* below small, it is a weak.

3.2 Neural Network Based Evaluation

The patterns of images are quite important factors in affecting human emotional features. In order to investigate correlations among patterns and emotional features, we tested additional 4 emotional features, which could not be distinguished by color and texture.

Table 2 shows sample textile images and their corresponding 4 emotional features. Subjects classified the (natural, unnatural) emotional feature by recognizing natural object such as leaves, fruit, or types of line. In the case of the (gaudy, plain) pair, the flower of the textile image seem to make people feel 'gaudy', while the straight lines and the squares make people feel 'plain'. The 'dynamic' and 'static' pair is distinguished by line-types and arrangement of patterns.

Table 2. Sample images for each emotion feature



Natural	*unnatural*	*gaudy*	*plain*	*dynamic*	*static*	*unstable*	*stable*

The emotional features could not be distinguished by the color feature and the texture feature using the fuzzy rule based system. Therefore, we developed a neural network based evaluation system for recognizing textile patterns.

Here, we use a neural network as a filter to automatically discriminate the patterns in a textile. The network scans all the pixels in the input textile so as to classify them as one of the ten emotional features. The network receives gray-scale values of a pixel and its neighboring pixel within a small window. Then, the output of the network represents the class of the central pixel in the input window. A diagram of the proposed pattern recognition system is presented in Fig 5.

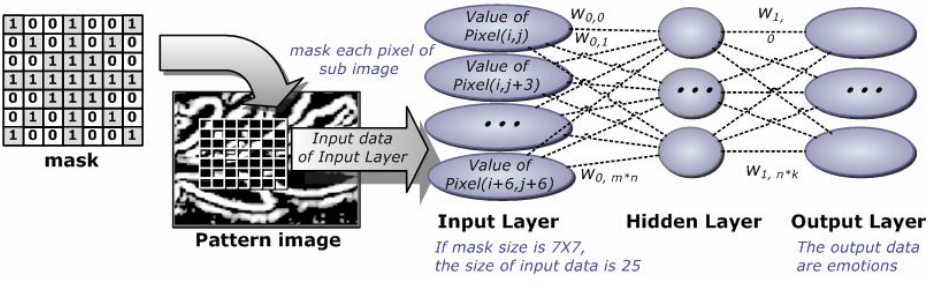


Fig. 5. The proposed neural network

We developed four evaluators for each emotional feature. Each evaluator is composed of an input layer, a hidden layer and an output layer. The input layer of the network has 129 nodes, the hidden layer has 77 nodes, and the output layer has 2 nodes. The adjacent layers are fully connected. The hidden layer operates as a feature

extraction module. The output layer is used to determine the class of a pixel: one of four emotional features.

When the neural network is used as a classifier, it is quite important to select an appropriate representation scheme to represent a pattern or a shape. The simplest way to characterize the variability in a pattern is by noting the gray-level values of the raw pixels. This set of gray values becomes the features set on which the classification is based. An important advantage of this approach is the speed with which images can be processed, as the features do not need to be calculated. However, the main disadvantage is that the size of the feature vector becomes large. Accordingly, we use a configuration for autoregressive features (only the pixels corresponding to the 1 in the mask), instead of all the pixels in the input window. This reduces the size of the feature vector, thereby results in an improved generalization performance and classification speed.

Using these extracted features, we train the neural network. After training with a back-propagation algorithm, the neural network outputs the real values between 0 and 1. If a node has the largest value among other nodes, the input pixel is labeled as the emotional feature corresponding to the output node. After scanning all the pixels with the neural network, the input textile image is assigned by the emotional feature which labels most of pixels.

4 Experimental Results

The proposed system was constructed with 160 textile images. The images were collected from a commercial curtain pattern book obtained from Dongdaemoon textile shopping malls in Seoul, Korea. The textile images were randomly selected and they are related to the emotional features. One image may be related to several emotional features, and numbers of images related to each emotional feature ranges from 17 to 73. 70 subjects participated in our experiments for validating the proposed system. They viewed textile images and were asked to manually annotate the images with the emotional features that they felt from the images.

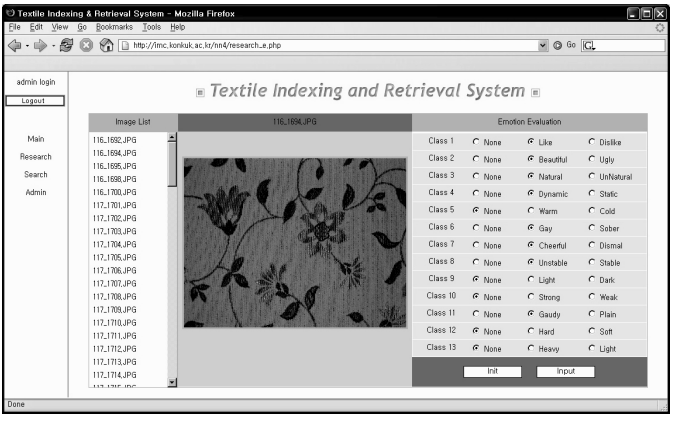


Fig. 6. Data collection system

A data collection system was constructed and the subjects accessed the system through the Internet. The experimental data were collected through the Internet, and the URL of the site is http://imc.konkuk.ac.kr/nn4/research_e.php.

The half of 160 textile images were used for generating the fuzzy rules (described in Section 3.1) and the neural network (explained in Section 3.2), the other images were tested for assessing the performance of the proposed system. The results are summarized in Table 3 and Table 4.

Table 3. Performance of the 6 emotional features according to color, texture, and pattern(%)

Emotional features	*color + texture*		*pattern*	
	*recall**	*precision***	*recall*	*precision*
cold-warm	85.64	79.99	62.50	93.48
dark-light	82.94	83.40	50.00	50.00
weak-strong	85.82	82.77	71.90	75.19
soft-hard	93.50	83.30	50.00	40.00
dismal-cheerful	100.00	100.00	75.00	95.83
sober-gay	86.56	87.36	62.50	78.57
total	89.08	86.14	61.98	72.18

* recall = # of correctly detected textile images / # of textile images.
** precision = # of correctly detected textile images / # of detected textileimages.

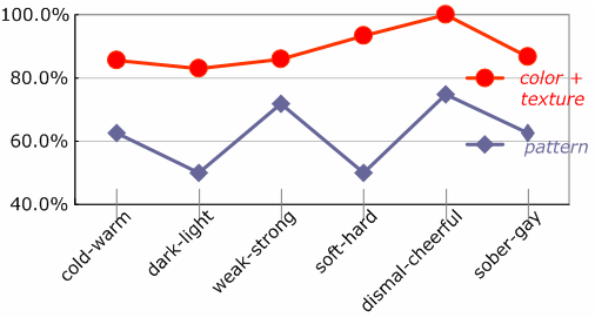


Fig. 7. Difference of performance according to each feature

Table 4. Performance of the 4 emotional features using pattern(%)

Emotional features	*pattern*	
	recall	*precision*
natural-unnatural	79.46%	80.56%
dynamic-static	92.86%	96.88%
unstable-stable	83.33%	85.00%
gaudy-plain	75.00%	76.89%
total	82.66%	84.83%

As can be seen in Table 3 and Fig 7, for the six emotional features, the indexing method using color and texture could guarantee superior performance than the pattern based method. The former produced the accuracy of 89.08%, and the latter showed the accuracy of 61.98%. This shows that there is relatively low links among the four emotional features and patterns as compared with color and texture.

On the other hand, the four additional emotional features, which could not been successfully evaluated by color and texture, could be classified using patterns. It showed the accuracy of 82.66 % as shown in Table 4. It demonstrated that they have relatively closer correlations than the pattern based method.

Therefore, when the color and texture based method can be used in order to predict the six pairs of emotional features and the pattern based method can be used in order to predict the additional four pairs of emotional features. It showed more than 85% performance in its accuracy.

5 Conclusions

As digital images of immense size are becoming accessible over the Internet, efficient image indexing has been receiving considerable attentions. Though many techniques have been developed so far, they lacked the capability to utilize human emotions in the process of image indexing. Accordingly, this paper presented an approach for labeling images based on human emotions and showed its affirmative results. Based on physical features of images (namely, color, texture and pattern), we aimed to index images using ten emotional features.

The proposed system automatically extracts physical features, and then predicts emotional features using them. The extraction stage extracts the color, texture, and pattern properties from a textile image using computer image processing techniques. The extracted features are analyzed and converted to the corresponding emotional features. The evaluation systems are based on the fuzzy rules and the neural network systems. The physical features such as color and texture were used in the fuzzy rule based system. And the pattern feature was used in the neural network based system.

Based on our experimental results, the six emotional features showed lower correlations with patterns as compared with color and texture. And the four emotional features seemed to be more closely link with patterns. So, the proposed system was composed of the fuzzy rule system for the 6 emotional features and the neural network system for the additional 4 emotional features. The combined system showed more than 85% performance in its accuracy.

Our studies demonstrated that the proposed system can classify emotional features according to physical features of images. Even though images of our study are textile images only, this method can be applied to other fields, because it can predict human emotional features according to physical features of images. Therefore, our study can be applied to many fields such as textile design, fashion design, and commodity design.

Acknowledgements

This work was supported by the Ministry of Information & Communications, Korea, under the Information Technology Research Center(ITRC) Support Program.

References

1. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. G., "Emotion recognition in human-computer interaction, IEEE Signal Processing magazine", Vol. 18, No. 1, pp. 32-80, Jan. 2001.
2. Bjom Schuller, Gerhard Rigoll, and Manfred Lang, "Emotion recognition in the manual interaction with graphical user interfaces", IEEE International Conference on Vol.2, pp.1215 - 1218, June 2004.
3. G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions, IEEE Trans. Pattern Anal. Machine Intell.", Vol. 21, pp. 974–989, Oct. 1999.
4. Hatice Gunes and Massimo Piccardi, "Fusing face and body gesture for machine recognition of emotions", IEEE International Workshop on Robots and Human Interactive Communication, 2005.
5. D. Ververidis, and C. Kotropoulos, "Automatic speech classification to five emotional states based on gender information", Proceedings of the EUSIPCO2004 Conference, Austria, pp. 341-344, Sept. 2004.
6. Shigenobu Kobayashi, "The Aim and Method of the Color Image Scale, Color Research & Application", Vol. 6, pp. 93~107, Summer 1981.
7. Naoki Kawamoto, Toshichi Soen, "Objective Evaluation of Color Design.II, Color Research & Application", Vol.18, pp. 260~266, August 1993.
8. Jinsub Um, Kyoungbae Eum, Joonwhoan Lee, "A Study of the Emotional Evaluation Models of Color Patterns Based on the Adaptive Fuzzy System and the Neural Network", Color Research & Application, Vol. 27, pp. 208-216, Jun 2002
9. Kim Jinwoo, Lee Jooeun, Choi Dongseong, "Designing emotionally evocative homepages : an empirical study of the quantitative relations between design factors and emotional dimensions", In International Journal of Human-Computer Studies, Vol. 59 , No. 6, pp. 899-940 , 2003.
10. Ping Zhang , Na Li, "The importance of affective quality", Communications of the ACM archive, Vol. 48 , Issue 9 , pp. 105-108, September 2005.
11. M.G.L.M van Doorn and A. P. de Vries, "Psychology of Multimedia Databases", In Proceedings of the 5th Digital Libraries Conference, San Antonio, TX, 2000.
12. Nadia Bianchi-Berthouze, "K-DIME: An Affective Image Filtering System", IEEE MultiMedia, Vol.10, No.3, pp103~106, July/September 2003
13. Tetsuro Hori, Kiyoharu Aizawa, "Context-based Video Retrieval System for the Life-Log Applications", Workshop on Multimedia Information Retrieval, pp 31-38, November 7, 2003.
14. Datchakorn Tancharoen, Toshihiko Yamasaki, Kiyoharu Aizawa, "Practical Experience Recording and Indexing of Life Log Video", CARPE'05, pp. 61-66, November 11, 2005.
15. Li-Chen Ou, M. Ronniner Luo, "A Study Of Colour Emotion and Colour Preference.Part I : Colour Emotions for Single Colours", Color Research & Application,Vol. 29, pp.232-240, Jun 2004.
16. Eun Yi Kim, Soo-jeong Kim, "Emotion-based Textile Indexing Using Color, Texture , Fuzzy Systems and Knowledge Discovery", Vol. 3613, pp. 1077~1080, 2005.
17. P. Pala, S. Santini, "Image retrieval by shape and texture", PATREC: Pattern Recognition, pp. 517-527, 1999.
18. Shigenobu Kobayashi, "COLOR IMAGE SCALE", Kodansha, 1991.
19. Gonzalez et al. "Digital Image Processing", Addison-Wesley, 2002.
20. Ricard O.Duda, Peter E.Hart, and David G.Stork, "Pattern Classification", 2nd Ed., Wiley Interscience, 2001.
21. Valluru B. Rao, Hayagriva Rao, "C++ Neural Networks & Fuzzy Logic", 2nd Bk&Dsk Ed., M&T Books, 1995.

Affine Camera for 3-D Retinal Surface Reconstruction

Thitiporn Chanwimaluang and Guoliang Fan

School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK 74078

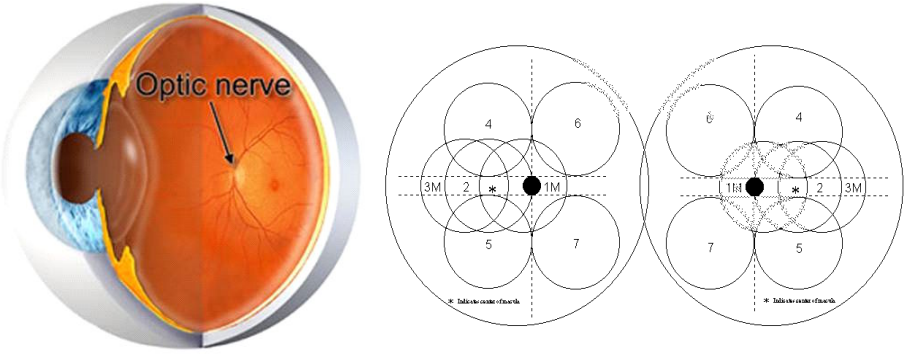
Abstract. We study 3D retinal surface reconstruction by using an affine camera due to two following reasons: (1) NIH's retinal imaging protocols specify a narrow field of view and (2) each retinal image has small depth variation. Specifically, we incorporate the prior knowledge of human retina geometry in the reconstruction process, and introduce a point-based approach to estimate the retinal spherical surface. We also show that lens distortion removal and affine bundle adjustment improve the reconstruction error in terms of the deviation from the underling spherical surface. Simulation results on both synthetic data and real images show the effectiveness and robustness of the proposed algorithm.

1 Introduction

Diabetes is the leading cause of blindness among working-age Americans, and many patients with vision-threatening diabetic retinopathy remain asymptotic until blindness occurs [1]. The great majority of this blindness can be prevented with proper eye examination and treatment by ophthalmologists who rely on the results of randomized clinical trials, called Early Treatment Diabetic Retinopathy Study (ETDRS), to guide their treatment of patients with diabetes [2]. ETDRS requires sets of retinal images to be captured from different fields of an eye. Because ophthalmologists rely on multiple retinal images for disease diagnosis and evaluation, these images need to cover a required area of the retina. The ETDRS imaging standard, specifying seven stereoscopic 30° fields of each eye, is illustrated in Fig. 1(b). A visual model, i.e. 3-D retinal surface reconstruction shown in Fig. 1(a), can (1) assist ophthalmologists in diagnosing, analyzing, and evaluating the disease; (2) facilitate clinical studies; and (3) be used as a spatial map during laser surgical procedures. Visual reconstruction is a process to recover a 3-D scene or a model from multiple images. It is usually referred to as the structure from motion, SFM, problem. A process usually recovers objects' 3-D shapes, cameras' poses (positions and orientations), and cameras' internal parameters (focal lengths, principle points, and skew factors). Many possible camera models exist. A perspective projection is the standard camera model. However, other simplified projections, e.g. affine or orthographic models, are proved useful and practical for a distant camera.

[1] http://www.nei.nih.gov/diabetes/
[2] http://www.nei.nih.gov/neitrials/viewStudyWeb.aspx?id=53

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 19–30, 2006.
© Springer-Verlag Berlin Heidelberg 2006



(a) (b)

Fig. 1. A 3D eye model and ETDRS seven-standard fields (right/left eyes) (http://eyephoto.ophth.wisc.edu/Photographers.html)

3-D retinal surface reconstruction refers to the global geometric shape recovery that has been an interesting and challenging problem. Deguchi et. al. [1,2] modelled both the fundus camera and the human cornea with a virtual optical lens. They utilized the fact that a retinal surface has a spherical shape and imaging a sphere through the eye lens results in a quadratic surface. The camera was calibrated by using the two-plane method. Then, eye lens parameters were estimated iteratively to recover fundus's spherical surface. Choe et. al. [3] used PCA-based directional filters to extract candidate seed points (Y features), and plane-and-parallax was employed to estimate the epipolar geometry based on which the stereo pair was rectified. Then, a Parzen window-based mutual information was used to generate dense disparity map. Promising 3D retinal reconstruction results were reported in [3]. However, the stereo reconstruction technique does not work for ETDRS stereo image pairs due to two problems. First, each ETDRS image covers a small region on a retina, and the overlapps between seven fields are relatively small (as shown in Fig. 1), leading to sparse feature points. Second, there is a series of optics involved in ETDRS imaging, resulting in significant lens distortions in retinal images.

In this work, we will derive an affine camera from the standard projective camera and show that an affine camera is an appropriate simplified model for retinal surface reconstruction from ETDRS images. Retinal images, first, need to be corrected due to lens distortions. An initial affine shape is obtained from a previously proposed factorization method. The affine shape and the camera model are, then, jointly refined by an affine bundle adjustment algorithm. Later, the geometrical constraint of human retina is imposed to recover a Euclidean structure up to a similarity transform and we introduce an efficient point-based linear approach to approximate the retinal spherical surface. Compared with previous methods, the proposed one is robust, efficient, and less sensitive to noise and lens distortion due to the linear nature of the affine camera.

2 Affine Camera for 3-D Retinal Surface Reconstruction

We provide a mathematical proof of an affine camera from the standard projective camera and derive its condition for good reconstruction performance. Let us start with general projective camera. Image and world coordinates are related by

$$m = \begin{bmatrix} (f_x R_1^T + sR_2^T)M + f_x D_x + sD_y + c_x \\ f_y R_2^T M + f_y D_y + c_y \\ R_3^T M + D_z \end{bmatrix}, \tag{1}$$

where m and M denote image and world homogeneous coordinates respectively. R_i^T is the $i-$th row of the rotation matrix $\mathbf{R}$ and $D_x = -R_1^T t_x$, $Dy = -R_2^T t_y$, $D_z = -R_3^T t_z$ where t_x, t_y and t_z are translation parameters. f represent the focal length of the camera. (c_x, c_y) and s are principal point and skew angle of image's pixel respectively.

If we assume that the relative depth in principal ray direction $R_3^T M_i$ is small compared to D_z, the distance between camera center and the world's origin in the direction of camera's principal ray, then Equation (1) can be rewritten as

$$m = \begin{bmatrix} (f_x R_1^T + sR_2^T)M + f_x D_x + sD_y + c_x \\ f_y R_2^T M + f_y D_y + c_y \\ D_z \end{bmatrix}. \tag{2}$$

Therefore, we get an affine camera model which can be represented in a mathematical form as

$$\mathbf{T}_{affine} = \begin{bmatrix} f_x R_1^T + sR_2^T & f_x D_x + sD_y + c_x D_z \\ f_y R_2^T & f_y D_y + c_y D_z \\ \mathbf{0}_3^T & D_z \end{bmatrix}. \tag{3}$$

By comparing Equations (1) and (2), the difference is only in the last row. Hence, we will rewrite both equations as follows

$$m_{projective} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ d+\delta \end{bmatrix} m_{affine} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ d \end{bmatrix}, \tag{4}$$

$$\hat{m}_{projective} = \begin{bmatrix} \frac{\tilde{x}}{d+\delta} \\ \frac{\tilde{y}}{d+\delta} \end{bmatrix} \hat{m}_{affine} = \begin{bmatrix} \frac{\tilde{x}}{d} \\ \frac{\tilde{y}}{d} \end{bmatrix}, \tag{5}$$

where $\tilde{x}$ and $\tilde{y}$ represent $(f_x R_1^T + sR_2^T)M + f_x D_x + sD_y + c_x$ and $f_y R_2^T M + f_y D_y + c_y$ respectively. d denotes D_z and δ denotes $R_3^T M$. $\hat{m}$ represents a non-homogeneous coordinate. Then,

$$\frac{\hat{m}_{projective}}{\hat{m}_{affine}} = \frac{d}{d+\delta}. \tag{6}$$

From Equation (6), we can conclude that if δ, or relative depth in principal ray direction, is small compared with D_z then the affine camera model is a good choice. In a retinal image case, the retinal surface depth is relatively small

compared with the distance from the retina to a fundus camera. Therefore, an affine camera is a preferred camera model to represent a fundus camera.

3 Correspondence Selection

Correspondence selection is a critical step for feature-based 3D reconstruction algorithms. In this work, point correspondences (bifurcations and crossovers) are automatically selected by using our previously proposed hybrid retinal image registration [4]. The algorithm can be summarized as follows. First, binary vascular trees are extracted from retinal images by match filtering and appropriate thresholding. Second, zeroth-order translation is estimated by an area-based approach that maximizes the mutual information between two binary images. Specifically, a local entropy-based peak selection scheme and a multi-resolution searching strategy are developed to improve the accuracy and efficiency of translation estimation. Third, a feature-based registration method is used along with other decision-making criteria to estimate higher-order transformation and further refine point correspondences. Two registration results are shown in Fig. 2, and some correspondence selection examples shown in Fig. 3.

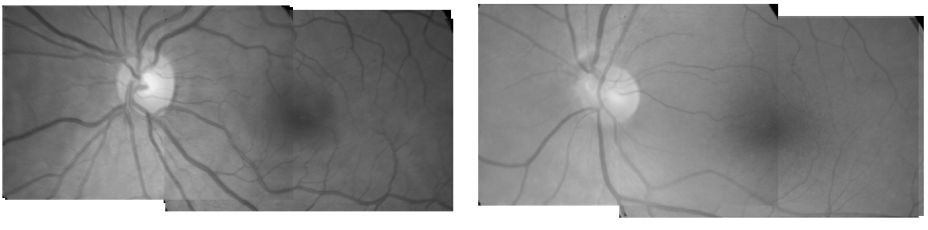


Fig. 2. Two registration results each of which has two stereo ETDRS pairs (fields 1/2)

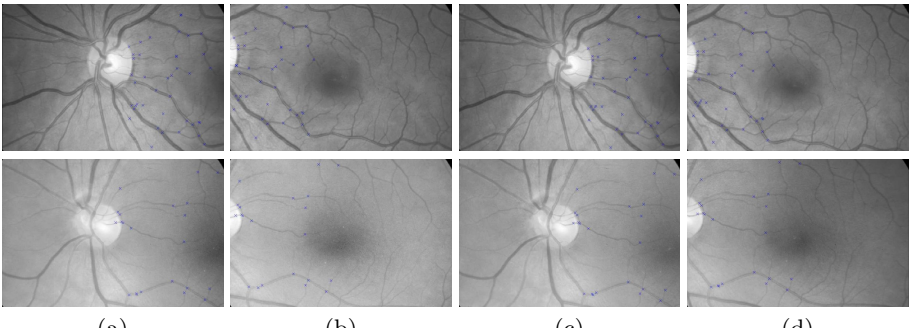


Fig. 3. Two sets of retinal images with marked point correspondences. The first/second rows (corresponding to Fig. 2 (left) and (right), respectively) include a stereo pair of field 1 ((a) and (c)) and that field 2 ((b) and (d)).

4 Lens Distortion Removal

There is a series of optics involved in the retinal imaging process, which includes the actual fundus camera, the digital camera, and the human cornea. All of these optics could be modeled as one virtual lens that contributes to certain lens distortion, e.g., radial distortion, in retinal images [1,2]. The lens distortion has to be removed prior to 3D retinal surface reconstruction. In this work, we employ the planar pattern calibration method proposed in [5]. The solution can be solved through minimizing an algebraic distance then refining it through Levenberg-Marquardt algorithm with a following cost function

$$\sum_{i=1}^{f}\sum_{j=1}^{n} \|m_{ij} - \check{m}(\mathbf{K}, \mathbf{R}_l, k, d_i, M_j)\|^2. \tag{7}$$

where we have f views/images and n correspondences. $\check{m}(\mathbf{K}, \mathbf{R}_i, d_i, M_j)$ is the projection of point M_j in an image i. R_l denotes $l-$th row of rotation matrix $\mathbf{R}$ and d denotes $-\mathbf{R}t$. k are coefficients for lens distortion. We have created a set of chessboard images using the actual fundus camera. Then we obtained all lens parameters by using the camera calibration toolbox [3] which are used to remove the lens distortion in real retinal images.

5 Affine Structure from Motion

An affine structure from motion theorem was first proposed by Koenderink and Van Doorn [6]. They have shown that two distinct views are enough to reconstruct a scene up to an arbitrary affine transformation without a camera calibration. They have suggested the use of local coordinate frame (LCF). Later their algorithm has been refined by Quan et.al. [7], Demy et.al. [8], and Shapiro [9]. Tomasi and Kanade [10], then, proposed an affine factorization method which eliminates the use of LCF and instead utilizes the entire set of points.

5.1 Initial Retinal Affine Surface

We assume an affine projection for a fundus camera because (1) the ETDRS imaging standard specifies a 30° field of view each eye (narrow field of view); (2) each retinal image has small depth variation. We use affine factorization [10] method for initial reconstruction because the approach can accommodate multiple images and utilize the use of all feature points. Suppose there are f retinal images and n point correspondences from each image.

$$\mathbf{W} = \mathbf{PM}, \tag{8}$$

where $\mathbf{W}$ denotes a $2f \times n$ matrix containing a set of 2D point correspondences with respect to the center of mass. $\mathbf{M}$ denotes a $3 \times n$ matrix containing affine

[3] http://www.vision.caltech.edu/bouguetj/calib_doc/

shape of the retinal surface. $\mathbf{P}$ denotes $2f \times 3$ matrix comprising f fundus camera model. With the rank theorem, $\mathbf{W}$ is at most rank three. Singular value decomposition (SVD) is used to factorized $\mathbf{W}$, therefore, $\mathbf{P}$ and $\mathbf{M}$ are the left and right eigenvectors corresponding to the three greatest eigenvalues.

5.2 Affine Bundle Adjustment

Bundle adjustment is an optimization process of refining a visual reconstruction to produce *jointly optimal* structure and viewing parameters [11] [12] [13]. In other words, all the parameters, structure and camera parameters, are optimized simultaneously. It is usually formulated as a nonlinear least square problem. For our case, we want to estimate and refine affine cameras $\breve{\mathbf{P}}$ and the affine retinal surface $\breve{\mathbf{M}}$ which can project expected correspondence points $\breve{m}$. We try to minimize the distance between projected points $\breve{m}$ and observed points m as:

$$\min_{\breve{P}_i, \breve{M}_j} \sum_{i=1}^{f} \sum_{j=1}^{n} \|\breve{m}(\breve{P}_i, \breve{M}_j) - m_{ij}\|^2. \tag{9}$$

Bundle adjustment can become an extremely large minimization problem because of a large number of parameters, rather than directly using the Levenberg-Marquardt algorithm, a sparse Levenberg-Marquardt algorithm is employed to reduce the computational cost. This is due to the fact that a specific residual is only dependent on one 3-D point and one camera which yields a very sparse structure defined in Equation (9).

5.3 Euclidean Reconstruction of Retinal Surface

To recover a retina's Euclidean surface from an affine surface, a 3×3 nonsingular matrix, $\mathbf{D}$ needs to be identified. From Equation (8), we get

$$\mathbf{W} = \mathbf{PDD}^{-1}\mathbf{M} = (\mathbf{PD})(\mathbf{D}^{-1}\mathbf{M}) = \acute{\mathbf{P}}\acute{\mathbf{M}}, \tag{10}$$

where D is called a metric constraint. Several different solutions for different affine camera projections were proposed. Tomasi and Kanade [10] proposed a solution for orthographic projection. Weinshall and Tomasi [14], [15] introduced a solution under weak-perspective camera. Poleman and Kanade [16] [17] proposed a solution for paraperspective projection. Quan [18], Kurata et.al. [19] attempted to congregate those solutions into one unified framework for general affine camera without having to calibrate the camera. If the images are assumed to be taken by the same affine camera, then the intrinsic parameters $\mathbf{K}$ are the same for every views. The following constraints are obtained.

$$\arg\min_{\mathbf{X}} \sum_{i=1}^{f-1} \left(\left(\frac{u_i^T \mathbf{X} u_i}{v_i^T \mathbf{X} v_i} - \frac{u_{i+1}^T \mathbf{X} u_{i+1}}{v_{i+1}^T \mathbf{X} v_{i+1}} \right)^2 + \left(\frac{u_i^T \mathbf{X} v_i}{v_i^T \mathbf{X} v_i} - \frac{u_{i+1}^T \mathbf{X} v_{i+1}}{v_{i+1}^T \mathbf{X} v_{i+1}} \right)^2 \right), \tag{11}$$

where $\mathbf{P}_i = \begin{bmatrix} u_i^T \\ v_i^T \end{bmatrix}$ and $\mathbf{X} = \mathbf{D}\mathbf{D}^T$. The minimization problem defined in (11) can be solved by any nonlinear optimization algorithms such as a Levenberg-Marquardt algorithm.

6 Point-Based Surface Approximation

We take into an account of an eyeball's geometric constraint in order to approximate the 3-D retinal surface. We assume that eyeball is an approximated sphere. We introduce a point-based sphere fitting method. The method is accomplished by first selecting a reference point $M_k = (X_k, Y_k, Z_k)$ from 3-D point cloud. Every point has to satisfy the sphere equation.

$$\begin{aligned} &(X_k - A)^2 + (Y_k - B)^2 + (Z_k - C)^2 = R^2 \\ &(X_j - A)^2 + (Y_j - B)^2 + (Z_j - C)^2 = R^2,\ j \in 1, ..., n, j \neq k, \end{aligned} \tag{12}$$

where (A, B, C) and R are sphere's center point and radius respectively. By subtracting the two equations and rearranging the terms, we get

$$\begin{aligned} (X_k^2 - X_j^2) + (Y_k^2 - Y_j^2) + (Z_k^2 - Z_j^2) = \\ 2(X_k - X_j)A + 2(Y_k - Y_j)B + 2(Z_k - Z_j)C, \\ j \in 1, ..., n, j \neq k. \end{aligned} \tag{13}$$

Equation (13) is in a linear format. Sphere's center point (A, B, C) can be obtained by solving multiple linear equations. Then, radius R can be computed in a least mean square sense. Ideally speaking, every point has to satisfy the sphere equation. An error at a particular point j is calculated by the following equation:

$$E_j^{(k)} = \|M_j^T \mathbf{Q}_k M_j\|, \tag{14}$$

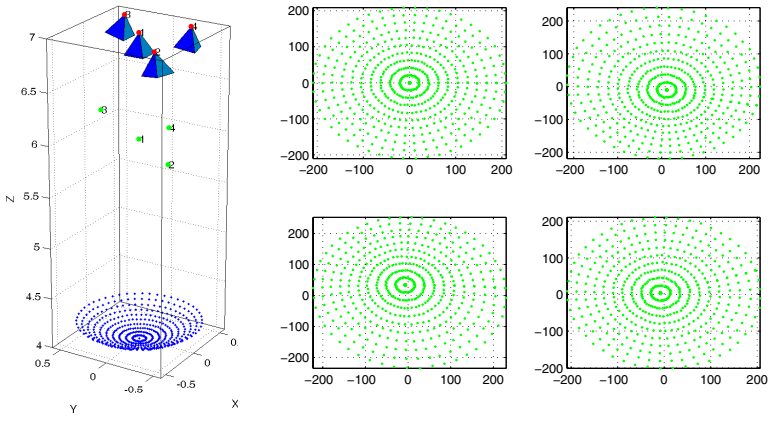
where $\mathbf{Q}_k$ is a 4×4 sphere matrix by a reference point k. $E_j^{(k)}$ represents an error at point j by using $\mathbf{Q}_k$. By minimizing the following equation, the optimal sphere surface with best fitness to all points can be achieved.

$$\hat{\mathbf{Q}} = \arg \min_{k \in 1,...,n} \sum_{j=1}^{n} E_j^{(k)}, \tag{15}$$

where $E_j^{(k)}$ is defined in Equation (14).

7 Experimental Results

We tested our proposed method on both synthetic data and real retinal images. We have generated a 3-D partial sphere point cloud with the spreading angle of 90^o, as shown in Fig. 5(a). Then, four virtual cameras are positioned according to the ETDRS imaging setting, as shown in Fig. 4(a). The four synthetic images captured by these cameras are shown in Fig. 4(b). In the following experiments, we added zero-mean and isotropic Gaussian noise of different levels



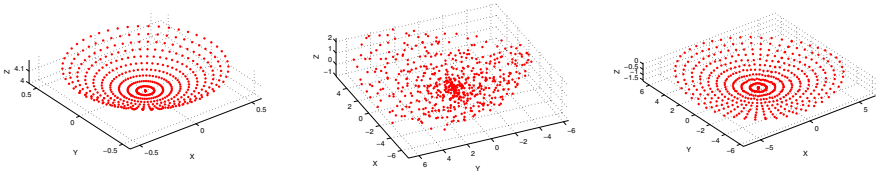
(a) (b)

Fig. 4. (a) The set up of four synthetic cameras. Point cloud is constructed on a spherical surface with spreading angle of 90^o. (b) Four images generated by the four cameras shown in (a) with zero mean Gaussian noise (variance = 0.5).

to correspondence measurements for algorithm evaluation. Four images are the minimum settings for Euclidean reconstruction using an affine camera .

7.1 Surface Approximation on Synthetic Data

Qualitative Analysis. The 3D affine reconstruction results on synthetic data are shown in Fig. 5 under noise variance 0.5. The initial affine shape is shown in Fig. 5(b) which is more like a quadratic surface instead of a sphere. This shape distortion is probably because the affine camera is only an approximation to the ideal projective camera. After affine bundle adjustment and Euclidean reconstruction, the reconstructed surface can be obtained as shown in Fig. 5(c) which is much closer to the original spherical surface. However, without affine bundle adjustment, the Euclidean reconstruction result still keeps its quadratic shape without significant improvement.



(a) (b) (c)

Fig. 5. (a) The original spherical shape. (b) Initial affine reconstruction result. (c) Euclidean reconstruction with bundle adjustment.

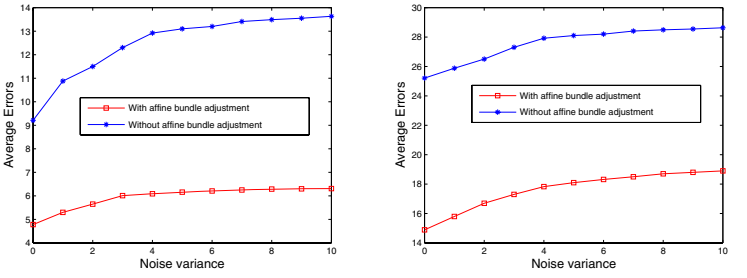


Fig. 6. Surface approximation errors v.s. noise variances (left) and spreading angle errors v.s noise variances (right). At each noise level, the algorithm is performed ten times to obtain the average error.

Quantitative Analysis w.r.t. Noise. Zero-mean, isotropic, Gaussian noises with different variances are added to image measures to test the robustness of affine bundle adjustment. Two numerical criteria are defined to evaluate the effectiveness of 3D surface reconstruction. We first define the surface approximation error according to Equation (16) as

$$F_k = \frac{1}{n}\sum_{j=1}^{n} E_j^{(k)}, \tag{16}$$

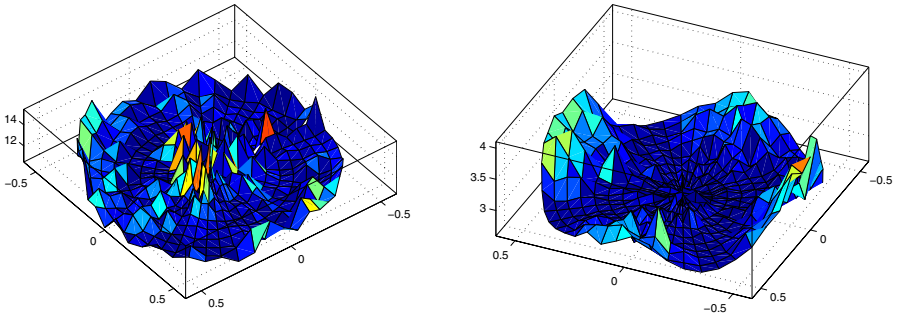
where $E_j^{(k)}$ is defined in Equation (14) and F_k represents the overall surface approximation error by picking initial point k. Additionally, we also define a spreading angle as the secondary measure as

$$\theta = 2\arcsin(\frac{r}{R}), \tag{17}$$

where R and r denote radii of approximated sphere and circle respectively. A sphere's radius R estimation is given in section 6. Fig. 6(a)(b) show the errors of surface approximation versus noise variances in terms of two criteria. It is shown that bundle adjustment does improve the reconstruction performance significantly and sustain good performance under strong noises.

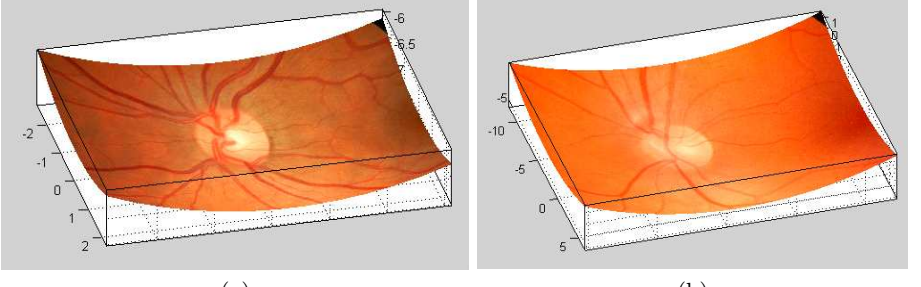
Quantitative Analysis w.r.t. Reference Point Selection. We plot the errors of reconstructed surface versus 2-D spatial locations of the reference point in order to understand the relationship between the location of reference point and the surface approximation error. Regardless of the noise level in images, if an affine bundle adjustment is not performed, the error plot always possesses a similar shape as shown in Fig. 7(a). We observe that points around an average depth yield the minimum error. If an affine bundle adjustment is involved, the error plot always retains a similar shape as shown in Fig. 7(b). Points around the bottom produce the minimum error. These observations implicitly convey useful information for selecting a good reference point. Given a quadratic surface (affine

reconstruction without affine bundle adjustment shown in Fig. 5(b)), the point-based surface approximation estimates a spherical surface along a quadratic's average depth. If the reconstructed shape is closer to a sphere (affine reconstruction with affine bundle adjustment shown in Fig. 5(c)), then the algorithm would produce a surface that matches the bottom of the point cloud.



(a) (b)

Fig. 7. The errors versus 2-D spatial locations of the reference point selection. (a) Without affine bundle adjustment. (b) Affine bundle adjustment is involved.



(a) (b)

Fig. 8. 3D reconstruction results with retinal images mapped onto sphere surfaces

7.2 Retinal Surface Approximation

Two sets of retinal images, illustrated in Fig. 3, were used in the experiment. Each row of Fig. 3 depicts a set of retinal images which includes two stereo pairs of field 1 and 2. Point correspondences, also shown in Fig. 3, have been automatically extracted by our previously proposed algorithm [4]. The experimental results of 3-D retinal surface reconstruction are shown in Fig. 8(a) and (b). The surface approximation error between a reconstructed 3D point and the approximated surface is calculated according to Equation (14). Fig. 9(a) shows the

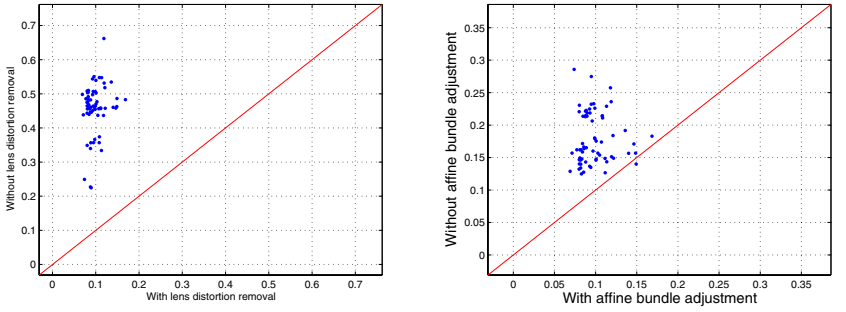


Fig. 9. The errors between reconstructed 3-D points and the approximated spherical surface without/with radial distortion removal (a) and without/with affine bundle adjustment (with the radial distortion removed) (b)

improvement due to lens distortion removal and Fig. 9(b) illustrates the further improvement from affine bundle adjustment. The best performance is achieved by removing the lens distortion and by using affine bundle adjustment.

8 Conclusions

We have showed 3D retinal surface reconstruction using an affine camera model for ETDRS retinal images. The robustness and effectiveness of the proposed algorithm are rooted in the linear nature of the affine camera and the prior knowledge about the shape of human retinal. Also, the reconstruction performance is significantly improved by lens distortion removal and affine bundle adjustment. In the future, we want to incorporate the geometrical constraint into affine bundle adjustment to further improve the reconstruction accuracy.

Acknowledgments

We thank Dr. Stephen R. Fransen (Inoveon Crop., Oklahoma City, OK) for providing the retinal image database. This work was supported by an OHRS award (HR03-33) from the Oklahoma Center for the Advancement of Science and Technology (OCAST). This work is also partially supported by the NSF under Grant IIS-0347613.

References

1. Deguchi, K., Kawamata, D., Mizutani, K., Hontani, H., Wakabayashi, K.: 3d fundus shape reconstruction and display from stereo fundus images. IEICE Trans. Inf. & Syst. **E83-D** (2000) 1408–1414

2. Deguchi, K., Noami, J., Hontani, H.: 3d fundus pattern reconstruction and display from multiple images. In: IEEE Int'l conference on Pattern Recognition. Volume 4. (2000) 94–97
3. Choe, T., Cohen, I., Medioni, G.: 3-d shape reconstruction of retinal fundus. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2006) 2277–2284
4. Chanwimaluang, T., Fan, G.: Hybrid retinal image registration. IEEE Trans. Information Technology in Biomedicine **10** (2006) 129–142
5. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: IEEE Int'l Conference on Computer Vision. Volume 1. (1999) 666–673
6. Koenderink, J.J., Doorn, A.J.V.: Affine structure from motion. Journal of Optical Society of America **8** (1991) 377–385
7. Quan, L., Mohr, R.: Towards structure from motion for linear features through reference points. In: IEEE Workshop on Visual Motion. (1991) 249–254
8. Demey, S., Zisserman, A., Beardsley, P.: Affine and projective structure from motion. Proc. British Machine Vision Conference (BMVC) (1992) 49–58
9. Shapiro, L.S.: Affine Analysis of Image Sequences. PhD thesis, Sharp Laboratories of Europe, Oxford, Oxford, UK (1995)
10. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization method. Intl. Journal of Computer Vision **9** (1992) 137–154
11. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: A modern synthesis. Vision Algorithms: Theory And Practice, Springer-Verlag (2000)
12. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision: Second Edition. Cambridge University Press (2003)
13. Tresadern, P., Reid, I.: Uncalibrated and unsynchronized human motion capture: a stereo factorization approach. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR. Volume 1. (2004) 128–134
14. Weinshall, D., Tomasi, C.: Linear and incremental acquisition of invariant shape models from image sequences. In: IEEE Proc. of 4th Int'l Conference on Computer Vision. Volume 17. (1993) 512–517
15. Weinshall, D., Tomasi, C.: Linear and incremental acquisition of invariant shape models from image sequences. IEEE Trans. pattern Anal. Machine Intell. PAMI **17** (1995) 512–517
16. Poleman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. In: Proc. of 3rd European Conference on Computer Vision. (1994) 97–108
17. Poleman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. IEEE Trans. Pattern and Machine Intelligent **19** (1997) 206–218
18. Quan, L.: Self-calibration of an affine camera from multiple views. International Journal of Computer Vision **19** (1996) 93–110
19. Kurata, T., Fujiki, J., Sakaue, K.: Affine epipolar geometry via factorization method. In: Proc. of 14th Int'l Conference on Pattern Recognition. Volume 1. (1998) 862–866

The Diagnostic Application of Brain Image Processing and Analysis System for Ischemic Stroke

Tzyh-Chyang Chang[1,2], Jiann-Der Lee[1,*], Chung-Hsien Huang[1], Tony Wu [3], Chi-Jen Chen[4], and Shwu-Jiuan Wu[5]

[1] Department of Electrical Engineering, Chang Gung University, Tao-Yuan, Taiwan
259 Weng Hwa 1st Rd., Kweishan, Taoyuan 333, Taiwan
{tcc200, jdlee}@mail.cgu.edu.tw,
d8921004@stmail.cgu.edu.tw
[2] Department of Occupational Therapy, Chang Gung University, Tao-Yuan, Taiwan
259 Weng Hwa 1st Rd., Kweishan, Taoyuan 333, Taiwan
tcc200@mail.cgu.edu.tw
[3] Department of Neurology, Chang-Gung Memorial Hospital, Taiwan
tonywu@adm.cgmh.org.tw
[4] Department of Radiation Diagnosis, E-DA Hospital, Taiwan
ed100975@edah.org.tw
[5] Taipei Veterans General Hospital, Taipei, Taiwan
201, Sec. 2, Shih-Pai Rd., Taipei 112, Taiwan
sjwu123@yahoo.com.tw

Abstract. The diffusion weighted image (DWI) technique is routinely used for diagnosis and treatment of early stroke due to its superior performance, especially when compared with conventional magnetic resonance image (MRI) for detection of acute ischemic stroke. Using DWI examination, this paper proposes an application of image processing in a computer-aided diagnosis system, which can effectively calculate the volume size and provide 3D reconstruction data of a lesion. The potential benefits of using our system include the higher accuracy of acute stroke lesion definition, the reduced time and procedure of calculating the volume, and providing 3D reconstruction image of stroke patients, which can effectively assist doctors in making more accurate diagnoses and treating patients in a more convenient way. Compared with the traditional method, the experimental results have shown the superior performance of this proposed system.

1 Introduction

The diffusion weighted image (DWI), one of the magnetic resonance image (MRI) techniques, is invaluable to define and show the initial infarct region of stroke and enable doctors to truly see the stroke. Compared with CT, MRI provides a better visualization of soft brain tissue. However, CT provides more accurate geometrical imaging. As far as DWI is concerned, it is more sensitive than conventional MRI for

* Corresponding author.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 31 – 38, 2006.
© Springer-Verlag Berlin Heidelberg 2006

detection of acute ischemia stroke. For this reason, it is routinely used for the diagnosis and treatment of an early stroke. After an acute ischemic stroke is diagnosed, we can give the patient suitable drugs to reduce the damage and improve the prognosis of the disease in the future. The effective medicine to treat patients within 3 hours of symptom onset of stroke is intravenous recombinant tissue plasminogen activator (rt-PA), which the American Food and Drug Administration (FDA) has already approved [1,2,3,4,5].

When the patient is delivered to the emergency room, he should be suspected to have an acute ischemic stroke, for there is no obvious internal hemorrhage in the brain, but the clinical symptom is obvious. If the patient is suspected to have the acute ischemic stroke within 3 hours of symptom onset, he would be carried and examined by the radiography techniques, including DWI, to determine whether there is any ischemic lesion. If necessary, doctors may use the magnetic resonance angiography technique (MRA) to appraise if the main cerebrum artery affects the lesion, or use the magnetic resonance spectroscopic imaging (MRSI) to determine the degree of the brain cell injury [3,6,7].

The American National Institute of Health Stroke Scale (NIHSS) is the common assessment of acute brain stroke diagnosis and treatment. For severe patients (NIHSS $\geqq$ 7) onset within 3 hours, DWI can demonstrate the exact lesion in the brain. Doctors then will give the patient the rt-PA in order to reduce the effect of the damage caused by a brain stroke. For the appropriate definition of the patient, it is restricted in the existing magnetism to inspire imagery function the radiography technology machine at present. According to the criterion of using rt-PA, doctors only can assess the patient with the lesion smaller than one-third of all brain area by the DWI estimates, but this limitation, in fact, is indefinite in volume or proportionality [3,7,8].

The purpose of this study is to design an image processing system that will enable doctors to calculate the lesion size and volume of a lesion from the DWI examination. In addition, we will also compare the volume calculation results from two different algorithms. Finally, we will provide clinical doctors the data of the patient's lesion and 3D reconstruction in order to make more accurate diagnoses and treating patients in a more convenient way.

2 Methods

We have already developed a multiple atlas-based environment for a computer-aided stereotactic functional neurosurgery software system, named brain deep tracer [9]. This system can help a neurosurgeon perform minimally invasive brain surgery and diagnosis brain disease. Generally speaking, this system includes 3D localization and registration of the patient's CT data, image fusion, and surgical trajectory planning of Parkinson's disease. In this study, we developed an integrative platform named Brain Image Analysis system based on the brain deep tracer. The Brain Image Processing and Analysis system can provide the accurate location of target tissue for doctors to reach the optimal treatment position. We find that the localization error can be

neglected when compared with other factors. In addition, the system not only provides the neurosurgeon with advanced and intuitive visualization facilities to observe the 3D structure of the target tissue before they perform the operation but also show exactly how it would appear in relation to the patient's anatomy. The Brain Image Processing and Analysis system also allows the neurosurgeon to determine the entry point and target [9].

In this study, we try to add some functions to the Brain Image Processing and Analysis system in order to improve the more accurate diagnosis of an acute stroke. In general, the overall procedure of study is as follows (Figure 1).

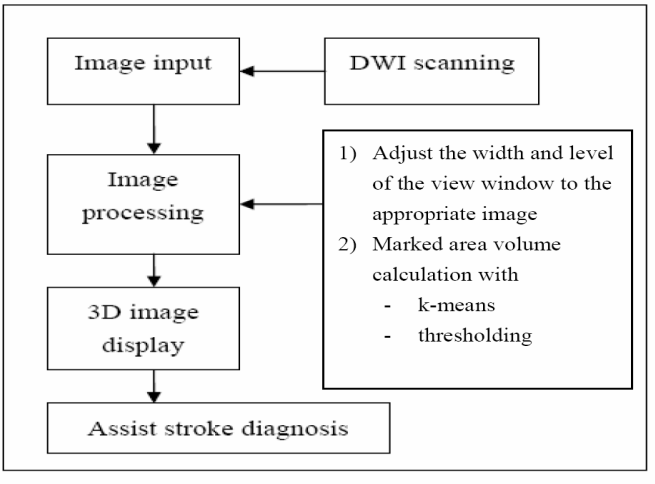


Fig. 1. The flowchart of the Brain Image Processing and Analysis system

1) When the patient is delivered to the emergency room, he is transferred to CT, MRI and DWI scanning, which can help doctors to acquire his brain image data. The DWI scanning machine used in this study is PHILIPS Medical System.
2) By using the Brain Image Processing and Analysis system, the patient's DWI data can be obtained and transferred for later use. We open patient's file and adjust the width and level of the view window to the appropriate image (from the doctor's view).
3) We apply the image techniques to process the patient's DWI data and then choose the area we are interested in. We may mark this special area and calculate the volume of the lesion.
4) We first use the Brain Image Processing and Analysis system to do not only k-means volume calculation but also thresholding volume calculation. At the same time, the source DWI image will also be shown as reference.
5) The 3D display of Brain Image Processing and Analysis system can be rotated and zoomed in real-time for viewing from any perspectives.

In this study, the patient diagnosed to have ischemia stroke is from Chang Gung Memorial Hospital in Taiwan. The details of experiments and results are illustrated as follows.

3 Results

In this study, we took a patient who met the diagnostic criteria of the acute stroke as the sample of this system. The patient was sent to the emergency room and assessed by the physical exam and radiological images. Then we loaded the patient's DWI image data to our advanced Brain Image Processing and Analysis system. In this system, we encouraged the neurosurgeon to use DWI images in order to increase the accuracy of localization. After DWI scanning, the actual patient's data, read as DICOM DWI images, were loaded into our system, and the axial, coronal, and sagittal views were generated after interpolation.

The results of the operation procedure and the comparison of the two volume algorithms were discussed as follows. Figure 2 shows the graphic user interface of the advanced Brain Image Processing and Analysis system.

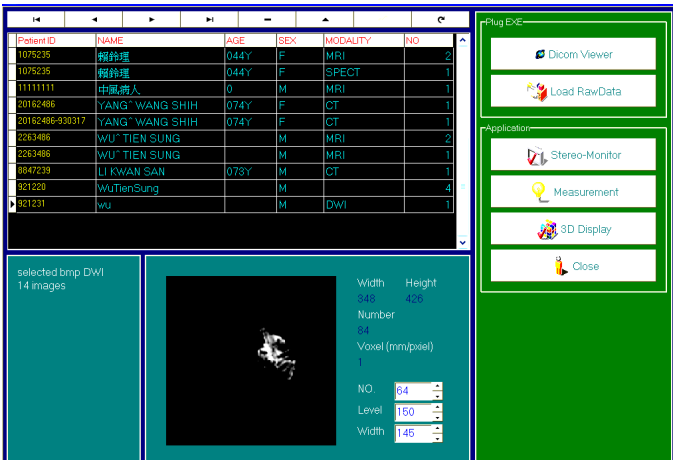


Fig. 2. Graphic user interface of the Brain Image Processing and Analysis system

After the patient's DWI images were collected, we loaded the data to the system. In measuring the data, we opened patient's file and adjust the width and level of the view window to the appropriate image (from the doctor's view). The results of the width and level adjustment comparison which is used to decide the best DWI lesion calculation are shown in table 1. Finally, we choose the level 150, width 145 and threshold 89 as the standard of the patient in this study because these values contribute to the best result (the result of lesion volume calculation is 383065 mm^3) near the real lesion size (about 350000-400000 mm^3).

Table 1. The comparison of width adn level adjustment

Test	Level	Width	Threshold	Volume (mm^3)
I	1	256	128	1046256
II	1	256	89	2178297
III	1	145	128	2381346
IV	1	145	89	2592305
V	150	256	128	0
VI	150	256	89	21359
VII	150	145	128	206838
VIII	150	145	89	383065

In the traditional method of lesion volume calculation, the doctor marksthe patient's lesion area on each DWI image and manually calculates the volume of the patient's lesion area. Using this method, we took considerable time calculating the lesion volume and easily made mistakes in calculation.

In this study, we proposed that the Brain Image Analysis system is the best way to calculate the lesion volume. We first mark the stroke lesion area with doctor's aid and use the k-means algorithm [9] of our proposed system to calculate the lesion volume (see figure 3).

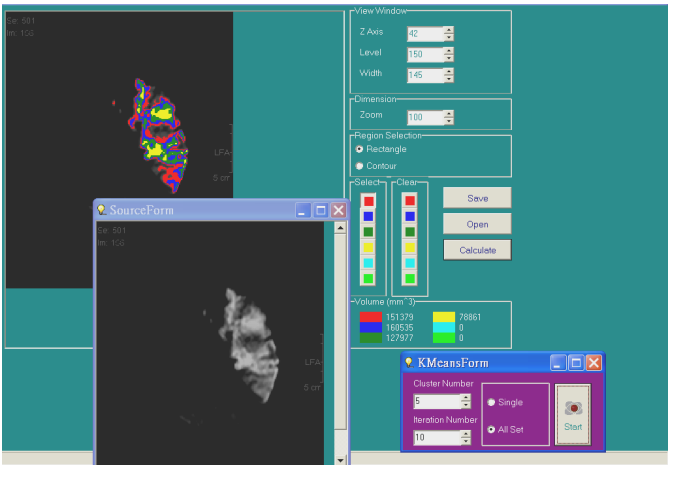


Fig. 3. The DWI volume calculation by k-means of the Brain Image Processing and Analysis system

Then, we use the thresholding algorithm [9] of our proposed system to calculate the lesion volume (see figure 4). In this study, we add the source image to this Brain Image Analysis system in order that we can make the original lesion and marked area consistency.

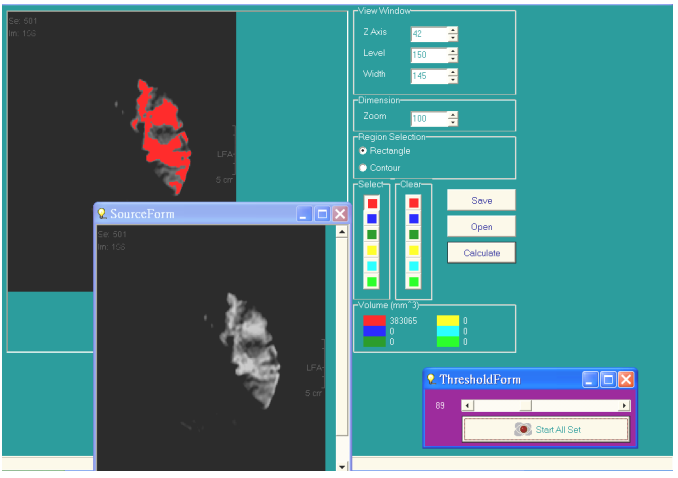


Fig. 4. The DWI volume calculation by thresholding of the Brain Image Processing and Analysis system

The results of the above two image process algorithm and the comparison between the traditional method and this proposed method (volume calculation with k-means or thresholding) are shown in table 2.

Table 2. The comparison of different methods of stroke leision volume calculation

Function Comparison	Traditional Method	Proposed method with k-means	Proposed method with Thresholding
Calculation	Manual	Computer	Computer
Volume (mm^3)	smaller than 1/3 all brain (350000 ~ 400000)	367373	383065
Time	84 min	24.8 sec	24.8 sec
Accuracy	Moderate	High	High
Speed	Slow	High	High
Mark Area	Hard	Easy	Easy
3D Display	No	Yes	Yes
Easy Operation	Average	Good	Good

In general, the results illustrated the accuracy of the proposed advanced system and provide atlas fusion on the patient's DWI data. The results from the advanced system can assist doctors in calculating the volume size of a lesion from the DWI examination. This Brain Image Processing and Analysis system can provide clinical doctors the data of the patient's lesion and 3D reconstruction in order to make more accurate diagnoses and treating patients in a more convenient way.

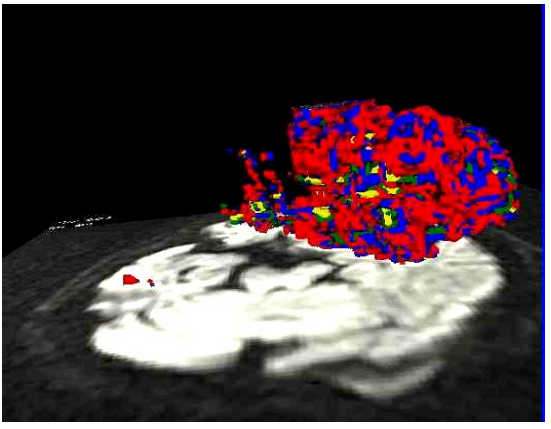


Fig. 5. The 3D image display of the lesion of the Brain Image Processing and Analysis system

4 Discussion

In emergency room, the correct diagnosis and the intervention time are very important. When the patient was suspected to have the ischemic stroke, we have to make the definite diagnosis and consider criteria of the rt-PA drug use as soon as possible. The volume of stroke lesion is the key point for us to consider if we use rt-PA. In this study, we apply the image process technique to aid the diagnosis of acute stroke. By using our advanced Brain Image Processing and Analysis system, we can calculate the volume of the ischemic lesion area immediately and decide whether to use rt-PA treatment or not. This method is more effective and accurate than the traditional one. If the patient was met the criteria of using rt-PA treatment after lesion volume calculation, he could get better recovery and reduce the side effect of lesion damage.

The limitation of this study is that we only have one patient assessed. In the future study, we will improve the graphic user interface and the speed of volume calculation algorithm of this system to let users operate it more easily. In addition, we plan to establish the database with more patients and we can compare different treatment prognoses.

5 Conclusion

In summary, the experimental results have shown the superior performance of this proposed system compared with the traditional method. The potential benefits of using our system include higher accuracy of acute stroke lesion definition, reduced time of the calculate lesion volume and procedure, and provided 3D reconstruction image of the stroke patient to assist doctors to make more accurate diagnoses and treating patients in a more convenient way.

References

1. Clay, M.T., Ferree, T.C.: Weighted Regularization in Electrical Impedance Tomography with Applications to Acute Cerebral Stroke. IEEE Trans. Medical Imaging, vol. 21, no. 6, (2002) 629-637
2. Schormann, T., Kraemer, M.: Voxel-guided Morphometry ("VGM") and Application to Stroke. IEEE Trans. Medical Imaging, vol. 22, no. 1, (2003) 62-74
3. Provenzale, J.M., Jahan, R., Naidich, T.P., Fox, A.J.: Assessment of the Patient with Hyperacute Stroke: Imaging and Therapy. Radiology, vol. 229, no. 2, (2002) 347-359
4. Beaulieu, C., Crespigny, A., Tong, D.C., Moseley, M.E., Albers, G.W., Marks, M.P.: Longitudinal Magnetic Resonance Imaging Study of Perfusion and Diffusion in Stroke: Evolution of Lesion Volume and Correlation with Clinical Outcome. Ann Neurol, vol. 46, (1999) 568-578
5. Latour, L.L., Warach, S.: Cerebral Spinal Fluid Contamination of the Measurement of the Apparent Diffusion Coefficient of Water in Acute Stroke. Magnetic Resonance in Medicine, Vol. 48, (2002) 478–486
6. Neumann-Haefelin, T., Moseley, M.E., Albers, G.W.: New Magnetic Resonance Imaging Methods for Cerebrovascular Disease: Emerging Clinical Applications. Ann Neurol, vol. 47, (2000) 559-570
7. Schellinger, P.D., Fiebach, J.B., Jansen, O. etc.: Stroke Magnetic Resonance Imaging within 6 Hours after Onset of Hyperacute Cerebral Ischemia. Ann Neurol, vol. 49, (2001) 460–469
8. Wintermark, M., Reichhart, M., Thiran, J.P., Maeder, P. etc.: Prognostic Accuracy of Cerebral Blood Flow Measurement by Perfusion Computed Tomography, at the Time of Emergency Room Admission, in Acute Stroke Patients. Ann Neurol, vol. 51, (2002) 417-432
9. Lee, J.D., Huang, C.H., Lee, S.T.: Improving Stereotactic Surgery Using 3-D Reconstruction. IEEE Engineering in Medicine and Biology, (2002) 109-116

Development of Early Tunnel Fire Detection Algorithm Using the Image Processing

Dongil Han and Byoungmoo Lee

Dept. of Computer Engineering
Sejong University 98 Kunja-dong, Kwagjin-gu, Seoul, 143-747, Korea
dihan@sejong.ac.kr, bmclass81@empal.com

Abstract. To avoid the large scale of damage of fire occurred in the tunnel, it is necessary to have a system to minimize and to discover the incident fast. However it is impossible to keep the human observation of CCTV in tunnel for 24 hour. So if the fire and smoke detection system through image processing warn fire state, it can be very convenient, and it can be possible to minimize damage even when people is not in front of monitor. In this paper, we proposed algorithm using the image processing, which is an early detection of the fire and smoke occurrence in the tunnel. The fire and smoke detection is different from the forest fire detection as there are elements such as car and tunnel lights and others that are different from the forest environment so that an indigenous algorithm has to be developed. The two algorithms proposed in this paper, are able to detect the exact position, at the earlier stay of detection. In addition, by comparing properties of each algorithm throughout experiment, we have proved the propriety of algorithm.

1 Introduction

To obtain the useable space of increasing metropolis area, development of underground space is necessary. Following its needs, tunnels are actively being developed for shorten of transportation time and efficiency of space. The needs of maintenance and management of the tunnel hence arose, due to longer and increased construction of the tunnel. Therefore, because of increased demand of the tunnel construction, the prevention of fire and ventilation of the tunnel became important matters to consider.

Especially when fire occurs in the tunnel, there will be mass injury of people and big damage to the properties. The detection system in early stage is required to reduce such a large amount of damage. We tried to minimize the damages by inventing the device which extinguishes fire as early as possible.

In case of forest fire detection, it was possible to detect the forest fire by distinguishing the color between the green forest and red fire [1], or by using the difference of previous field and gained images to detect the fast producing smoke [2]. In the forest fire perception system, it is generally a stationary image with no movement in each frame, and the tiny moving elements such as birds could be possibly ignored. Also when using the color information, it is easy to detect both green forest and red fire, even easier at night times.

However, when applying these kinds of algorithms in tunnel occasions, it is difficult to detect the fire due to elements such as car and tunnel lights and other various

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 39 – 48, 2006.
© Springer-Verlag Berlin Heidelberg 2006

situations. Therefore, the forest fire detection and the tunnel fire detection must be produced with independent algorithm development. In case of fire detection algorithm in tunnel, study of the differences between normal situation and the fire situation is compared by histogram[3], or a detection of entering cars and trains irregular overheat [4], or monitoring system of irregular situation in the tunnel[5] is being progressed. It's able to detect the smoke in buildings or other various circumstances using sensor [6]. But in the tunnel, there are chances of malfunction by automobile exhaust fume which don't exist in buildings or other various circumstances.

However, there are some problems such as inability of copping with fast moving cars in the tunnel, weak performance with the automobile exhaust fumes, and possibility of fault fire detection by car's back and front lights. As these problems lower the possible rate of fire detection in the tunnel, it causes many problems in making commercial product.

This paper describes the use of image obtained in the tunnel to produce early fire detection algorithm, and proposal of two algorithms which are complement with each other to solve problems as mentioned above. In sec. 2 it presents a brief review of the proposed algorithm and in sec. 3, experimental results based on the global feature and the local feature is shown.

2 Fire Detection Algorithms

Inside the tunnel, there are light and smoke factors such as car and tunnel lights and automobile exhaust fumes. As these factors may be the cause of fault fire detection, it is necessary to have more accurate and faster algorithm to distinguish above factors. The proposed algorithms in this paper are designed to give the alarm based on obtained images captured by cameras installed in the tunnel at regular intervals. The algorithms proposed in this paper are divided into two parts, Fire Detection algorithm and Smoke Detection algorithm. By experiments, each algorithm is proved to detect fire and smoke early when it occurs.

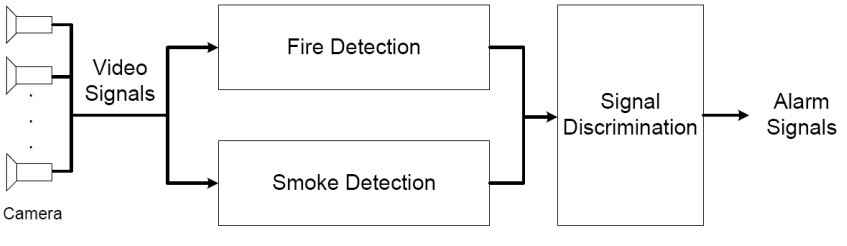


Fig. 1. The outline of algorithm

The outline of the proposed algorithm is shown in figure 1.

2.1 Fire Detection Algorithm

In this article, the fire detection algorithm used color image for effective detection of exist fire in image. Also A median filtering method is introduced for effective

removal of noise in images. When fire occurs in tunnel, the color information of image takes important role because fire itself makes a lot of light. However, car and tunnel lights have similar color information with those of fire; it is difficult to separate it from normal fire. On proposed fire detection algorithm, light of tunnel can be successfully removed with comparison of input images and stored image which is normal state of the tunnel. Car lights elimination was harder to do, due to it was created randomly. However, we were able to separate it using slope density function, since shapes of cars light are circular and the shape of fire is irregular.

The block diagram of the proposed algorithm of fire detection is shown in figure 2.

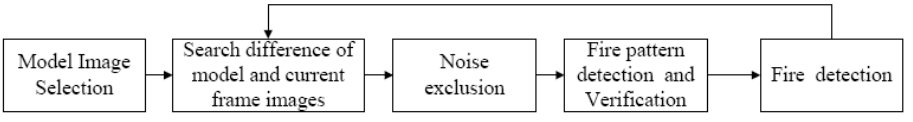


Fig. 2. The block diagram of the proposed fire detection algorithm

As shown in the block diagram, we have to select and store model image of normal state. This is very effective for removal of lights in the tunnel, also it make faster capacity since it eliminate needs of verification of lights during fire pattern verification. We make binary image that have pixel which is bigger than threshold of RGB value that is the color information of input image and stored model image, and we calculate differences between those input image and model image.

$$F(x,y)=\begin{cases}255 & R,G,B\geq T\\ 0 & otherwise\end{cases} \tag{1}$$

$$N(x,y)=\begin{cases}255 & R,G,B\geq T\\ 0 & otherwise\end{cases} \tag{2}$$

$$\begin{aligned}&D(x,y)=N(x,y)-F(x,y)\\ &if\ \ D(x,y)<0\ then\ D(x,y)=0\ else\ D(x,y)=D(x,y)\end{aligned} \tag{3}$$

Formula (1) and (2) is shown the binary image of the normal state model F(x, y) and the input image N(x, y), T means the threshold value. Formula (3) is shown the difference of obtained binary image through formula (1) and (2). Obtained binary image through formula (3) includes noises. These noises can be created because of waving light in tunnel or car lights. Figure 3 shows the differential of the input image (left) and the binary image with noise.

These noises are able to be removed by median filtering. And we were noticed that application of 7x7 window mask results best performance by experiment. Even after noise removal through the median filtering, such as forward light of cars still remains. As this case can be the cause of fault fire detection, it is necessary to remove with other method. We are able to separate the car lights using slope density function, since a shape of the cars light is circular and a shape of fire is irregular. This function represents simply a histogram of tangent-angle values. As a histogram is a measurement of

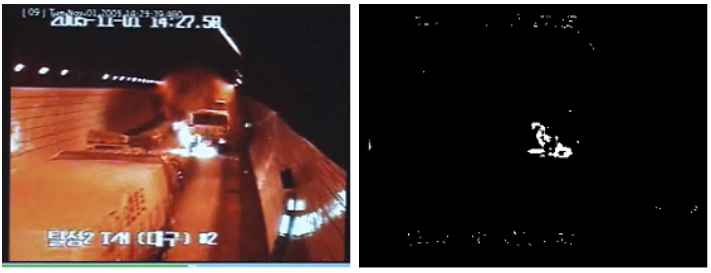


Fig. 3. The input image (left) and the binary image of difference image including noise

concentration of values, the slope density function responds strongly to sections of the boundary with constant tangent angles(straight or nearly straight segments) and has deep valleys in sections with sharp varying angles(corners of other sharp inflections). However the shape such as circle can be separated with fire, due to it has fixed change. We obtained region information about the binary image after such processing, and we are able to detect fire indicating obtained region at the input image.

2.2 Smoke Detection Algorithm

The smoke is very sensitive to lights. So the smoke detection is very difficult, since color information of the smoke is not simple. However, in this article, we approached with various image processing techniques such as moving detection, edge detection, and color information, resulting better performance. The block diagram of the proposed algorithm of smoke detection is shown in figure 4.

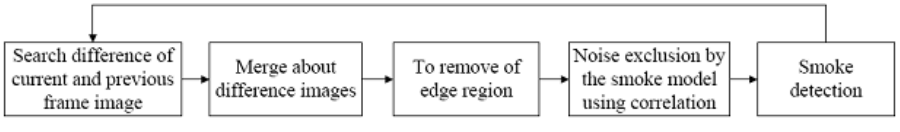


Fig. 4. The block diagram of the proposed smoke detection algorithm

The gray image is made out of nearby input images as many of N, and region of threshold on the gray image is removed. Such processes achieve the improvement of image processing speed by removing too bright or too dark region such as ceiling and light in tunnel. In formula (4), f(x, y) is an image transformed from input image to gray image. T1 and T2 means maximum and minimum value of threshold

$$G(x,y)=\begin{cases} F_{input}(x,y) & T_1 < F_{input}(x,y) < T_2 \\ 0 & otherwise \end{cases} \tag{4}$$

We obtained the absolute value of difference image from nearby image with obtained gray images as many as N. And we are able to detect motion by adding image to obtained different images as many as N-1. It has advantage of being able to remove region of fast moving objects by motion detection, since the motion of smoke is relatively slow against the motion of cars. Also it is applicable to object with few movements, too.

$$D_T(x, y) = | G_T(x, y) - G_{T-1}(x, y) | \qquad T = 1,2,..., N \tag{5}$$

$$A_{TN}(x, y) = \sum_{i=1}^{N} D_i \tag{6}$$

In formula (5), G(x, y) is processed input image of formula (4) and T show order of input images. Formula (6) is shown the sum of obtained difference image through formula (5). N of the two formula means number of input images. We used six input image for experiment. We used edge detection on region of N-th input image using the edge detection algorithm, since smoke doesn't have edge to be found by edge detection. Also we made the binary image on pixel has more values than threshold. Figure 5 is shown reverse image of obtained binary image which is out of edge detection using Sobel mask on N-th input image.

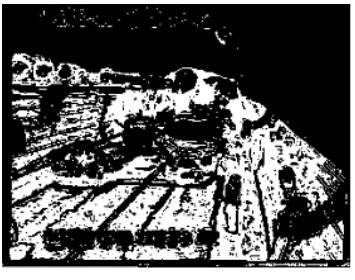


Fig. 5. Reversed binary image after edge detection on N-th input image

Also we are able to detect possible smoke region using the color information of N-th input image. This process have higher detection success ratio compare to simple selection of region using RGB values. Formula (7) is shown transformation of RGB value of each input image when it is within specified region. C(x, y) is output image, and T1~T6 is shown each threshold value of RGB value of the smoke region.

$$C(x, y) = \begin{cases} 255 & T_{1,3,5} < F_{R,G,B}(x, y) < T_{2,4,6} \\ 0 & otherwise 0 \qquad otherwise \end{cases} \tag{7}$$

We are able to detect smoke by comparison of obtained three images such as edge detection, color information, and motion detection image. When obtained information of the merged different images has a little of change, doesn't have edge region, and are within region of possible smoke area, it happens a little noise. We tried to remove this noise by median filtering, to detect more precise region. As the result, we are able to show the improvement of performance by experiment.

3 Experimental Results

We obtained nine hundreds of images using video data of real case of fire in tunnel. Also we made thirty composed images with normal state and fire state of it. We verified performance by various experiments. In case of smoke detection algorithm, we

only need continuous image as many as N so test was conducted with nine hundreds of images out of video data.

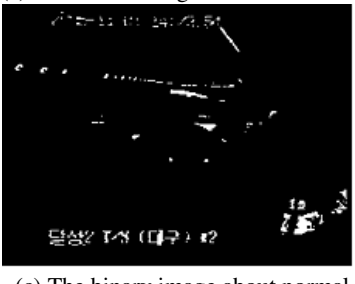
Figure 6 is samples of experiment step for the fire detection algorithm, (a) is stored image of normal state, and (b) is an input image. (c) and (d) selected binary image with bigger value than threshold of (a) and (b). (e) is difference image of two images of (c) and (d). (f) is an image after adopted filtering process to (e).



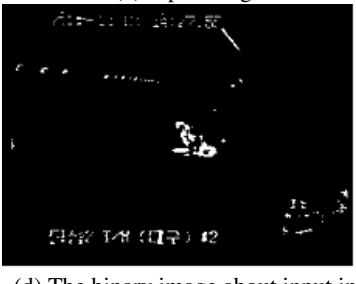
(a) The stored image of normal state



(b) Input image



(c) The binary image about normal state



(d) The binary image about input image



(e) The difference image before filtering

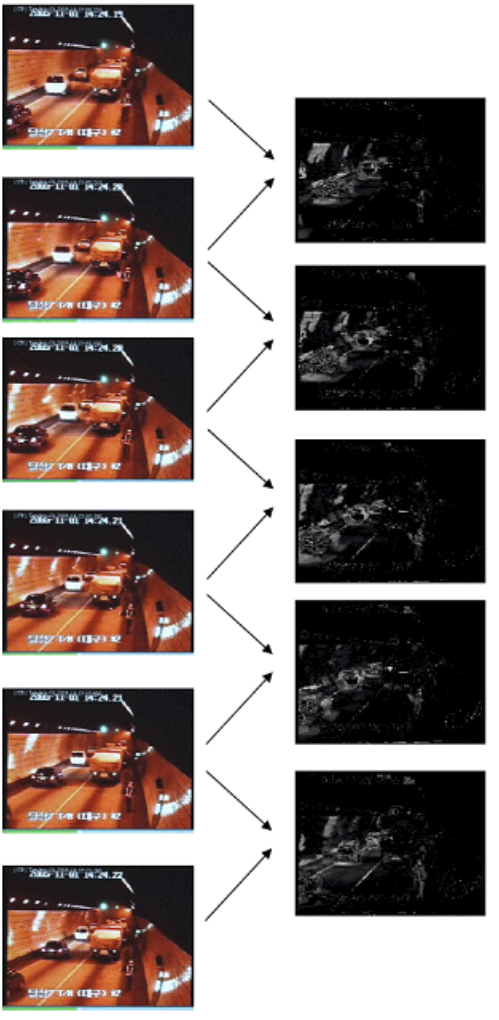


(f) The difference image after filtering



(g) The final fire detection image

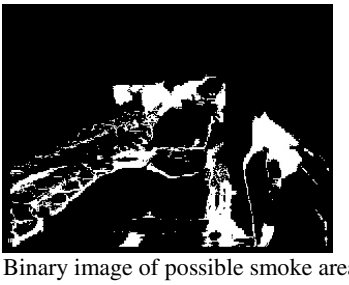
Fig. 6. Experiment step image of the fire detection algorithm



(a) Difference image of nearby 6 input images



(b) Merged image of 5 difference images



(c) Binary image of possible smoke area.

Fig. 7. Image of experimental steps of smoke detection algorithm

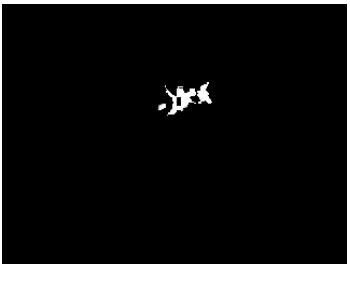
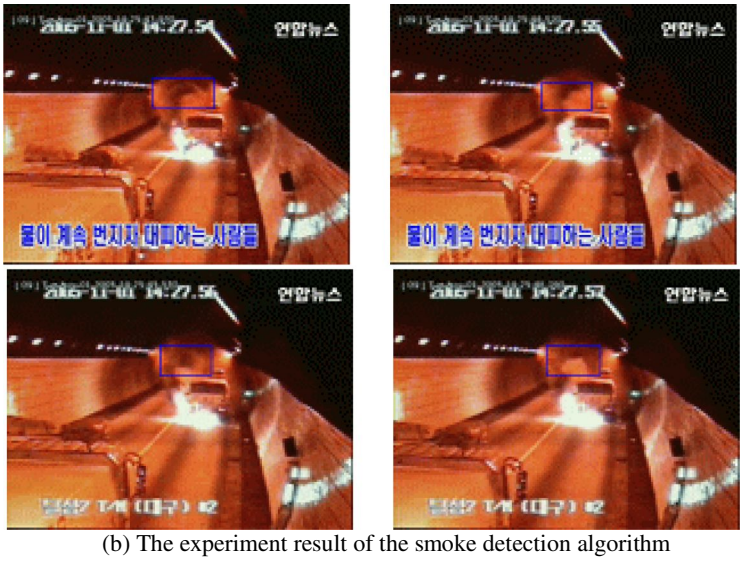


Fig. 8. Detected smoke region



(a) The experiment result of the fire detection algorithm



(b) The experiment result of the smoke detection algorithm

Fig. 9. The final result image of fire and smoke detection algorithms

Figure 7 is samples of experiment step for the smoke detection algorithm. (a) is a difference image of nearby input images, (b) is a binary image to show clearer image on specific value(with few change) of merged image of difference images which obtained through (a). (c) is a transformed binary image indicating possible region of smoke using the color information of recent input images. Figure 8 is shown to detected smoke region. Figure 9 is a final result image of fire and smoke detection algorithms. We can see (a) is possible to correctly detect most of fire, but (b) is failed to

detect full region of smoke since the smoke detection is sensitive to lights, yet it shows possibility of detection of parts of smoke area.

4 Conclusion

The human observation of CCTV in tunnel for 24 hours is very difficult task. So when the fire and smoke detection and warning system which is using applicable image processing is used, it makes fire detection more convenient and it is also able to minimize damage when human observation is not available. The proposed two algorithms are able to detect different situations. The fire detection algorithm detects fire by compared image of normal state and input image using color information, while the smoke detection algorithm detects smoke using motion detection, edge detection and comparison of color information of input images. And we verified its performance by experiments. The fire detection algorithm showed better performance since it is not sensitive to lights of car and tunnel compare to the smoke detection algorithm, which failed to detect full region of smoke. However it is possible to detect parts of smoke. For the later work, we will conduct a research of color information measurement regarding various possible smoke areas at smoke detection algorithm, also performance test and improvement have to be done by tests in real fire cases. Also we plan to provide ideas to make users more convenient by suggesting real-time fire detection monitoring system built for real-time movie.

Acknowledgments

This work is supported the Construction Technology Research Development funded by The Ministry of Construction and Transportation.

References

1. Thou-Ho Chen, Cheng-Liang Kao, Sju-Mo Chang,“An intelligent real-time fire-detect ion method based on video processing“ Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on 14-16 Oct. 2003 Page(s):104 - 111.
2. Cappellini V., Mattii. L., Mecocci A., “An intelligent system for automatic fire detection in forests” Image Processing and its Applications, 1989, Third International Conference on 18-20 Jul 1989 Page(s):563 - 570.
3. Noda S, Ueda K., “Fire detection in tunnels using an image processing method“ Vehicle Navigation and Information Systems Conference, 1994. Proceedings, 1994 31 Aug.-2 Sept. 1994 Page(s):57 - 62
4. Cigada A., Ruggieri D, Zappa E., “Road and railway tunnel fire hazard: a new measurement method for risk assessment and improvement of transit safety” Measurement Systems for Homeland Security, Contraband Detection and Personal Safety Workshop, 2005. (IMS 2005) Proceedings of the 2005 IEEE International Workshop on 29-30 March 2005 Page(s):89 - 94

5. Koga K, Inobe T, Namai T, Kaneko Y, “Integrated traffic flow monitoring system in a large-scale tunnel” Intelligent Transportation System, 1997. ITSC 97. IEEE Conference on 9-12 Nov. 1997 Page(s):165 – 170
6. Liyang Yu, Neng Wang; Xiaoqiao Meng; “Real-time forest fire detection with wireless sensor networks” Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on Volume 2, 23-26 Sept. 2005 Page(s):1214 - 1217

Simulation of Artificial Winds Using a Hardware Illumination Technique

Namkyung Lee[1], Nakhoon Baek[2,⋆], and Kwan Woo Ryu[1]

[1] Dept. of Computer Engineering, Kyungpook National Univ., Daegu 702-701, Korea
namklee@hotmail.com, kwryu@knu.ac.kr
[2] School of EECS, Kyungpook National Univ., Daegu 702-701, Korea
oceancru@gmail.com

Abstract. Real world winds can be classified into two categories: natural winds and artificial winds. While previous researchers have focused on natural winds, we also need to simulate artificial winds, which may be generated by electric fans, air conditioners, human mouths, etc. In this paper, we present a mathematical model for artificial winds, and also present an efficient method for calculating the forces applied on the objects under the influence of artificial winds. Our method uses a hardware illumination technique to finally calculate the artificial wind forces, and thus the simulation process is accelerated considerably. Hence our model achieves real-time display of visually plausible motions, and its implementation shows sufficient performance for real-time interactivity. Through combining our method with previous natural wind models, we can also expect to build up an integrated wind simulation system for computer graphics applications.

1 Introduction

Wind is one of familiar natural phenomena that we could find easily everywhere. Due to winds, windows rattle, flags fly, leaves flutter down, papers are scattered, etc. However, it is difficult to mathematically model these matters, since such natural phenomena have irregular shapes and change their shapes quickly. Additionally, we should face the difficulty in dealing with fluid currents [1].

In computer graphics, we have several research results on modeling winds. Most research efforts are, however, focused on expressing them as natural phenomena [1, 2, 3, 4, 5, 6, 7, 8]. We additionally need to model artificial winds, since we cannot ignore the artificial winds produced by electric fans, air conditioners, human mouths, and so forth for more realistic simulations.

All the previous works simulating natural winds are not appropriate for our model, since such models express winds as velocity fields affecting on the whole environment. However, artificial winds usually have a direction and affect only the limited areas in the whole environment. Therefore, we need to propose a new model which is more suitable and more efficient for representing artificial winds.

⋆ Corresponding author.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 49–58, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we present a model for artificially generated winds with direction, and by using this model we propose a method for efficiently calculating the forces applied on the objects under the influence of the artificial winds. We also show that the formula for determining wind forces is similar to the formula for calculating effects of spotlights which are point light sources emitting cone-shaped lights. Thus, we can calculate the force due to the wind by using the graphics hardware originally designed for simulating the spotlights, and can accelerate the simulation process considerably. Hence our method can be used to quickly simulate artificial winds even in real-time applications such as computer games, virtual environments, etc.

The rest of this paper is organized as follows. Previous works are summarized in Section 2. In Section 3.1, we describe our wind model. We present a formula for calculating forces due to the wind, and then show that this formula is similar to the formula for calculating effects of spotlights in Section 3.2. We present a method for finding objects under the influence of the wind in Section 3.3. Some implementation results are followed in Section 4. Finally, we conclude this paper in Section 5.

2 Previous Works

Several approaches for simulating winds have been focused on expressing them as natural phenomena. When wind blows, every object in the simulation environment is affected by the wind. Therefore, the natural wind has been expressed as velocity fields affecting on the whole environment.

Wejchart et al. [7] introduced an aerodynamics model to simulate the motion of leaves blown by wind. The linearized flow could be represented as the Laplace equation, and thus was mathematically identical to a static electric field. They superimposed these static fields to finally generate various wind motions. This model is appropriate for simulating fluttering objects such as leaves that are traveling along the flow, but inappropriate for generating natural wind motions since the field should be static at a specific time instance. Shinya and Fournier [5] used a simple stochastic wind model obtained from the structural engineering field. They employed stochastic processes and Fourier synthesis to derive a wind field in spatiotemporal frequency domain, and inverted the result to get a periodic space-time wind field. Stam and Fiume [6] decomposed the turbulent wind field into two components. The Kolmogorov spectrum was used to model the small-scale random vector field. Although this approach is useful in cases where turbulent details are required, there is little control over the details of the field and no way to satisfy anything other than periodic boundary conditions. Perbet and Cani [3] simulated the movement of grass in a wind field generated by combining procedural primitives with an additional time-varying stochastic component. However, all these methods used fixed flow functions, and thus lacked interaction with the user in the scene.

Computational fluid dynamics methods for solving the Navier-Stoke's equations have led to significant developments in the visual simulation of fluids. Stam [1] proposed an unconditionally stable fluid solver using semi-Lagrangian

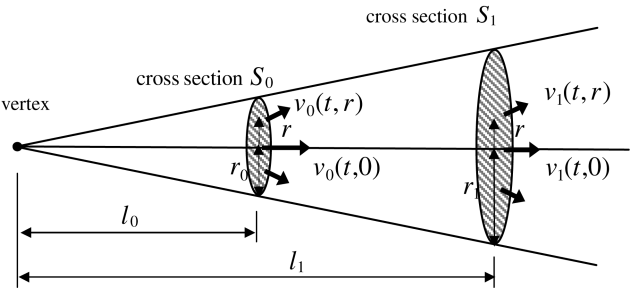


Fig. 1. The shape of the wind field. The artificial wind can be described to propagate in an infinite cone shape, and it would affect only on the objects located in the fields.

advection schemes and implicit solving method. We can achieve fast simulation without sacrificing visual plausibility by using this model. Losasso et al. [2] extended this Navier-Stoke solver from the regular grid structure to the more efficient octree structure. Wei et al. [8] presented an approach for modeling a wind field affecting on lightweight deformable objects. They modeled the wind field by using the Lattice Boltzmann Model from computational fluid dynamics. However, all these methods need heavy computation.

3 Artificial Wind Simulation

Our simulation of the artificial wind can be described as follows. In Section 3.1, we explain how to model the shape of the artificial wind field. Then we derive a formula for efficiently calculating the force applied on the object by the wind, and then show that this formula is similar to that for the illumination model in Section 3.2. Finally, we describe a method for deciding the objects which are under the influence of the artificial wind in Section 3.3.

3.1 Shape of the Wind Field

The artificial wind can be described to propagate in an infinite cone shape, as shown in Fig. 1. For example, electric fans usually make cone-shaped wind fields, and the wind would affect only on the objects located in the fields. A circular cross-section S_0 with radius r_0 plays the role of the wind source. The vertex of the cone is located at the distance l_0 from the center of S_0. User can provide r_0 and l_0 to control the shape of the cone, and the wind propagates from S_0 in the direction opposite to the vertex. User also controls the speed of the wind $v_0(t, 0)$ at S_0.

It is obvious that objects near the center line of the cone would be affected by the wind more than objects far from the center line. Hence, the speed of the wind near the center line can be considered to be faster than that far from the center line. This fact can be expressed by the cosine function as $v_0(t, r) = v_0(t, 0) \cos r$, where r, $0 \leq r < r_0$ is the distance from the center line of the cone, as shown in Fig. 1.

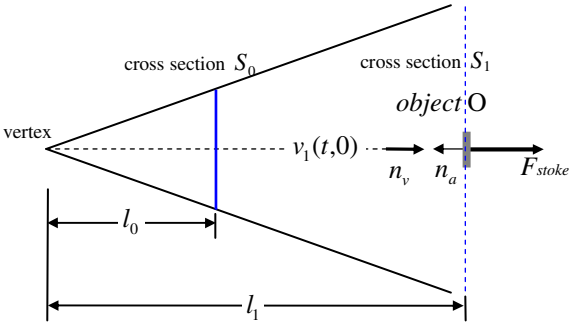


Fig. 2. Side view of the cone. The object located on the cross-section S_1 is affected by the wind force. The force is calculated by using the *Stoke's drag equation*.

3.2 Calculation of the Wind Force

In this section, we propose a formula for calculating the wind force, and then show that this formula is similar to the formula for intensity calculation of an illumination model. Since the intensity calculation feature is already implemented in the recent graphics hardware, we can calculate the wind force by using the hardware, and hence accelerate our simulation process remarkably.

We now show how to calculate the forces that the wind gives to objects within the cone. We begin by calculating the speed of the wind at distance $l_1 > l_0$ from the vertex. There are several research results [4,9] for doing this, and these results are from the aerodynamics for computer graphics applications. We choose a simple one out of them to achieve our goal of real-time display.

We assume that the fluid (in this case, air) is incompressible, and that no fluid can cross the boundary of the cone shape. This is a reasonable assumption for air at normal speed [7]. Then, the following equation follows from the Equation of Continuity [9] in fluid dynamics:

$$C_0 v_0(t,0) = C_1 v_1(t,0), \tag{1}$$

where C_i and $v_i(t,0)$ represent the area of the cross-section and the speed of the fluid at the cross-section S_i, respectively, for $i = 0, 1$. From Equation (1) and the geometric configuration of the cone, it is clear that

$$\frac{v_1(t,0)}{v_0(t,0)} = \frac{C_0}{C_1} = \frac{\pi r_0^2}{\pi r_1^2} = \frac{\pi l_0^2}{\pi l_1^2} = \frac{l_0^2}{l_1^2}. \tag{2}$$

Let an object O be located on the cross-section S_1 as shown in Fig. 2. Then the force acting on the face of the object O with area A can be calculated by using the *Stoke's drag equation* [9] as follows:

$$\boldsymbol{F}_{\text{stoke}} = \rho A v_1^2(t,0)(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v), \tag{3}$$

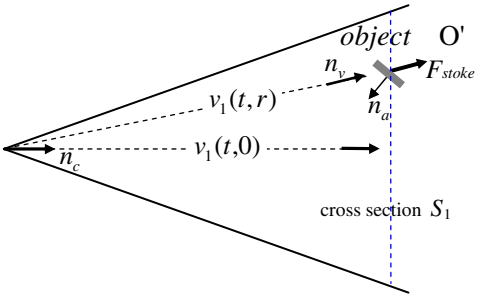


Fig. 3. Side view of the cone. The object O' located in arbitrary way on cross-section S_1. The wind force acting on O' can be calculated by using Equation (5).

where $\boldsymbol{n}_v$ and $\boldsymbol{n}_a$ are the unit directional vector of the wind and the normal vector of the face, respectively. The constant ρ is the density of the fluid. Equation (3) can be rewritten by using Equation (2) and the user-controllable term $v_0(t,0)$ as follows:

$$\boldsymbol{F}_{\text{stoke}} = \rho A \frac{v_0^2(t,0)}{l_1^4}(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v). \tag{4}$$

When the object is located at the center of the cross-section as shown in Fig. 2, the wind force is calculated by using Equation (4). When the other object O' is located in arbitrary way on the cross-section, as shown in Fig. 3, Equation (4) needs to be modified to calculate the wind force as follows:

$$\begin{aligned}\boldsymbol{F}_{\text{stoke}} = {} & \rho A \frac{v_0^2(t,r)}{l_1^4}(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v) \\ & \rho A \frac{v_0^2(t,0)\cos r}{l_1^4}(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v).\end{aligned} \tag{5}$$

It is clear that $\cos r = \boldsymbol{n}_c \cdot \boldsymbol{n}_v$, where $\boldsymbol{n}_c$ is the unit vector of the central axis of the cone. Hence, Equation (5) can be rewritten as

$$\begin{aligned}\boldsymbol{F}_{\text{stoke}} = {} & \rho A \frac{v_0^2(t,0)\cos r}{l_1^4}(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v) \\ & \alpha A v_0^2(t,0)(\boldsymbol{n}_c \cdot \boldsymbol{n}_v)(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v),\end{aligned} \tag{6}$$

where $\alpha = \rho / l_1^4$.

For simulating the turbulent behavior of the wind, we need to add a random noise term and hence the final force can be expressed as

$$\boldsymbol{F}_{\text{wind}} = \boldsymbol{F}_{\text{stoke}} + \boldsymbol{F}_{\text{random}}, \tag{7}$$

where the direction of $\boldsymbol{F}_{\text{random}}$ is selected by Perlin noise function [4], and bound by $|\boldsymbol{F}_{\text{random}}| < \beta |\boldsymbol{F}_{\text{stoke}}|$ for a user-definable constant β. After all the forces

are calculated, the position and the orientation of objects are updated by the physically based modeling method [10].

We now show that the previous formula is similar to the formula for the illumination model of spot light sources. We can write the spotlight illumination model as:

$$I = I_{\text{amb}} + A_{\text{att}}(-\boldsymbol{L} \cdot \boldsymbol{X})^e (I_d k_d (\boldsymbol{L} \cdot \boldsymbol{N}) + I_{\text{spec}}), \tag{8}$$

where I_{amb} and I_{spec} are the ambient and the specular lighting terms and I_d and k_d are diffusion constants, respectively [11, 12]. $\boldsymbol{L}$, $\boldsymbol{X}$ and $\boldsymbol{N}$ are the direction vector to the light source, the spot light cone axis vector, and the direction vector to the eye, respectively. The constant A_{att} is an attenuation term representing the attenuating effect of the light with distance. If we eliminate the ambient and the specular light terms, Equation (8) can be rewritten as

$$I = A_{\text{att}}(I_d k_d)(-\boldsymbol{L} \cdot \boldsymbol{X})^e (\boldsymbol{L} \cdot \boldsymbol{N}). \tag{9}$$

See Equation (6) and Equation (9). If the point light source of the spotlight and the vertex of the wind model are located at the same position, and if the central axis $\boldsymbol{X}$ of the spotlight cone and the unit vector $\boldsymbol{n}_c$ of central axis of the wind model are arranged in the same direction, then $(-\boldsymbol{L} \cdot \boldsymbol{X})$ corresponds to $(\boldsymbol{n}_c \cdot \boldsymbol{n}_v)$, and $(\boldsymbol{L} \cdot \boldsymbol{N})$ does to $(\boldsymbol{n}_v \cdot -\boldsymbol{n}_a)$. The constant terms of Equation (9) can be also related to those of Equation (6). By using these correspondences, we can rewrite Equation (6) as follows:

$$\boldsymbol{F}_{\text{stoke}} = \alpha A v_0^2(t, 0)(\boldsymbol{n}_c \cdot \boldsymbol{n}_v)(\boldsymbol{n}_v \cdot (-\boldsymbol{n}_a))(\boldsymbol{n}_v) = \kappa I \boldsymbol{n}_v, \tag{10}$$

where κ is a constant.

Thus, we can calculate the wind force through calculation of the spotlight intensity function. Since this calculation feature is already implemented in the graphics card hardware, we can accelerate our wind force calculation considerably by using this illumination technique.

3.3 Detection of Affected Objects

We now detect the objects that are affected by the wind. We do this by using a depth buffer method which is one of the visible surface detection methods [11,12]. Since this method uses graphics pipeline, we can speed up our simulation process a lot.

Before calculating the wind force on an object, we need to detect whether the wind is directly delivered to the object or not. When the object is occluded by another object in the air flow, we simply assume that the occluded object is not influenced by the wind. Without this assumption, it is hard to achieve the real-time display. In this simplified wind model, the air flow reaches only objects which are inside the cone and directly visible from the vertex of the cone.

If we implement our detection process with OpenGL [13], we can use the *gluProject* function that maps object coordinates to window coordinates to find window coordinates of objects. We also use the *glReadPixels* function to read the

intensity of pixels in the frame buffer for deciding whether the object is inside the cone and is not occluded by other objects.

To simulate partially occluded cases, objects are first partitioned into small areas on which sampling points are assigned. Then, we record the information such as positions and indices of the sampling points for simulation. The graphics pipeline is used to capture the image containing the cross-section S_0 with the synthetic camera located at the vertex of cone. And then we decide whether the sampling points are in the cone of the wind by using the *gluProject* function and the *glReadPixels* function. If the intensity of a sampling point is bigger than the threshold, the sampling point is determined to be affected by the wind, and we can calculate the force of wind by using Equation (10).

4 Implementation Results

Our method is implemented in C++ using OpenGL, and is tested on a personal computer with Intel Pentium IV CPU 2.3GHz and 2G Byte main memory. Our system has a GeForce FX5700 based graphic card with 128M video memory, and runs on Windows XP operating system. Since we speed up our calculation by using the illumination technique, the time for computing the animation is insignificant in comparison with the rendering time.

Fig. 4 shows the toy pinwheels affected by winds. The left column of Fig. 4 is the front view, while the right column is the side view of the same frame. In Fig. 4, the source of our wind model is located at the light source of the spotlight, and the central axis of the wind model and that of the spotlight cone are arranged in the same direction. Hence brighter areas in the scene are the positions that are more influenced by the wind as described in Section 3.2. For example, in the simulation of Fig. 4, the toy pinwheel located at the center of the bottom row is the brightest one, and hence it is the most rapidly rotating one since it is influenced most by the wind.

Fig. 5 shows our simulation of 1,000 toy pinwheels. The toy pinwheels located outside of the boundary of the wind model are not affected by the wind, and hence their fans do not move.

The images in Fig. 6 show pieces of paper scattered by the wind. Papers near the center axis of the wind are blown away farther than those far from the axis. Our simulations of Fig. 4 and Fig. 6 run faster than 60 frames per second and that of Fig. 5 shows more than 30 frames per second. Hence our system allows real-time visualization and interaction.

5 Conclusions

There are two kinds of winds in the real world: natural winds and artificial winds, and researches in computer graphics have been focused on modeling natural winds. However, these research results are not appropriate for simulating artificial winds. and thus, we need a new model which is more suitable and more efficient for artificial winds

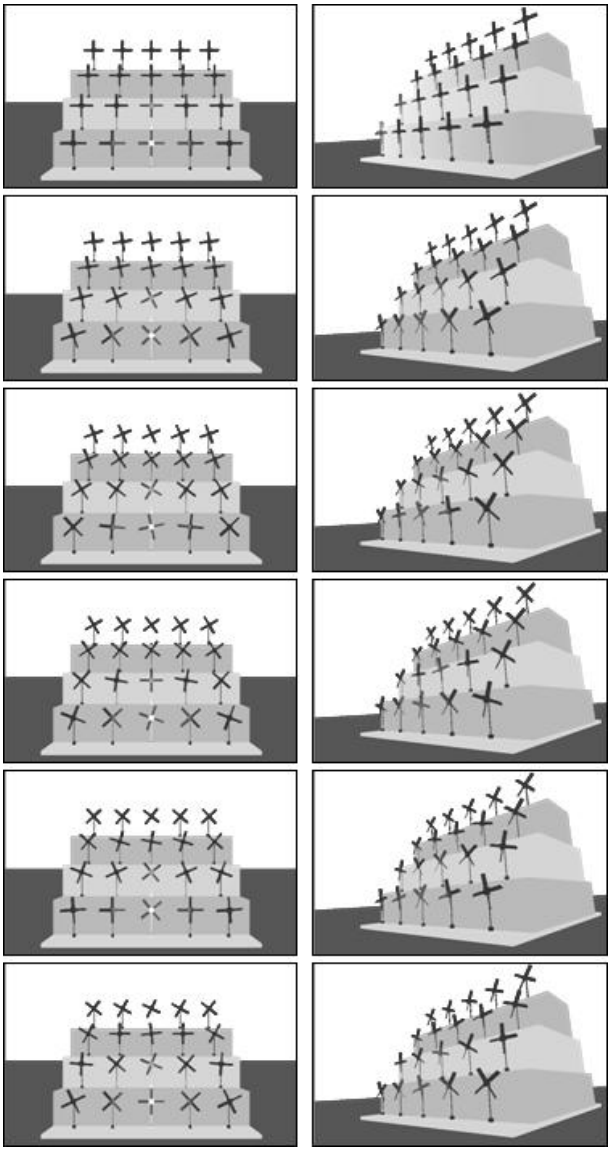


Fig. 4. The toy pinwheels affected by wind. The toy pinwheel located in the center of the bottom row is the brightest one, and hence it is the most rapidly rotating one because it is influenced most by the wind.

In this paper, we present an efficient model for artificial winds. We first setup the shape of winds, and then we present a method for efficiently calculating the forces applied to the objects affected by the artificial wind. We also show that our method is similar to the method for calculating the intensity of illumination

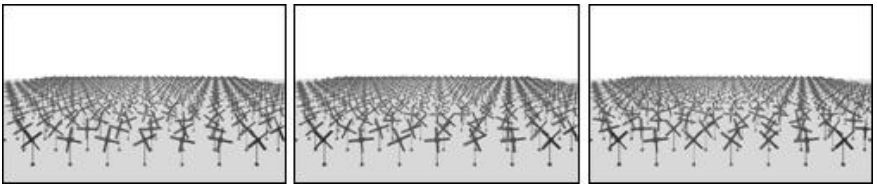


Fig. 5. Simulation of 1,000 toy pinwheels. It runs more than 30 frames per second.

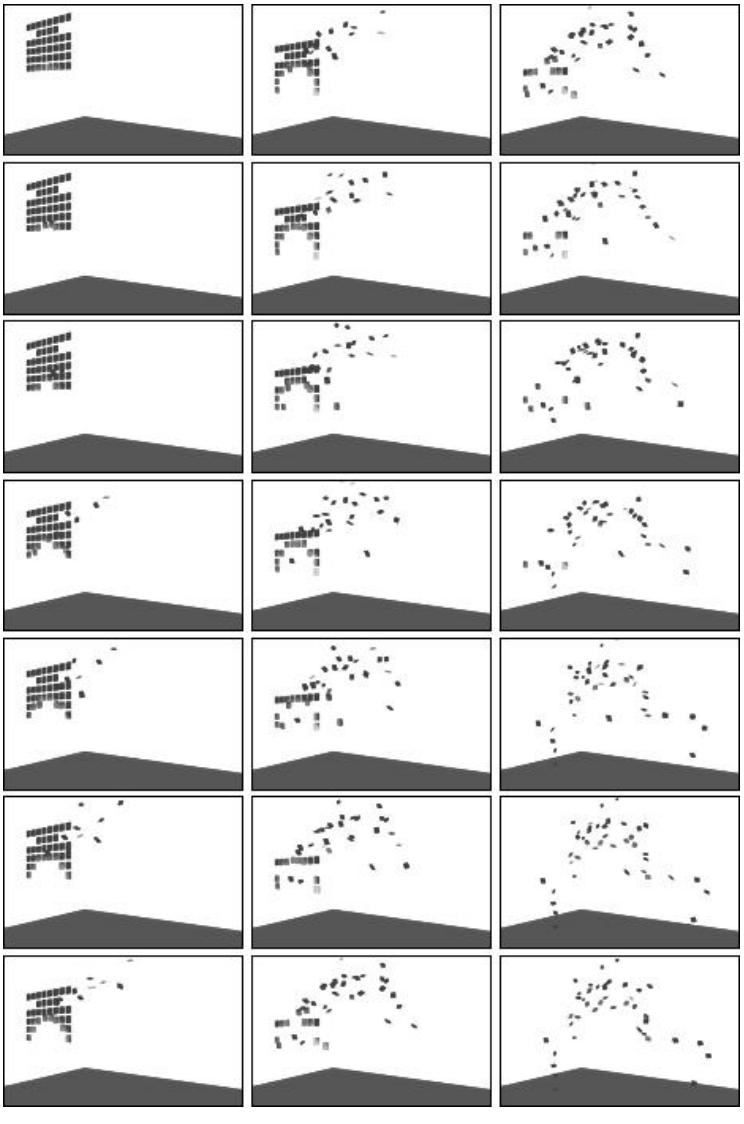


Fig. 6. Pieces of paper scattered by the wind. Papers near the center axis of the wind are blown away farther than those far from the axis.

model. Finally, we present a method for determining objects directly affected by the wind.

Our methods can take advantage of the graphics hardware. We read pixel data from the frame buffer and process our methods by using graphics hardware at a time. Thus the computation time is reduced a lot, and hence our model can be used in real-time applications such as computer games and virtual environments.

There can be many extensions to be done to our results. Our model can be improved by integrating the previous natural wind models. A user interface utilizing an artificial wind can also be developed by our method, and it can be used in the field of virtual reality.

Acknowledgements

This work was partially supported by *High Definition Game Research Team*, ETRI(Electronics and Telecommunications Research Institute), Korea.

References

1. Stam, J.: Stable fluids. SIGGRAPH '99 (1999) 121–128
2. Losasso, F., Gibou, F., Fedkiw, R.: Simulating water and smoke with an octree data structure. SIGGRAPH '04 (2004) 457–462
3. Perbet, F., Cani, M.P.: Animating prairies in real-time. ACM Symposium on Interactive 3D Graphics (2001) 103–110
4. Perlin, K.: An image synthesizer. SIGGRAPH '85 (1985) 287–296
5. Shinya, M., Fournier, A.: Stochastic motion – motion under the influence of wind. Eurographics '92 (1992) 119–128
6. Stam, J., Fiume, E.: Turbulent wind fields for gaseous phenomena. SIGGRAPH '93 (1993) 369–376
7. Wejchert, J., Haumann, D.: Animation aerodynamics. SIGGRAPH '91 (1991) 19–22
8. Wei, X., Zhao, Y., Fan, Z., Li, W., Yoakumstover, S., Kaufman, A.: Blowing in the wind. ACM SIGGRAPH/EUROGRAPHICS Symposium on Computer Animation (2003) 75–85
9. Streeter, V.L., Benjamin, E.: Fluid Mechanics. McGraw-Hill (1998)
10. Witkin, A., Baraff, D.: SIGGRAPH 2001 Course Notes on Physically-Based Modeling. ACM SIGGRAPH (2001)
11. Foley, J.D., van Dam, A.S., Feiner, K., Hughes, J.F.: Computer Graphics: Principles and Practice. 2nd edn. Addison Wesley (1990)
12. Hearn, D., Baker, M.P.: Computer Graphics. Prentice-Hall (1997)
13. Neider, J., Davis, T., Woo, M.: OpenGL Programming Guide: The Official Guide to Learning OpenGL. Addison-Wesley (1993)

Learning for Multi-view 3D Tracking in the Context of Particle Filters

Juergen Gall[1], Bodo Rosenhahn[1], Thomas Brox[2], and Hans-Peter Seidel[1]

[1] Max-Planck Institute for Computer Science
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{jgall, rosenhahn, hpseidel}@mpi-sb.mpg.de
[2] CVPR Group, Department of Computer Science, University of Bonn
Römerstr. 164, 53113 Bonn, Germany
brox@cs.uni-bonn.de

Abstract. In this paper we present an approach to use prior knowledge in the particle filter framework for 3D tracking, i.e. estimating the state parameters such as joint angles of a 3D object. The probability of the object's states, including correlations between the state parameters, is learned a priori from training samples. We introduce a framework that integrates this knowledge into the family of particle filters and particularly into the annealed particle filter scheme. Furthermore, we show that the annealed particle filter also works with a variational model for level set based image segmentation that does not rely on background subtraction and, hence, does not depend on a static background. In our experiments, we use a four camera set-up for tracking the lower part of a human body by a kinematic model with 18 degrees of freedom. We demonstrate the increased accuracy due to the prior knowledge and the robustness of our approach to image distortions. Finally, we compare the results of our multi-view tracking system quantitatively to the outcome of an industrial marker based tracking system.

1 Introduction

Model-based 3D tracking means to estimate the pose of a 3D object where the pose is determined by a value in a state space E. In the case of an articulated model of a human body, the pose is completely described by a 3D rigid body motion that has 6 degrees of freedom and the joint angles, which are 12 in this paper. This yields a high-dimensional state space that makes the tracking process difficult. Particle filters [1], however, can deal with high dimensions. A basic particle filter termed condensation has been used for contour tracking [2]. However, this algorithm lacks performance for 3D tracking. A heuristic that is based on these filters and that was successfully used for multi-view 3D tracking is the annealed particle filter (APF) [3]. In contrast to conventional particle filters, this method does not estimate the posterior distribution. Instead it performs a stochastic search for the global maximum of a weighting function. The two main drawbacks of the APF as applied in [3] are the simplified, unconstrained

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 59–69, 2006.
© Springer-Verlag Berlin Heidelberg 2006

kinematic model that results in a large number of particles needed for tracking and the assumption of a static background. The present paper addresses the first one by considering correlations between the state parameters as a soft constraint where the correlations are learned a priori. Using a level set based segmentation instead of background subtraction lifts the second assumption.

The idea to improve the model for 3D tracking by integrating prior knowledge is not new. In [4], training data acquired with a commercial motion capture system was used to learn a dynamical motion model (e.g. walking). This stabilizes the tracking as long as the assumptions are valid, but otherwise it is misleading and results in tracking failures. Hence, a large motion database is needed to learn more complicated motion models [5]. Hard constraints were also introduced for the 3D model such as anatomical joint angle limits and prevention of self-intersections [6]. This reduces the state space, but it does not consider the probability of different poses. In [7], it was suggested to learn a Gaussian mixture in a state space with reduced dimension, whereas the work in [8] captures the training data by a nonparametric Parzen density estimator. Our approach embarks on this latter strategy.

In previous works, a variational model for level set based image segmentation incorporating color and texture [9] has already been successfully used for pose estimation [10]. It is not based on background subtraction and, thus, does not necessarily need a static background. We combine this method with the APF to make the algorithm more flexible for applications.

The paper is organized as follows. We begin with a brief outline of the fundamental techniques, namely the APF and the variational model for image segmentation. Afterwards, in Section 3, we present the probabilistic model and motivate its choice. Furthermore, a full integration into a Bayesian framework is derived. Section 4 combines the prior knowledge with the methods from Section 2 and applies it to multi-view 3D tracking. The effect of the learned prior is demonstrated in Section 4. For our experiments we use a four camera set-up for tracking the lower part of a human body. Our articulated model consists of 18 degrees of freedom, and we will report on the robustness in the presence of occlusions and noise. Finally, we compare the results of our multi-view tracking system with a marker based tracking system. This provides a quantitative error measure. The paper ends with a brief summary.

2 Previous Work

2.1 Annealed Particle Filter

The APF does not approximate a distribution, usually the posterior distribution, like other particle filters [11]. Instead it performs a stochastic search for the global minimum of an "energy" function $V \geq 0$ by using n particles that are random variables in the state space. In accordance with simulated annealing [12], the

weighting function is a Boltzmann-Gibbs measure that is defined in terms of V and an inverse "temperature" $\beta > 0$ by

$$g(x)^{\beta}\,\lambda(dx) := \frac{1}{Z}\exp\left(-\beta V(x)\right)\,\lambda(dx), \tag{1}$$

where λ is the Lebesgue measure and $Z := \int \exp(-\beta V)\,d\lambda$. These measures have the property that the probability mass concentrates at the global minimum of V as $\beta \to \infty$. For avoiding that the particles are misguided by a local minimum, an annealing scheme $0 < \beta_M < \ldots < \beta_0$ is used. It provokes that the particles are weighted by smoothed versions of the weighting function with β_0 where the influence of the local minima is first reduced but then increases gradually as depicted in Figure 1. After the particles are initialized in accordance with an

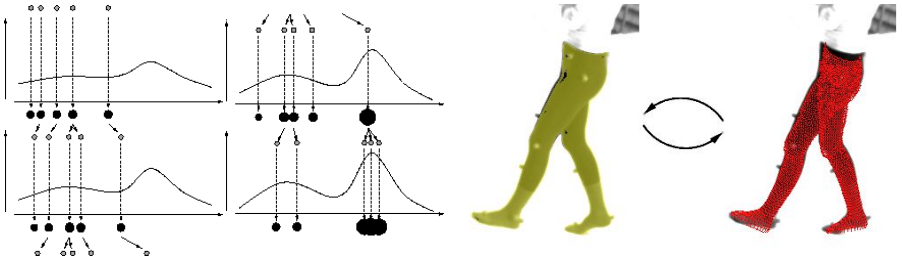


Fig. 1. Left: Illustration of the annealing effect with three runs. After weighting the particles (*black circles*), the particles are resampled and diffused (*gray circles*). Due to annealing, the particles migrate towards the global maximum without getting stuck in the local maximum. **Right:** The pose estimate (*right*) is obtained by weighting the particles according to the segmentation result (*left*). In return the pose result is used as shape prior for the segmentation of the next frame.

initial distribution, the APF with M annealing runs consists of a prediction step and an update step:

Prediction: Sample $\tilde{x}^{(i)}_{t+1,M}$ from $p(x_{t+1}|x^{(i)}_{t,0})\,\lambda(dx_{t+1})$
Update: For m from M to 0:

- Calculate weight $\pi^{(i)} = g(\tilde{x}^{(i)}_{t+1,m})^{\beta_m}$ and normalize weights to $\sum_i \pi^{(i)} = 1$.
- Generate $x^{(i)}_{t+1,m}$ by resampling with replacement, where $\tilde{x}^{(j)}_{t+1,m}$ is selected with probability $\pi^{(j)}$.
- Diffuse particles.

In the last run $m = 0$, the pose at time $t+1$ is estimated by $\widehat{x}_{t+1} = \sum_i \pi^i\,\tilde{x}^{(i)}_{t+1,0}$, and the particles are not diffused.

2.2 Variational Model for Segmentation

Level set based segmentation for r views splits the image domain Ω^i of each view into two regions Ω^i_1 and Ω^i_2 by level set functions $\Phi^i : \Omega^i \to \mathbb{R}$, such

that $\Phi^i(x) > 0$ if $x \in \Omega_1^i$ and $\Phi^i(x) < 0$ if $x \in \Omega_2^i$. The contour of an object is thus represented by the zero-level line. The approach described in [13] uses a variational model that integrates the contour of a prior pose $\Phi_0^i(\widehat{x}\,)$ for each view i. It minimizes the energy functional $E(\widehat{x}, \Phi^1, \ldots, \Phi^r) = \sum_{i=1}^{r} E(\widehat{x}, \Phi^i)$ where

$$\begin{aligned} E(\widehat{x}, \Phi^i) = &- \int_{\Omega^i} H(\Phi^i) \ln p_1^i + (1 - H(\Phi^i)) \ln p_2^i \, dx \\ &+ \nu \int_{\Omega^i} \left| \nabla H(\Phi^i) \right| \, dx + \lambda \int_{\Omega^i} \left(\Phi^i - \Phi_0^i(\widehat{x}\,) \right)^2 \, dx \end{aligned} \tag{2}$$

and H is a regularized version of the step function.

Minimizing the first term corresponds to maximizing the a-posteriori probability of all pixel assignments given the probability densities p_1^i and p_2^i of Ω_1^i and Ω_2^i, respectively. These densities are modeled by local Gaussian densities. The second term minimizes the length of the contour and smoothes the resulting contour. The last one penalizes the discrepancy to the shape prior. The relative influence of the three terms is controlled by the constant weighting parameters $\nu \geq 0$ and $\lambda \geq 0$. The interaction between segmentation with shape prior and the APF is illustrated in Figure 1. It has been shown that this method is robust in the case of a non-static background and that it is also able to deal with clutter, shadows, reflections, and noise [13].

3 Prior Knowledge in the Bayesian Framework

In the Bayesian framework, the particles are first predicted according to the transition density $p(x_{t+1}|x_t)$ and then updated by the likelihood $p(y_{t+1}|x_{t+1})$, where y_t is the observation at time t. The transition density, denoted by p_{pred}, is often modeled as zero-mean Gaussian since an accurate model is not available. This weak model does not include prior knowledge in an appropriate way. Since a precise model of the dynamics is not available for many applications, we combine the simple dynamical model p_{pred} with the probability density of the resulting pose p_{pose} that leads to a new transition density

$$p(x_{t+1}|x_t) := \frac{1}{Z(x_t)} \, p_{pred}(x_{t+1}|x_t) \, p_{pose}(x_{t+1}), \tag{3}$$

where $Z(x_t) := \int p_{pred}(x_{t+1}|x_t) \, p_{pose}(x_{t+1}) \, dx_{t+1}$. As it is often expensive to sample from the corresponding distribution, we show that it is possible to integrate p_{pose} in the update step. Following the basic notations of [1, p. 6], we obtain

$$p_\star(x_{t+1}|y_0, \ldots, y_t) := \int \frac{1}{Z(x_t)} \, p_{pred}(x_{t+1}|x_t) \, p(x_t|y_0, \ldots, y_t) \, dx_t, \tag{4}$$

$$p(x_{t+1}|y_0, \ldots, y_{t+1}) = \frac{p(y_{t+1}|x_{t+1}) \, p_{pose}(x_{t+1}) \, p_\star(x_{t+1}|y_0, \ldots, y_t)}{\int p(y_{t+1}|x_{t+1}) \, p_{pose}(x_{t+1}) \, p_\star(x_{t+1}|y_0, \ldots, y_t) \, dx_{t+1}} \tag{5}$$

where Equation (4) describes the prediction step and Equation (5) the update step. It is obvious that $p_\star$ is a density function, but not a probability density function, satisfying $p(x_{t+1}|y_0,\ldots,y_t) = p_{pose}(x_{t+1})\, p_\star(x_{t+1}|y_0,\ldots,y_t)$. Note that sampling from the distribution $p_{pred}(x_{t+1}|x_t)/Z(x_t)\,\lambda(dx_{t+1})$ is equivalent to sample from $p_{pred}(x_{t+1}|x_t)\,\lambda(dx_{t+1})$ for a given x_t. Hence, the prediction step of the particle filter remains unchanged, while the particles are weighted by the product $p(y_{t+1}|x_{t+1})\, p_{pose}(x_{t+1})$ instead of the likelihood during updating.

Only in rare cases we are able to give an analytical expression for p_{pose}. Instead, we suggest to learn the probability of the various poses from a finite set of training samples. For a nonparametric estimate of the density we use a Parzen-Rosenblatt estimator [14]

$$p_{pose}(x) = \frac{1}{(2\,\pi\,\sigma^2)^{d/2}\,N} \sum_{i=1}^{N} \exp\left(-\frac{d(x,x_i)^2}{2\,\sigma^2}\right) \tag{6}$$

to deal with the complexity and the non-Gaussian behavior of the distribution, where N denotes the number of training samples and the function d is a distance measure in E. This estimate depends on the window size σ that is necessary to be chosen in an appropriate way. While a small value of σ forces the particles to stick to the training data, a greater value of σ approximates the density smoother. In order to cope with this, we chose σ as the maximum second nearest neighbor distance between all training samples, i.e. the two neighbors of a sample are at least within a standard deviation. Other values for the window size are discussed in detail in [15].

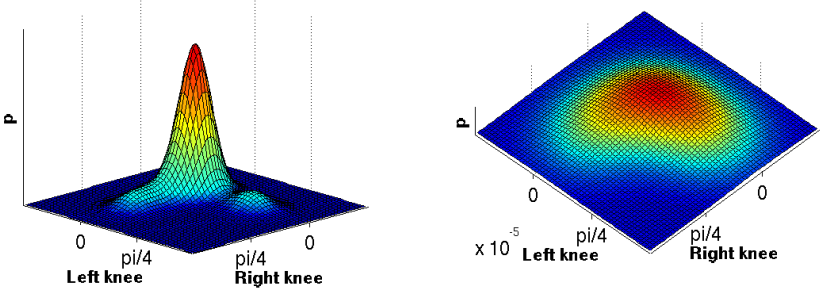


Fig. 2. The Parzen estimate subject to the angles of the knee joints. **Left:** Using the Euclidean distance leads to a domination of the knee joints. The density rapidly declines to zero as the values differ from the data. **Right:** The influence of the knees is reduced by the weighted Euclidean distance.

We have not yet specified the norm for evaluating the distance between a training sample x_i and a value x in the d-dimensional state space E for Equation (6). The commonly used Euclidean distance weights all dimensions of the state space uniformly. This means in the context of human motion estimation

that a discrepancy of the knee contributes to the measured distance in the same matter as a discrepancy of the ankle. As illustrated in Figure 2, this involves a dominated measure by joints with a relatively large anatomical range as the knee in comparison to joints with a small range as the ankle. Therefore, we propose using a weighted Euclidean distance measure that incorporates the variance of the various joints, i.e.

$$d(x, x_i) := \sqrt{\sum_{k=1}^{d} \frac{\left((x)_k - (x_i)_k\right)^2}{\rho_k}}, \qquad \rho_k := \frac{\sum_{i=1}^{N} \left((x_i)_k - \overline{(x)_k}\right)^2}{N-1} \tag{7}$$

where $\overline{(x)_k}$ denotes the arithmetic mean of the samples in the kth dimension. This distance is generally applied in image analysis [16] and is equivalent to a Mahalanobis distance in the case that the covariance matrix is diagonal. A full covariance matrix significantly increases the computation in high dimensional spaces.

Additionally, the prior knowledge is suitable for setting the covariance matrix of the zero-mean Gaussian density p_{pred}. One approach is to estimate the variance of the differences between succeeding samples. However, this has the drawback that training data from a quite large range of dynamics are needed a priori. In the case where the sample data only include walking sequences, the prediction is not accurate for tracking a running person. Thus setting the variances proportional to ρ_k is generally applicable and better than adjusting the parameters manually. We remark finally that not all parameters of a pose can be learned. For example, it does not make sense to learn the position of an object. Therefore, the density is usually estimated in a slightly lower dimensional space than the state space.

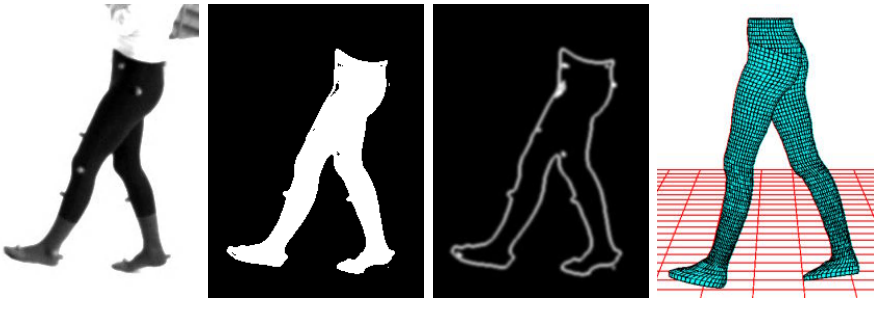


Fig. 3. Feature extraction by level set segmentation. **From left to right:** (*a*) Original image. (*b*) Extracted silhouette. (*c*) The smoothed contour is slightly deformed by the markers needed for the marker based system. (*d*) 3D model with 18 DOF used for tracking.

4 Application to Multi-view 3D Tracking

4.1 Feature Extraction

For weighting the particles during the update step of the APF, features from an image y_t have to be extracted. In previous works, only low-level features assuming a static background as foreground silhouette, edges, or motion boundaries [3,6] were considered. In our work, the level set based image segmentation from Section 2.2 with the experimentally determined parameter $\nu = 4$ is applied using the estimated pose $\widehat{x}_{t-1}$ from the previous time step. The resulting level set describes the silhouette and the contour of the observed object. We remark that the extraction of this image feature is not independent of the estimate anymore. This yields a weighting function that depends not only on the current image and the particle itself, but also on the whole set of particles defining the estimate. Even though particle filters already provide an interaction between the particles due to the normalization of the weights, it holds the danger that a segmentation error leads to an estimate error and vice-versa. However, the influence of the estimate on the segmentation can be regulated by the parameter λ. Our experiments, where we set $\lambda = 0.04$, show indeed that a proper value for this parameter avoids this problem.

4.2 Weighting Function

The error between a particle and the observed image y is calculated pixel-wise similar to [3]. Each particle $x \in E$ determines a pose of our 3D model. The projected surface of the model into the image plane gives a set of silhouette points $S_i^S(x)$ and a set of contour points $S_i^C(x)$ for each view $i = 1, \ldots, r$, where a set contains all pixels $p \in \mathbb{R}^2$ of the silhouette and the contour, respectively. The silhouette S_i^y of the observed object is obtained from the level set function Φ^i, where $S_i^y(p) = 1$ if $\Phi^i(p) > 0$ and $S_i^y(p) = 0$, otherwise. The contour C_i^y is just the boundary of the silhouette smoothed by a Gaussian filter and normalized between 0 and 1, cf. Figure 3. Then the error functions are defined by

$$err_L(x, y, i) := \frac{1}{\left|S_i^L(x)\right|} \sum_{p \in S_i^L(x)} (1 - L_i^y(p))^2. \tag{8}$$

for $L \in \{S, C\}$. Following Section 3, we integrate the learned prior knowledge in form of the probability density p_{pose}. Altogether the energy function of the weighting function (1) can be written as

$$V(x, y) := \sum_{i=1}^{r} (err_S(x, y, i) + err_C(x, y, i)) - \eta \ln(p_{pose}(x)), \tag{9}$$

where the parameter $\eta \geq 0$ controls the influence of the prior knowledge. It is obvious that $V \geq 0$ and $g(x, y)^{\beta_m} \lambda(dx)$ is thus a Boltzmann-Gibbs measure. Furthermore, the constant term $(2\,\pi\,\sigma^2)^{d/2}$ of p_{pose} can be omitted since it is

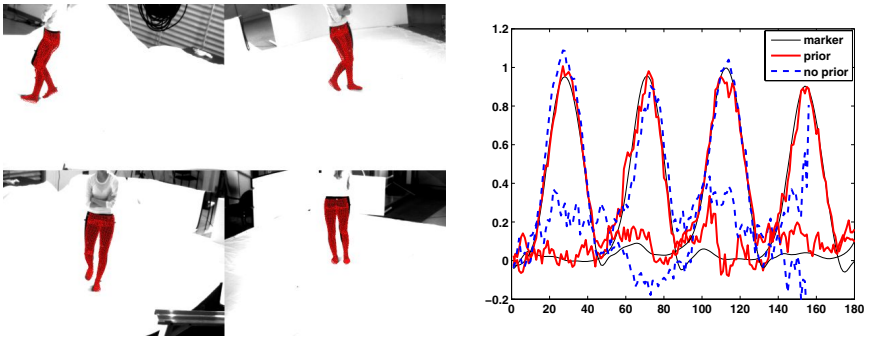


Fig. 4. Left: Results for a walking sequence captured by four cameras. **Right:** The joint angles of the right and left knee. *Solid (thin):* Marker based system. *Solid (thick):* Prior with weighted distance. *Dashed:* Without prior (Tracking fails).

canceled out when normalizing the weights. Note that the prior knowledge is embedded in accordance with the Bayesian framework by multiplying the old-weighting function with $(p_{pose})^{\eta}$. Our method performs well with $\eta \in [0.06, 0.1]$ as we demonstrate below.

5 Experiments

In our experiments we track the lower part of a human body using four calibrated and synchronized cameras. The sequences are simultaneously captured by a commercial marker based system[1] allowing a quantitative error analysis. The black leg suit and the attached retroflective markers are required by the marker based system, see Figure 3.

The training data used for learning p_{pose} consists of 480 samples obtained from walking sequences of the same person. The data was captured by the commercial system before recording the test sequences. The parameters of the APF are set during the experiments as follows: 10 annealing runs are applied with $\beta_m = 8\,(1 - 1.6^{m-11})$ and 250 particles. The resampling step includes a crossover operator [3], and the particles are diffused according to a zero-mean Gaussian distribution with covariance matrix determined by $0.1\,\rho_k$, see (7). The initial distribution is the Dirac measure of the initial pose. Our implementation took several minutes for processing 4 images of one frame.

Figure 4 visualizes results of a walking sequence that is not contained in the training data. For the sake of comparison, the results of the APF without using prior knowledge at all are also visualized in Figure 5. The estimated angles of the left and the right knee are shown in the diagram in Figure 4 where the values acquired from the marker based system provide a ground truth with an accuracy of about 3 degrees. It allows to analyze the quantitative error of our method in contrast to previous works, e.g. [3], where visual comparisons indicate

[1] We used the Motion Analysis system with 8 Falcon cameras.

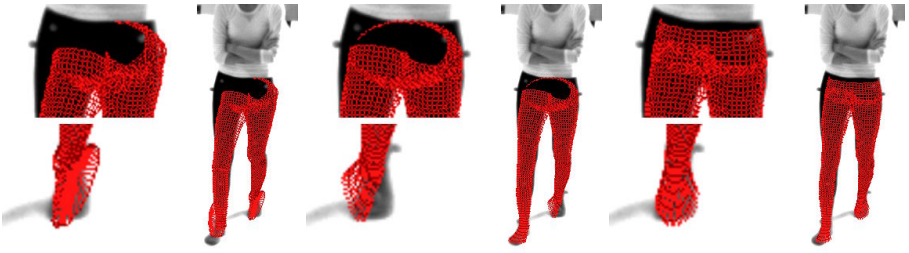


Fig. 5. Visual comparison of results. **From left to right:** (*a*) Without prior. (*b*) Without weighted distance. (*c*) With weighted distance.

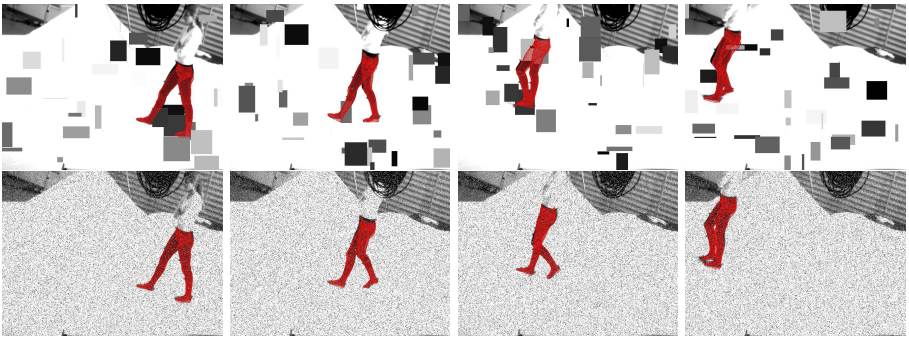


Fig. 6. Results for distorted sequences (4 of 181 frames). Only one camera view is shown. **Top:** Occlusions by 30 random rectangles. **Bottom:** 25% pixel noise.

roughly the accuracy of the pose estimates. The root mean square (RMS) error for both knees is 6.2 degrees (red line). While tracking with 100 particles failed, our method also succeeded using 150 and 200 particles with RMS errors 15.3 and 8.8 degrees, respectively.

Figure 6 shows the robustness in the presence of noise and occlusions. Each frame has been independently distorted by 25% pixel noise and by occluding rectangles of random size, position and gray value. The legs are tracked over the whole sequence with RMS errors 8.2 and 9.0 degrees, respectively. Finally, we applied the method to a sequence with scissor jumps, see Figure 7. This demonstrates that our approach is not restricted to the motion patterns that were used for training as it is when learning the patterns instead of the poses. However, the 7th image also highlights the limitations of the prior. Since our training data are walking sequences, the probability that both knees are bended is almost zero, cf. Figure 2. Therefore a more probable pose is selected with less bended knees. It yields a higher hip of the 3D model than in the image. Overall, the RMS error is 8.4 degrees. A similar error can be observed for the feet since they are more bended for jumping as for walking. Nevertheless, the result is much better than without using any prior.

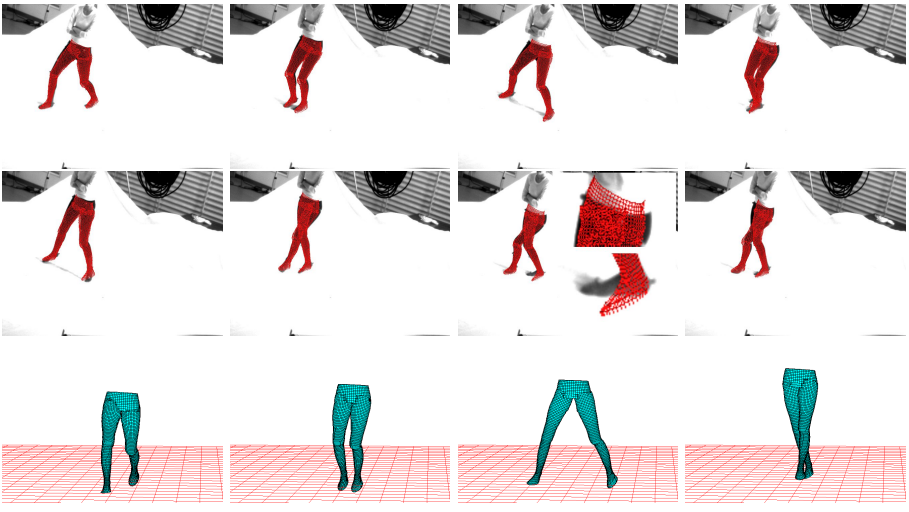


Fig. 7. Rows 1-2**:** Results for a sequence with scissor jumps (8 of 141 frames). **Row** 3**:** The 3D models for the 4 poses on the left hand side of rows 1 and 2 are shown from a different viewpoint.

6 Summary

We have presented a method that integrates a-priori knowledge about the distribution of pose configurations into the general model of particle filters as well as into the special APF scheme. Thereby, the prior ensures that particles representing a familiar pose are favored. Since only single pose configurations and not whole motion patterns are learned, a relatively small set of training samples is sufficient for capturing a variety of movements. Our experiments provide a quantitative error analysis that clearly demonstrates the increased accuracy of the APF due to the incorporated prior knowledge. Moreover, we have shown that our approach combined with a variational model for level set based image segmentation is able to deal with distorted images, a case where common techniques that rely on background substraction fail. Since we were restricted to use artificial distortions by the marker-based system, further work will be done to evaluate the system in real examples like crowded and outdoor scenes. Work on acquiring training data from motion databases and handling occlusions by clothes is also in progress.

References

1. Doucet, A., de Freitas, N., Gordon, N., eds.: Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science. Springer, New York (2001)
2. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. Int. J. of Computer Vision **29** (1998) 5–28

3. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. Int. J. of Computer Vision **61** (2005) 185–205
4. Sidenbladh, H., Black, M., Fleet, D.: Stochastic tracking of 3d human figures using 2d image motion. In: European Conf. on Computer Vision. Volume 2. (2000) 702–718
5. Sidenbladh, H., Black, M., Sigal, L.: Implicit probabilistic models of human motion for synthesis and tracking. In: European Conf. on Computer Vision. Volume 1. (2002) 784–800
6. Sminchisescu, C., Triggs, B.: Estimating articulated human motion with covariance scaled sampling. Int. J. of Robotics Research **22** (2003) 371–391
7. Sminchisescu, C., Jepson, A.: Generative modeling for continuous non-linearily embedded visual inference. In: Int. Conf. on Machine Learning. (2004)
8. Brox, T., Rosenhahn, B., Kersting, U., Cremers, D.: Nonparametric density estimation for human pose tracking. In: Pattern Recognition (DAGM). LNCS, Springer (2006) To appear.
9. Brox, T., Rousson, M., Deriche, R., Weickert, J.: Unsupervised segmentation incorporating colour, texture, and motion. In Petkov, N., Westenberg, M.A., eds.: Computer Analysis of Images and Patterns. Volume 2756 of LNCS., Springer (2003) 353–360
10. Rosenhahn, B., Brox, T., Smith, D., Gurney, J., Klette, R.: A system for markerless human motion estimation. Künstliche Intelligenz **1** (2006) 45–51
11. Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners. IEEE Transaction on Signal Processing **50** (2002) 736–746
12. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220** (1983) 671–680
13. Brox, T., Rosenhahn, B., Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose estimation. In Kropatsch, W., Sablatnig, R., Hanbury, A., eds.: Pattern Recognition (DAGM). Volume 3663 of LNCS., Springer (2005) 109–116
14. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics **33** (1962) 1065–1076
15. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
16. Mukundan, R., Ramakrishnan, K.: Moment Functions in Image Analysis: Theory and Application. World Scientific Publishing (1998)

Improving Brightness for a Multi-projector Display Considering Image Content

Hee-Won Lee and Byung-Uk Lee

Department of Information Electronics, Ewha Womans University
lepink00@hotmail.com, bulee@ewha.ac.kr

Abstract. Very large-area multi-projector display systems have wide applications since they provide immersive environments. The brightness of projectors in a system is not identical due to the design and aging of the bulbs, and optical properties of the projectors. Therefore, it is imperative to coordinate the brightness level and characteristics of projectors in the system. This issue has been addressed by many researchers. However, the state of the art solution has considered the characteristic of the projector's brightness only. In this paper, we propose to take into account the dynamic range of the image content. We expand the dynamic range of the input image to utilize the maximum possible brightness of a projector while satisfying the spatial smoothness constraint. The proposed method achieves a seamless multi-display with impressive improvement of the brightness level.

1 Introduction

The need for very large screen display is increasing for virtual reality or image display due to its immersive impacts. The use of projectors is prevalent because of cost effectiveness of the device; projectors show high quality images with large screen size at affordable cost. Therefore the application area for projectors has been expanded. Since we can generate a large screen using a projector with relatively low price, projectors are widely used for education or business purposes. Because the screen size from one projector is limited, it is a general trend to combine many projectors to compose a very large screen size. Stitching images from multiple projectors has several issues such as aligning various images without distortion and matching the brightness from each projector.

Several algorithms achieve geometrically undistorted images after correcting for geometric distortions [1],[2],[3],[4],[5]. Fig.1 shows an example of intensity mismatch among projectors for tiled display. The brightness of projectors from the same manufacture model is not identical due to aging of bulbs or other optical characteristics. Thus, a great deal of research focuses on solving these problems. Techniques to correct for the uniformity of brightness of a large display system can be categorized broadly into the following three methods: edge blending, brightness matching, and perceptible variation constraint method.

The edge blending method [6],[7] corrects for the brightness of overlapping regions between projectors; however, it does not resolve the different level of brightness between projectors and the non-uniformity of brightness within a projector. On the other hand, the brightness matching method [8],[9],[10] constrains the

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 70 – 78, 2006.
© Springer-Verlag Berlin Heidelberg 2006



Fig. 1. Image degradation arising from multi-projector display showing mismatch and overlap of projected images

brightness range of each projector to the range displayable by all projectors, and therefore can correct for the brightness within a projector, the mutual brightness difference between projectors, and the brightness of overlapping regions. However, since this method limits the level of contrast and brightness of the displayed image to the worst projector, it leads to severe degradation of the image quality.

Therefore, recently, a new technique [11] is proposed which allows the brightness to vary across the screen in a smooth manner such that it stays within the sensitivity level of the human eye. It results in a brighter image while the spatial variation is not perceptible. This method can enhance the brightness depending on the rate of the brightness gradient variation across the screen. In this paper, we propose to improve the technique to utilize the maximum brightness level possible in the projector by considering the brightness property of the input image. We can enhance the brightness dramatically even when the rate of variation on the screen is very small.

This paper consists of the following sections. Section 2 explains the existing method of perceptible variation constraint. In Section 3 we propose a new method of expanding the dynamic range of the input image content to make use of the maximum brightness of projectors. Section 4 compares the test results of the existing method with those of the proposed method to illustrate the enhanced level of brightness. We conclude in Section 5 and present directions for further research.

2 Previous Research: Perceptible Variation Constraint Method

We first describe perceptible variation constraint (PVC) method proposed by Majumder [11]. This method enhances the brightness by allowing gradual increase of image magnitude across the screen. Our improved technique builds upon this method.

The maximum and minimum brightness of each projector is denoted by H'_D and L'_D respectively. Let G be the projector function which represents the transfer function between the input image intensity i and the output brightness of a projector. Then the brightness $B'_D(x,y,i)$ at position (x,y) can be represented as equation (1).

$$B'_D(x,y,i) = G(i)\,(H'_D(x,y) - L'_D(x,y)) + L'_D(x,y) \tag{1}$$

The perceptible variation constraint method takes advantage of the property of human eyes to increase the maximum brightness H'_D. We employ the Weber's law [12] stating that spatially varying brightness, given by the derivative of the brightness at that point, cannot be noticed by human eyes if the rate of change is within a fraction λ of the original brightness.

To derive the contrast λ, we consider the perpendicular distance of the user from the display d, resolution of the display on pixels per unit distance r and the brightness threshold per degree of visual angle for humans τ. Hence, $\lambda = 900\tau / d\pi r$ [11]. The following equation (2) is the Weber's law describing the property of human vision.

$$\left|\frac{\Delta L}{\Delta x}\right| \le \lambda L, \tag{2}$$

In the perceptible variation constraint method, the maximum brightness H'_D can be represented as equation (3) by applying equation (2) to surrounding eight neighboring pixels at position *(x,y)*.

$$\left|\frac{H'_D(x,y) - H'_D(x',y')}{\sqrt{(x-x')^2 + (y-y')^2}}\right| \le \lambda H'_D(x,y), \quad \forall\, x, y, x', y' \tag{3}$$

where, *(x',y')* represents eight neighboring positions of *(x,y)*, $x' \in \{x-1, x, x+1\}$, and $y' \in \{y-1, y, y+1\}$. Of all the H'_D satisfying the above condition, we select H'_D which maximizes the sum of the $H'_D(x,y)$, i.e. $\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} H'_D(x,y)$.

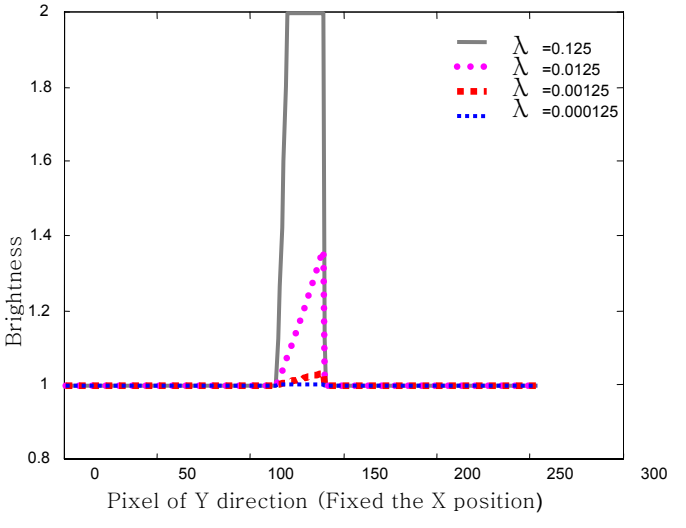


Fig. 2. Profiles of constrained maximum brightness H'_D for various λ values

Please note that H'_D converges to a uniform value as the contrast parameter λ approaches zero as shown in Fig. 2. At the same time the brightness is reduced since the maximum brightness converges to the darkest spot of the display, therefore it reduces to the brightness matching method.

3 Proposed Method: Expanding Dynamic Range of the Input Image Content

We can enhance the brightness of the multi-projector system by utilizing the maximum brightness of a projector considering the input image content as explained in the following subsections.

3.1 Expanding Image Dynamic Range to the Maximum Brightness of Projectors

First, we find the maximum brightness H'_D curve satisfying the smoothness constraint using equations (2) and (3). The relation between the maximum brightness of before and after correction is illustrated in Fig. 3.

The maximum brightness H'_D satisfying the perceptible variation constraint is displayed on the screen only when the input has the maximum value of one. If the maximum input value is less than one, the displayed image is darker, and therefore a projector does not exercise its maximum capability. Thus, we propose to increase the intensity of an input image to utilize the maximum possible brightness of the projector. If the value of an input image is $i(x, y)$, then the display brightness value $M'_D(x, y)$ is the product of H'_D and $G(i(x, y))$ as shown in the following equation (4). When the value of $L'_D(x,y)$ is assumed zero, the value of $B'_D(x,y,i)$ in equation (1) becomes $M'_D(x, y)$.

$$M'_D(x, y, i) = H'_D(x, y)\, G(i(x, y)) \tag{4}$$

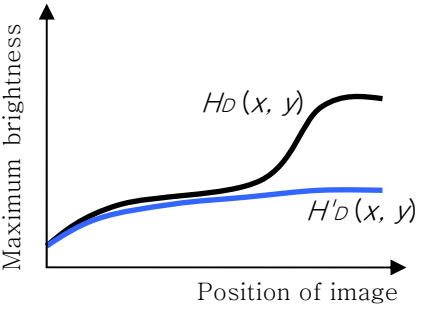


Fig. 3. Maximum brightness H_D and smoothed maximum H'_D of a projector

Since the constrained maximum H'_D is always less than the projector maximum brightness H_D, $M'_D(x, y, i)$ is less than H_D as shown in Fig. 4. We can find a ratio

$k(x, y)$ by dividing H_D by $M'_D(x, y, i)$, which will be used as a factor to increase the brightness $M'_D(x, y, i)$ to the maximum brightness value of H_D.

$$k(x, y) = H_D(x, y) / M'(x, y, i) \tag{5}$$

The value of the scale factor parameter k of an image is set as the minimum value of the $k(x, y)$ values on the image as shown in equation (6), so that the maximum brightness reaches the maximum of the projector brightness H_D.

$$k = \min k(x, y) \tag{6}$$

If we multiply the above factor to the displayed image, then the maximum brightness of the image on the screen reaches H_D. Therefore, the improved output image brightness, $D'_D(x, y, i)$ of the corrected image considering the brightness property of the projector and input image content can be represented as the following equation (7).

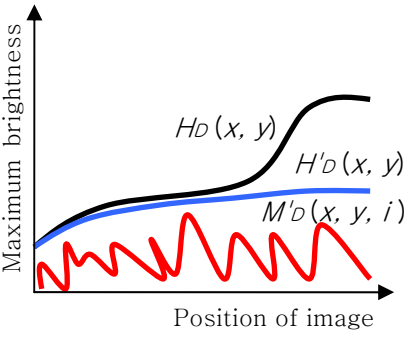


Fig. 4. Brightness of the displayed image, $M'_D(x, y, i)$ and the maximum brightness of a projector, H_D and H'_D

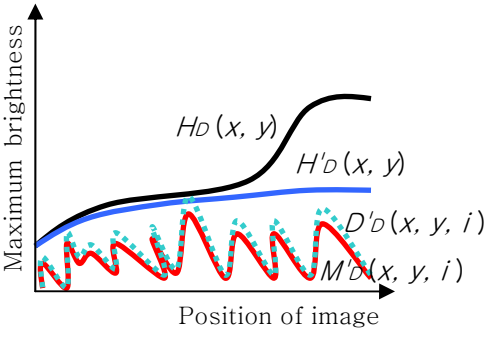


Fig. 5. Brightness of a display image before and after correction: $M'_D(x, y, i)$, and $D'_D(x, y, i)$ with the maximum brightness of a projector, H_D and H'_D

$$D'_D(x,y,i) = M'_D(x,y,i)\ k = H'_D(x,y)G(i(x,y))k \tag{7}$$

In other words, the improved brightness $D'_D(x,y,i)$ is obtained by multiplying the $M'_D(x,y,i)$ value by the k factor obtained from equation (6). The resulting number is greater than $M'_D(x,y,i)$ but less than H_D as illustrated in Fig. 5.

3.2 Saturating a Small Portion of Bright Pixels for Further Improvement

To improve the brightness further, we proposed to increase the k factor from the minimum value of $k(x,y)$. Some high intensity pixels may become larger than the maximum brightness if we increase the k factor above the minimum level of the whole image. In reality, the brightness of those pixels is saturated to the maximum value of the projector. If the number of saturated pixels is below 0.02% of the total image, then the distortion of brightness is not perceivable to the human eye while enhancing the brightness drastically.

To decide on the increased k value, we constrain the number of pixels whose value will be saturated to the maximum brightness of the projector H_D. We employ a symbol k_p to denote a k value with the number of $p\%$ pixels saturated over the H_D curve. After testing several cases, we concluded that $p = 0.02\%$ showed drastic improvement of brightness without noticeable image degradation.

3.3 Mapping of Input Image for Scaling [11]

In order to display an image with improved brightness from a projector, we must determine the value of the input image corresponding to the output brightness value

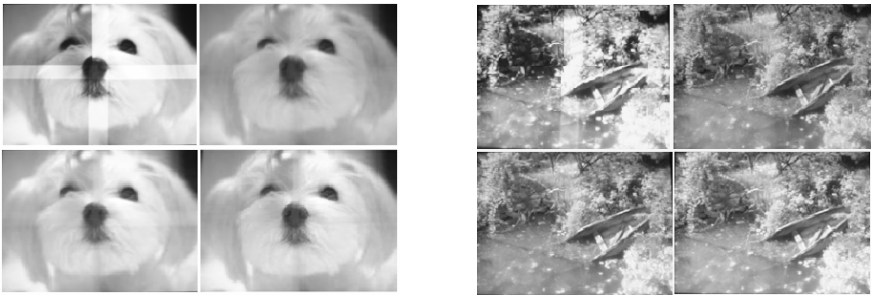


Fig. 6. Image displayed using four projectors of the 2x2 array. Upper left: image before correction, upper right: perceptible variation constraint method, lower left: proposed method with λ=0.0025, without saturation, lower right: proposed method with λ=0.0025,p=0.02%.

$D'_D(x,y,i)$ of the improved display. In this paper, we assume that L_D and L'_D values are zero for simplicity, and the corrected input image i' is given by Majumder [11].

$$i' = G^{-1}(G(i)\,D'_D(x,y)) \tag{8}$$

The function G is a nonlinear function representing the input image and the output brightness relationship of a projector. G^{-1} is the inverse function of G which can be represented using a look-up table. Thus, in order to obtain an image after brightness correction, we multiply the desired output brightness $D'_D(x, y, i)$ by the $G(i)$ and find each i' value using the input-output transfer function of the projector.

3.4 Limiting Brightness Fluctuation for Video images

We have described a method to increase the brightness of a single image in the previous subsections. We can again apply the same idea to motion pictures; we calculate the k factor for each frame to improve the level of brightness. However, we must limit the change in the multiplication factor to avoid annoying brightness fluctuation from frame to frame. We set the maximum variation of the k value using an exponential model. Equation (9) decides the maximum variation.

$$P = 1 - e^{-t/T} \tag{9}$$

We can determine the variation rate P between frames by equation (9), where t is the frame interval and T is the time constant. The time constant T is set to 2 seconds after subjective experiments to suppress the fluctuation. In other words, the maximum variation during 2 seconds is limited to e^{-1}, or 37%, which is an acceptable variation from the experiments. As a result, the variation rate P between frames became 0.017 during 1/30 seconds, which means that the k value may change below 1.7% in sequential frames.

4 Experimental Results

We experimented with various contrast parameters λ and scale factor k using two projector configuration of 2×1 array, and four projector of 2×2 array. Fig. 6 and 7 show the resulting images. First, we observe the effect of the contrast parameter λ and then we improve the brightness by applying the scale factor k obtained from equation (6).

As shown in Table 1 and Fig. 6, we observe that the image becomes darker with smaller values of the contrast parameter, since the maximum brightness converges to a uniform value, which is the darkest spot of the entire image. However, the brightness of the proposed method is much better than that of the perceptible variation constraint method since the brightness is improved by the factor k even when the value is small.

Fig. 7 shows images taken from experiments. It compares the performance of the PVC method and the proposed method. It also shows average output brightness ratio of a test image with increasing k_p values. With the increase of the ratio of the saturated pixels, the image becomes brighter while allowing greater contrast. However, if the image intensity saturation becomes severe, the quality of the image deteriorates. It is recommended to keep the saturated pixels under 0.05%.

Table 1. Comparison of the average brightness in the result image of a dog from the existing method (PVC) and the suggested method according to various λ values

Contrast parameter (λ)	Existing method	Suggested method
0	45.5	100.8
0.0005	51.7	106.7
0.00125	62.9	113.5
0.0025	88.2	112.2

When we observe the resulting images for the case of λ = 0.0005, the average brightness of the proposed method is 2.07 times brighter than the perceptible variation constraint method. If we allow saturation rate of 0.01%, the image becomes brighter by a factor of 2.34. We can observe visible enhancement of contrast compared to the proposed method. We also notice that the improvement becomes smaller with bright images or large λ. However, the improvement becomes more prominent for small values of λ.

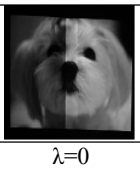
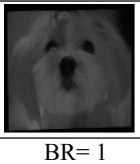
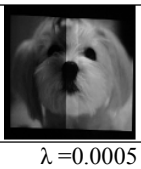
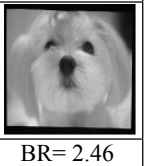
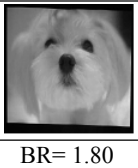
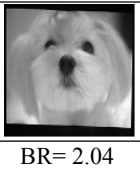
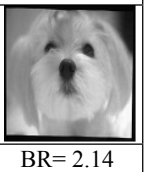
Before correction	PVC method	Proposed method		
		k_0	$k_{0.01}$	$k_{0.02}$
				
λ=0	BR= 1	BR= 2.22	BR= 2.50	BR= 2.64
				
λ =0.0005	BR= 1	BR= 2.07	BR= 2.34	BR= 2.46
				
λ =0.00125	BR= 1	BR= 1.80	BR= 2.04	BR= 2.14

Fig. 7. Photographs of images on the screen for various values of scale factor k

5 Conclusions

We can improve the brightness of the multi-projector system by increasing the dynamic range of the input image content so that the projector can make use of the

maximum brightness. The improvement ratio is approximately two for various natural images. The proposed method is based on the PVC [11], and it is improved by adding a simple process that finds a scale factor k. It is employed to change the input images in real-time on commodity graphics hardware to correct for the brightness variation.

The enhancement factor depends on the image content; therefore it varies for each image. Thus we suggest limiting the change for each consecutive frame of a video sequence. While the proposed method is confined to brightness enhancement, it is necessary to further pursue research on color matching after modifying the level of brightness and real time implementation.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communications), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. Chen, C.J.: Fundamentals of scalable high resolution seamlessly tiled projection system. SPIE, Projection Displays VII, Vol. 4294 (2001) 67-74
2. Hereld, M., Judson, I.R., and Stevens, R.: Dottytoto: A measurement engine for aligning multi-projector display system. Argonne National Laboratory preprint ANL/MCS-P958-0502 (2002)
3. Stone, M.C.: Color and brightness appearance issues for tiled displays. Computer Graphics and Applications (2001) 58-66
4. Raskar, R.: Multi projector displays using camera based registration. IEEE Visualization, San Francisco (1999) 161-168
5. Chen, H., Sukthankar. R., Wallace, G., and K. Li.: Scalable alignment of large-format multi-projector displays using camera homography trees. IEEE Visualization, Boston (2002) 339-346
6. Majumder, A., Stevens, R.: LAM: Luminance attenuation map for photometric uniformity in projected based display. ACM Virtual Reality and Software Technology (VRST), Hong Kong (2002) 147–154
7. Chen, C. J., and Johnson, M.: Fundamentals of scalable high resolution seamlessly tiled projection system. SPIE Projection Displays VII, Vol. 4294, San Jose (2001) 67-74
8. Majumder, A.: Achieving color uniformity across multi-projector displays. IEEE Visualization, Salt Lake City (2000) 117-124
9. Wallace, G., Chen, H., and Li, K.: Color gamut matching for tiled display walls," In Proc. of Immersive Projection Technology Symposium (2003) 293-302
10. Majumder, A.: Properties of color variation across multi-projector displays. SID Eurodisplay, Nice, France (2002)
11. Majumder, A.: Contrast enhancement of multi-displays using human contrast sensitivity. IEEE International Conference on Computer Vision and Pattern Recognition, San Diego (2005) 377-382
12. Pratt, W.K.: Digital Image Processing, 3rd edn., John Wiley & Sons, Inc., New York (2001)

VirtualQWERTY: Textual Communication in Virtual Reality

Jaewoo Ahn[1] and Kyungha Min[2,⋆]

[1] Mobience, Inc./Information Communication Univ.
[2] Sangmyung Univ.

Abstract. This paper presents *VirtualQWERTY*, an effective and powerful textual communication toolkit for virtual reality. The first key idea of this paper is that we exploit the text input mechanism on mobile phones. We present a widget whose user interface is similar to that of the keypad on a mobile phone. Each keypad is composed of several buttons that have several characters. The second key idea of this paper is that the characters on the keypad are arranged by mimicking the character layout of the QWERTY keyboard. These key ideas present users an intuitive and easily learnable textual communication toolkit in virtual reality. In order to exploit *VirtualQWERTY*, we present an indexing scheme by which users locate the button they want and a selection scheme by which users choose the desired character from the characters on the button using various input devices. After testing this tool by many subjects, we draw an important conclusion about the performance of the textual communication toolkit in virtual environments.

1 Introduction

The needs for textual communication in virtual reality are increasing very rapidly. For example, users in a traditional building walk-through application may want to communicate to other users by exchanging short messages, or users in a virtual museum may want to communicate to $Google^{TM}$ or $Yahoo^{TM}$ to send queries about the items they are looking at. In all these applications, a common requirement is a textual communication toolkit in virtual reality.

In this paper, we present *VirtualQWERTY* toolkit for textual communication in virtual reality. The first key idea in this paper is that we develop a textual communication scheme by exploiting the text input scheme on mobile phones. These days, people frequently compose and send short text messages using SMS (Short Message System) applications on the phones, and vendors of mobile phones provide simple text input applications in the phones. We analyze the text input schemes of the SMS applications and design a similar text input scheme for virtual environments. The major difficulty of practicing the current SMS applications is that the text input speed is not fast enough to be practical, and that people feel inconvenient or get some burden in entering the characters.

⋆ Corresponding author: minkh@smu.ac.kr. Prof. Kyungha Min, Dept. of Digital Media, Sangmyung Univ., 7 Hongji-dong, Jongro-gu, Seoul, 110-743, Korea.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 79–87, 2006.
© Springer-Verlag Berlin Heidelberg 2006

We argue that the difficulty mainly comes from the character layout on the keypad of mobile phones. Currently, most of the phones adopt the so-called "ABC" layout, which arranges characters in the alphabetical order.

The second key idea is that we present an intuitive and easily learnable character layout on the keypad. For this purpose we analyze the usage frequency of characters and the character layout of the most widely-used QWERTY keyboard. The characters are arranged on a keypad of 3×3 buttons according to the layout of QWERTY keyboard, and then the order of the characters on a button are determined by the usage frequency of the characters.

The third key idea is that the textual communication scheme should support numerics and symbols as well as characters. This can be simply achieved by adding a button for changing the text input mode. As a whole, the *VirtualQWERTY* toolkit possesses three input modes: character mode, numeric mode and symbol mode.

The *VirtualQWERTY* toolkit consists of a screen keypad that displays a layout of buttons and a text input scheme. Each button on the keypad possesses two or three characters on it. In the text input scheme, inputting a character is achieved in two steps: the first step is an indexing step, where users locate a button that possesses the character to input, and the second step is a selection step, where users select the character to input among the characters on the button indexed. The text input scheme can be implemented with various hardware devices. In this paper, we implement the scheme on a data glove, a multi-button joystick, a multi-button mouse, and a gaming gun. The selection step supports both of the multi-tapping and direct-tapping. In the multi-tapping scheme, the number of tapping times determine the desired character. For example, in selecting 'B' from "ABC", users should press the button twice. On the contrary, the direct-tapping uses different buttons to select the desired character. In selecting 'B' from "ABC", users are required to press the second button, which is the predefined button of the input device.

This paper is organized as follows. In Section 2, we briefly review the related work, and we analyze the text input mechanism on mobile phones in Section 3. We design the *VirtualQWERTY* in Section 4, and we implement the scheme in Section 5. In Section 6, we summarize this paper and suggest future work.

2 Related Works

In virtual reality, several schemes have been proposed to introduce text input in immersive virtual environments. Poupyrev et. al. [8] proposed a *virtual notepad*, which records characters written on a notepad in the virtual environment. This scheme enables users to leave a short hand written message in the virtual environment. This scheme, however, cannot be regarded as a text input scheme, since only the drawings of the characters written on the notepad are simply recorded and played back. Fels and Hinton [5] proposed *GloveTalkII* scheme that converts gestures into speech by recognizing the predefined gestures using neural nets. The hand gesture contains ten control parameters, which are

combined to represent the context to express. The limitation is that the range of expressible contexts is limited and that it requires a reasonable exercise to use this scheme. Rosenberg and Slater [9] presented a *chording glove*, which interprets combinations of touches of fingers on a hard surface as a character. Even though this scheme can represent all the alphanumeric characters, users are required to memorize the correspondences between the characters and the chords of fingers. Bowman et. al. [3] have presented a *virtual keyboard* implemented by a pinch glove and tested its usability over other text input schemes. This scheme presents an intuitive interface for the text input by displaying the keyboard in immersive virtual environments and selecting the characters on the keyboard by the interactions of the pinch glove. This scheme, however, suffers from the inefficiency of the input procedure. Shin and Hong [10] presents the *Keypad glove* scheme for the text input in wearable computing environments. A data glove in one hand plays the role of keypad of a mobile phone by assigning each button to the corresponding finger segment on the glove, and the other hand plays the role of pressing the button. The difference from the selection step of the text input scheme in a mobile phone is that a character among the characters on a button is selected directly by assigning different indices to the three fingers of the other hand. Frees et. al. [6] have presented *Connect The Dot* scheme for text input in immersive virtual environments. In the scheme, the user interface is composed of nine dots organized in 3×3 structure. Users input characters by connecting the dots in some predefined orders. In addition to these researches, several researchers [11, 7, 4] have tested existing text input schemes for virtual environments and wearable computing environments and compared them through empirical studies.

3 Analysis of Text Input Scheme of Mobile Phones

The fundamental operation of text input scheme for alphabetic characters is to choose one of possible twenty six characters. When we use a device that has twenty six individual buttons with each button having only one character, such as a keyboard or a typewriter, the only operation required to enter a character is to locate the button that contains the desired character. Since the size of a typical mobile device, however, is too small to support a full-sized keyboard, it requires a special scheme to support text input in those devices. The text input schemes developed for such devices can be classified into two categories: using a keypad of limited number of buttons or using character recognition techniques. Many mobile phones use the keypad-based scheme and many PDA's use the recognition-based scheme. We focus on the keypad-based scheme used in mobile phones. In the keypad-based scheme, each button on the keypad contains two or three characters due to the limited number of buttons. The basic philosophy of the text input operation in this scheme is to separate the operation into two steps. The first step, which is denoted as *indexing step*, is to locate a button whose character set contains the desired character and the second step, which is denoted as *selection step*, is to select the desired character from the character set.

In most mobile phones, the selection step is implemented by the multi-tapping scheme. For example, pressing the button whose character set is "A B C" once enters "A", and pressing the button twice enters "B".

3.1 Analysis on Usage Frequency of Characters

We analyzed dozens of novels written in the standard English [1, 2] to get a general knowledge of alphabet usage statistics. The usage frequency of alphabets and the space character is summarized in Figure 1.

char.	frequency	char.	frequency	char.	frequency	char.	frequency
space	18.70	s	5.16	c	1.92	v	0.76
e	10.18	h	5.15	w	1.86	k	0.68
t	7.33	r	4.70	f	1.82	x	0.12
a	6.61	d	3.69	g	1.70	j	0.10
o	6.16	l	3.42	y	1.62	q	0.07
n	5.60	u	2.33	p	1.37	z	0.06
i	5.59	m	2.08	b	1.22	Σ	100.0

Fig. 1. A table for the usage frequency of characters including alphabet and the space character

Another statistical factor we have to keep in mind is the frequency of adjacent characters. Those characters should be located in separate buttons in order to avoid the latencies of entering two characters in the character set of the same button consecutively. A table of frequencies of adjacent characters is shown in Figure 2 [1, 2].

pair	frequency	pair	frequency	pair	frequency	pair	frequency
he	3.94	ou	1.43	it	1.18	te	1.01
th	3.75	ed	1.41	ng	1.17	es	0.99
in	2.39	on	1.35	hi	1.14	ar	0.98
er	2.20	ha	1.35	is	1.06	of	0.98
an	2.20	at	1.32	as	1.05	le	0.95
re	1.66	en	1.31	or	1.05	se	0.88
nd	1.63	to	1.22	st	1.02	ve	0.88

Fig. 2. A table for frequencies of pairs of adjacent characters

4 Design of *VirtualQWERTY*

4.1 Consideration of QWERTY Keyboard

The most important factor in designing *VirtualQWERTY* is how to arrange characters on the buttons of a keypad. We observe that the characters can be grouped according to the following properties:

Grouping property 1
The characters in the QWERTY keyboard are arranged in three lanes: top, middle, and bottom lane. 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', and 'p' belong to the top lane, and 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', and 'l' belong to the middle lane. 'z', 'x', 'c', 'v', 'b', 'n', and 'm' belong to the bottom lane.

Grouping property 2
According to the hand that we enter a character in the QWERTY keyboard, we can group the characters into two groups: the left and the right group.

In *VirtualQWERTY*, we place buttons on the keypad in a 3×3 arrangement. Each button is labelled according to its position on the keypad, and the labels are the combinations of $\{ left, center, right \} \times \{ top, middle, bottom \}$.

In determining the positions of characters on the buttons of *VirtualQWERTY*, we use the first rule below:

Rule 1
The characters on the buttons of *VirtualQWERTY* should be positioned according to the positions on the QWERTY keyboard. It means those characters on the top lane are recommended to be placed in *top* buttons. Further, the characters in the left group are recommended to be positioned in $left$ or *center* buttons.

4.2 Consideration of the Usage Frequency

According to the usage frequency in Figure 1, we use the following rules in positioning the characters:

Rule 2
We position those characters of high usage frequency in separate buttons in *VirtualQWERTY*. Therefore, { 'e', 't', 'a', 'o', 'n', 'i', 's', 'h', 'r' }, the nine characters of highest usage frequency are positioned on separate buttons.

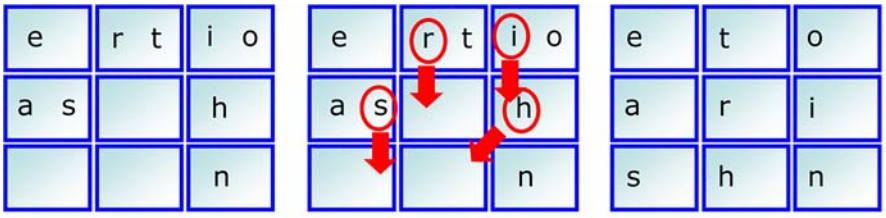
Rule 3
In positioning characters of lower usage frequency on the buttons, we try to distribute the frequencies of characters over the buttons on the keypad.

Figure 3 shows how the characters are positioned in the buttons of *VirtualQWERTY*. In Figure 3 (d), the final layout of *VirtualQWERTY* is illustrated.

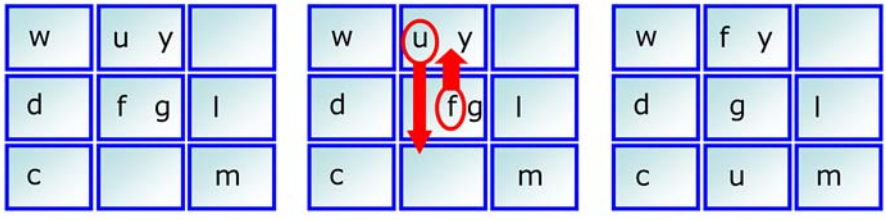
4.3 Design of Input Scheme

The input scheme of *VirtualQWERTY* is composed of two steps: indexing and selection. Both of the steps are designed as follows:

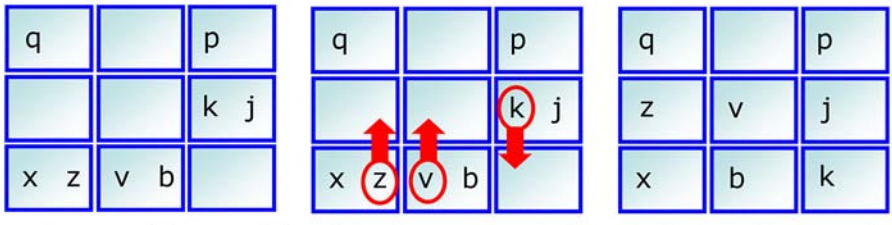
Indexing. Indexing a button is implemented in two ways. One way is to trace the *position* of the device and identify the button at the position of the device, and the other way is to recognize the *direction* of the device and identify the button in the direction.



(a) Layout of nine characters of highest frequencies: 'e','t','a','o','n','i','s','h', and 'r'



(b) Layout of next nine characters: 'd', 'l', 'u', 'm', 'c', 'w', 'f', 'g', and 'y'



(c) Layout of the remaining eight characters: 'p', 'b', 'v', 'k', 'x', 'j', 'q', and 'z'



(d) The layout of *VirtualQWERTY* (e) Sums of the frequencies of characters on buttons

Fig. 3. The layout of the characters on the buttons of *VirtualQWERTY*. The left column of (a), (b), and (c) shows the initial layout of the characters according to their positions on QWERTY keyboard, and the center column shows the change of the positions according to the usage frequency. The right column shows the resulting layout of the characters.

Selection. In mobile phones, the number of presses of a button selects the character on it. For example, pressing a button that contains "A", "B", and "C" once selects "A", twice selects "B", and three times selects "C". This selection scheme is due to the limited number of buttons of mobile phones. In this paper, we use two classes of selection schemes.

- **Multi-tapping:** The selection scheme for one-button devices is designed by the multi-tapping scheme, which selects a character on the button by the number of presses. In this scheme, a threshold value t_p is used as a timeout interval. Successive presses of the same button within t_p signify that the presses are within a selection. On the contrary, a successive press of the same button after t_p signifies the start of a new selection on the button. For example, pressing a button of "ABC" twice within an interval less than t_p enters "B" and greater than t_p enters "A A".
- **Direct-tapping:** The selection scheme for multi-button devices can be designed using the multiple buttons on the device. We assign different selection identifiers to each of the individual buttons on the device. The selection schemes for the devices exploited in this paper are listed as follows:
  - *Data glove:* In the case of using a data glove, we assign the selection function to different fingers. For example, an index finger selects the first character on the button, a middle finger selects the second, and a ring finger selects the third.
  - *Two-button device:* In the case of using a two-button device such as a mouse or a joystick, clicking the left button selects the first character, the right button selects the second, and both buttons simultaneously selects the third.
  - *Multi-button device:* Each button on the device is dedicated to each selection of the characters on a button.

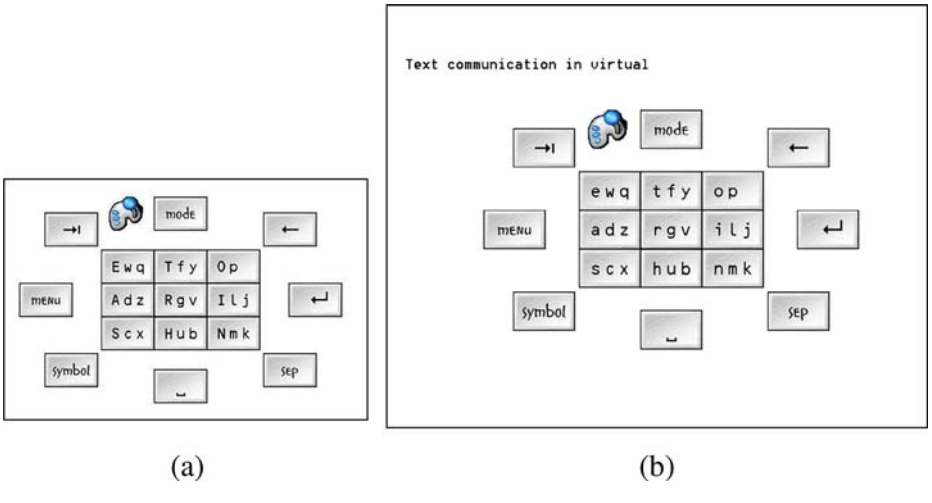
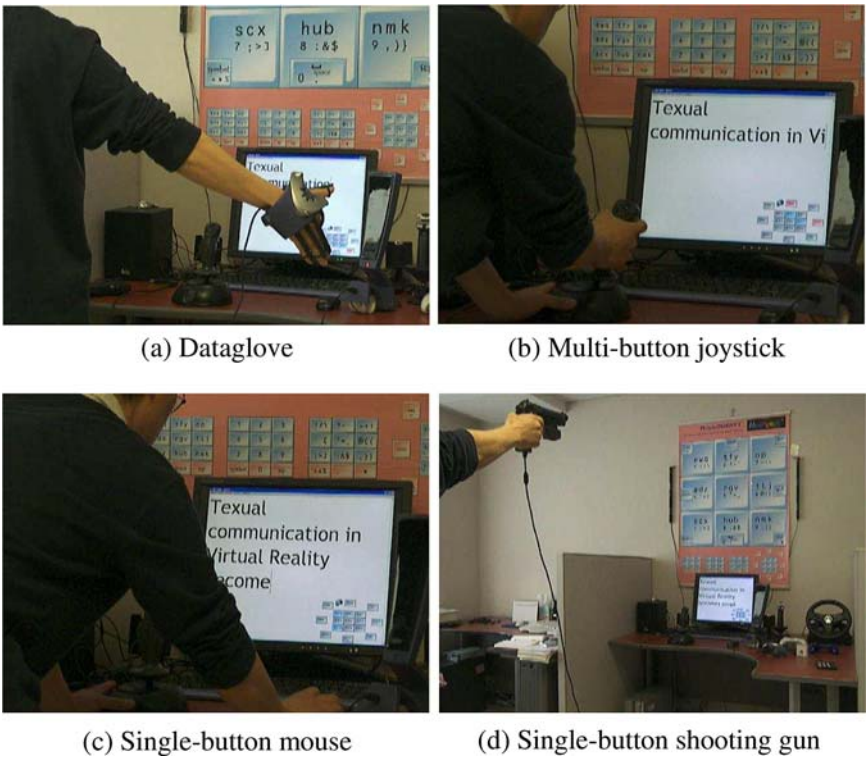


Fig. 4. The *VirtualQWERTY* toolkit implemented: (a) The shape of the toolkit, (b) Text input example on an empty notepad

5 Implementation of *VirtualQWERTY*

Figure 4 illustrates the shape of *VirtualQWERTY* toolkit, and Figure 5 illustrates the snapshots of *VirtualQWERTY* demonstration. In this demonstration, we have implemented the *VirtualQWERTY* in four devices: a data glove, a multi-button joystick, a mouse, and a gaming gun.



(a) Dataglove (b) Multi-button joystick

(c) Single-button mouse (d) Single-button shooting gun

Fig. 5. The demonstration of *VirtualQWERTY* using various devices

6 Conclusions and Future Work

In this paper, we have presented *VirtualQWERTY*, a textual communication toolkit for virtual reality applications. The toolkit presents an intuitive and easily learnable input scheme with a new layout of characters on a keypad of 3×3 buttons by exploiting the character layout of QWERTY keyboard and the usage frequency statistics of characters. This toolkit is proved to be very powerful and robust in text input applications in virtual environments.

As a future work, we plan to implement *VirtualQWERTY* in an immersive virtual environment. Our another future work is to develop several virtual reality applications that require textual communications such as googling or chatting in virtual reality.

References

[1] Project Gutenberg Official Home Site. *http://www.promo.net/pg/*.
[2] The Online Books Page. *http://onlinebooks.library.upenn.edu/*.
[3] Bowman, D. and Wingrave, D. and Campbell, J. and Ly, V.: Using Pinch Gloves for both Natural and Abstract Interaction Techniques in Virtual Environments. Proceedings of HCI International, 629–633, 2001.
[4] Bowman, D. and Rhoton, C. and Pinho, M.: Text Input Techniques for Immersive Virtual Environments: an Empirical Comparison. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 2154–2158, 2002.
[5] Fels, S. and Hinton, G.: Glove Talkll: A Neural Network Interface which Maps Gestures to Parallel Formant Speech Synthesizer Controls. IEEE Transactions on Neural Networks, **9(1)**, 205–212, 1998.
[6] Frees, S. and Khouri, R. and Kessler, G.: Connecting the Dots: Simple Text Input in Immersive Environments, Proceedings of IEEE Virtual Reality, 25–29, 2006.
[7] Lindeman, R. and Sibert, J. and Hahn, J.: Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments, Proc. Of the SIGCHI '99, 64–71, 1999.
[8] Poupyrev, I. and Tomokazu, N. and Weghorst, S.: Virtual Notepad: handwriting in immersive VR. Proceedings of the Virtual Reality Annual International Symposium, 126–132, 1998.
[9] Rosenberg, R. and and Slater, M.: A Chording Glove: A Glove Based Text Input Device, IEEE Transactions on Systems, Man, and Cybernetics, **29(2)**, 186–191, 1999.
[10] Shin, J.H. and Hong, K.S.: Keypad gloves: glove based text input device and input method for wearable computers, Electronics Letters, **41(16)**, 15–16, 2006.
[11] Thomas, B. and Tyerman, S. and Grimmer, K.: Evaluation of Text Input Mechanisms for Wearable Computers. Virtual Reality, **3(3)**, pp. 187-199, 1998.

Networked Heterogeneous Camera System for High Resolution Face Images

Sofiane Yous, Abdelaziz Khiat, Masatsugu Kidode, and Tsukasa Ogasawara

Graduate School of Information Science
Nara Institute of Science and Technology
Keihanna Science City, 630-0192, Japan
{yous-s, abdela-k, kidode, ogasawar}@is.naist.jp

Abstract. This paper presents a system that combines fixed stereo cameras with wide field of view (FOV) and high resolution active Pan/Tilt (PT) cameras into a networked platform. Designed for security applications, the purpose is to provide high resolution face images while covering a wide area. A custom PT unit allowing a fixed camera projection center was designed to hold the high resolution camera. Two such cameras in addition to one stereo camera were implemented in a real scene in order to test the effectiveness of the system. Preliminary results demonstrate the efficiency of the proposed system.

1 Introduction

Visual surveillance is getting an increasing interest as security issues are becoming a major concern nowadays, where surveillance devices are more and more spread in public and sensitive areas to prevent against crimes and accidents. Visual surveillance covers issues such as behavior analysis and people tracking and identification. To this last issue which is of key importance, an extensive literature is devoted. However, the main shortcoming of the existing systems is identification failure in wide area observation, due to low resolution images. In such situations, most of the systems limit there processing to a tracking task [6][4].

Few systems have been proposed in the literature to overcome these limitations. The face cataloger system [5] combines two stationary cameras with wide FOV with two active Pan/Tilt/Zoom cameras, in order to provide high resolution face images while covering a wide space. Foreground segmentation followed by a model-based head tracking is done, simultaneously, on the two stationary cameras. The 3D head position, obtained by triangulation, and the velocity of the person are used for steering the active cameras. These cameras wait for the face to be stable to zoom up and capture a close up image. This constraint, however, limits the usability of the system to situations where people are aware of its existence.

We propose in this paper a heterogeneous camera system made up of fixed wide FOV stereo and active high resolution PT cameras in order to provide high resolution face images while covering a wide area. The stereo cameras have as role

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 88–97, 2006.
© Springer-Verlag Berlin Heidelberg 2006

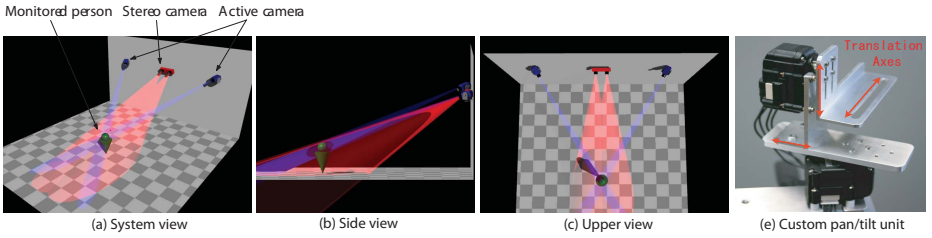


Fig. 1. The setup of the proposed system: the stereo camera detects the face within the observable area and estimates its 3D position. Accordingly, the active cameras gaze the face and provide high resolution images.

to detect a face within the observable scene, estimate its 3D position, and pass it to the active cameras. Based on the passed 3D position, the active cameras change their orientations to gaze the face and display it in high resolution, as shown in Fig.1. In order for an active camera to gaze any indicated face, the camera should have access to the entire region covered by the stereo camera. The collaboration between the camera hosts is made possible by inter-connecting the hosts through a network platform.

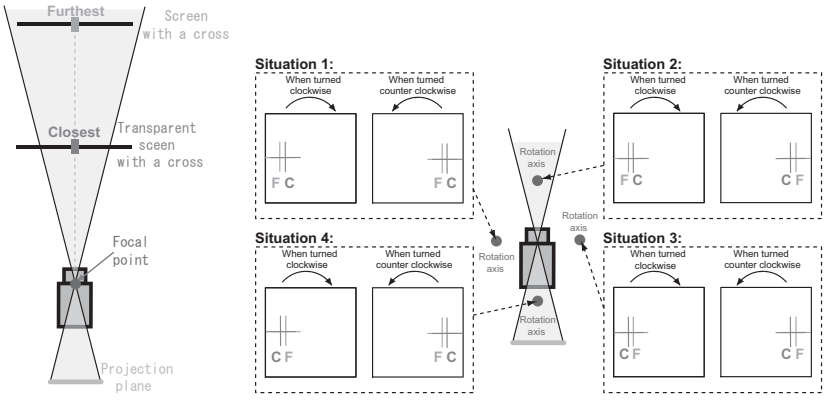
2 System Setup

2.1 Hardware Requirement

An active camera is made up of a 50-mm focal length camera (Flea^{TM}of Pt-Grey) held by a PT unit. We designed the PT unit shown in Fig.1(e), using two Dynamixel^{TM}DX-117 modules interconnected using three metal parts, such that the held camera can have 3DOF. Thus, the projection center of the camera can be adjusted to meet with the rotation center of the unit, through a fixed viewpoint calibration. The stereo camera we used is a Vedere Design MEGA-D^{TM}camera with 20-cm baseline and 25mm focal length. Each camera is connected to one PC host and communicates with the other hosts using sockets via a network.

2.2 Fixed Viewpoint Calibration

We aim through this procedure at bringing the projection center of the camera to meet with the rotation center of the PT unit whereon, the camera is attached. In many computer vision applications, the camera internal and external parameters are needed. For a stationary camera, these parameters are set once and for all, as a beforehand setup. For an active camera, however, the extrinsic parameters have to be re-set after each camera movement. If the projection center of the camera is itself the rotation center, then only the camera rotation matrix needs to be calculated. On the other hand, if we know with precision the pan and tilt rotation angles of the PT unit, and since the rotation matrices of the unit and the camera are the same, then even the rotation matrix could be calculated

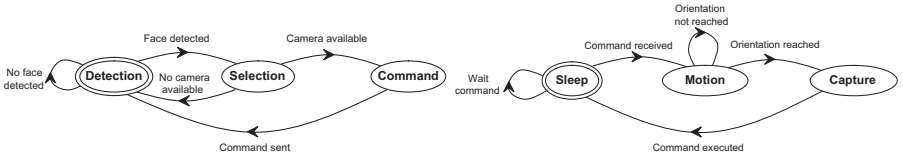


(a) Initial setup. (b) Possible situations during the calibration process

Fig. 2. Fixed-viewpoint calibration process

directly. For the fixed viewpoint calibration, and based on the method described in [7], we make use of two parallel planes in front of the camera. The two planes consist in two screens on which a cross sign is drawn in different color for each plane, and where the closer screen is transparent. As an initialization, we set the camera and the two planes such that, the center of the camera image and the two crossing points are on the same line of sight. We, then, apply a pan rotation in the two directions (left and right) and get two images. Depending on the position of the rotation center with respect to the camera projection center, different image is obtained and different camera position adjustment on the PT unit has to be operated. As shown in Fig.2, the fixed view point calibration can be summarized in the following pseudo-code:

repeat
  Rotate(CW); // Apply a clockwise pan rotation
  $X_{11} \Leftarrow CrossPosition(image1)$ // Crossing position on the near plane
  $X_{21} \Leftarrow CrossPosition(image2)$ // Crossing position on the far plane
  Rotate(CCW); // Apply a counterclockwise pan rotation
  $X_{12} \Leftarrow CrossPosition(image1)$ // Crossing position on the near plane
  $X_{22} \Leftarrow CrossPosition(image2)$ // Crossing position on the far plane
  if $X_{11} > X_{12}$ and $X_{21} < X_{22}$ **then**
    Move the camera backward
  end if
  if $X_{11} < X_{12}$ and $X_{21} > X_{22}$ **then**
    Move the camera forward
  end if
  if $X_{11} < X_{12}$ and $X_{21} < X_{22}$ **then**
    Move the camera leftward
  end if
  if $X_{11} > X_{12}$ and $X_{21} > X_{22}$ **then**
    Move the camera rightward



(a) Operation of the stereo camera host. (b) Operation of the active camera host.

Fig. 3. Finite state machines for the system operation

```
  end if
until X11 = X12 and X21 = X22
repeat
  Rotate(Upward);                    // Apply an up rotation
  Y1 ⇐ CrossPosition(image1) // Crossing position on the near plane
  Y2 ⇐ CrossPosition(image2) // Crossing position on the farth plane
  if Y1 > Y2 then
    Move the camera upward
  end if
  if Y1 < Y2 then
    Move the camera downward
  end if
until Y1 = Y2
```

2.3 Global System Calibration

In order for the active cameras to gaze the face indicated by the stereo camera, all camera, regardless of their characteristics, must be calibrated together to the same world coordinates system. The calibration is done for each camera separately using the OpenCV [2] implementation of the method presented in [3]. We begin by one common reference plane, to which corresponds the world coordinate system, and we complete the multi-plane calibration for each camera separately. After the calibration is done for all cameras, we set for each camera the external parameters to be those related to the common reference plane.

3 System Operation

As explained so far, the stereo host is in charge of:

- detecting a face within the covered space,
- estimating its 3D position,
- and pass this 3D position to the available active cameras.

The stereo host, as summarized by the finite state machine of Fig.3(a), keeps detecting,continuously, a face in the scene. It leaves this state when a face is detected and comes back to it if no active camera is available. If one or more cameras are available, the stereo host passes to them the 3D face position and goes back to the detection state. As for the active hosts, they:

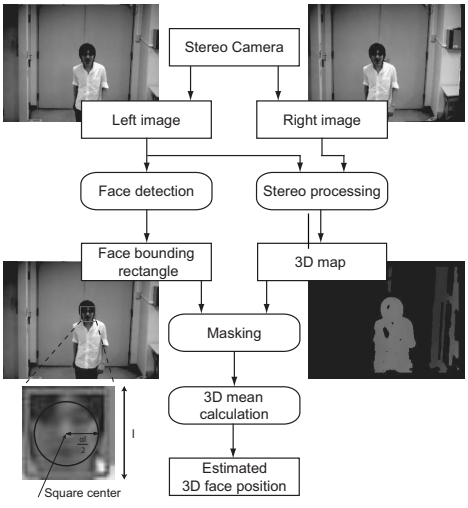


Fig. 4. Process of 3D face position estimation

– receive the 3D face position from the stereo host,
– adjust the local parameters accordingly,
– and control the PT unit in order for the camera to gaze the face.

When an active camera host receives the 3D face position from the stereo one, it changes its state to busy, processes the necessary transformations, and executes the rotation. When the 3D position is reached, the camera host executes a release notification and goes back to the sleeping state to wait for a new command, as shown in Fig3(b).

4 Estimation of the 3D Face Position

As shown in the diagram of Fig.4, the stereo camera provides two images. The left stereo camera is set as principal with respect to which, and using both images (left and right), the 3D map is obtained by stereo processing. Simultaneously, faces are detected in the left image. Once a face is detected, the offset and the size of its bounding square are returned. The image region bounded by the face square is, then, set as the region of interest (ROI), and the corresponding points from the 3D map are extracted. Finally, the mean position of the extracted points is set as the estimated 3D face position. For face detection several methods have been proposed [1]. Thanks to its good performance and speed, we have employed the appearance based approach presented in [8]. We used its OpenCV[2] implementation.

It happens that the square bounds a region larger than the face itself. In such cases, 3D points other than those contained in the face are taken into account, and consequently, the estimation error is large. To cope with this problem, the region from the square concerned by the mean calculation is limited by a circle

whose diameter is shorter than the square size. In our implementation, we have set the circle diameter, denoted d, as a scale of the square size, see Fig.4. If the square size is $l \times l$ then:

$$d = \alpha l \quad \alpha \in]0, 1] \tag{1}$$

Let us denote by $O = (x_o, y_o)$, the offset of the bounding square with respect to the image. The set V of the 3D points to be averaged is given by:

$$V = \{map(x, y) / \left(|x - x_o + \frac{l}{2}| < \frac{\alpha l}{2}\right) \\ \wedge \left(|y - y_o + \frac{l}{2}| < \frac{\alpha l}{2}\right) \wedge (valid(x, y))\} \tag{2}$$

where map(x,y) returns the 3D point related to the image point (x,y), and $valid(x, y)$ returns *true* if the image point corresponds to a valid disparity (successfully mapped).

Before passing it to the a free active camera, the face 3D position, calculated in the left stereo camera coordinates and denoted P_s, has to be transformed to the world coordinate system. Let us denote by A_s, the projection matrix of the left stereo camera and by P, the 3D point in the world coordinates. Then, P can be written as:

$$P = A_s^{-1} P_s \tag{3}$$

5 Active Camera Control

The active camera receives the 3D position P to gaze to from the stereo camera. As aforementioned, this position is expressed in the world coordinates. Before any control, the active camera host has to translate this position into the local coordinates. Let us denote by A_c, the camera projection matrix. Then $P_c = (x_c, y_c, z_c)$, the 3D position in the local coordinates, is given by :

$$P_c = A_c P \tag{4}$$

Using P_c, the camera host can deduce φ and γ, respectively, the pan and tilt rotation angles, as shown in fig.5. First, P_c is normalized to get the unit vector to this position, denoted $\overrightarrow{N} = (x_n, y_n, z_n)$. φ and γ are given by:

$$\varphi = arctan(\frac{x_n}{z_n}) \qquad \gamma = arcsin(y_n) \tag{5}$$

The PT unit executes a rotation in pitches units. The value of a pitch defines the angular resolution of any PT unit, and is $0.37\,deg$ for our designed unit. Having a target 3D position, one has to find the appropriate rotation angles and convert them to pitch units. Furthermore, the obtained orientations are with respect to the camera coordinates. However, it is not necessary that the camera and the corresponding PT unit have the same coordinate system. Therefore, we have to add to these rotation angles the rotations of the reference orientations

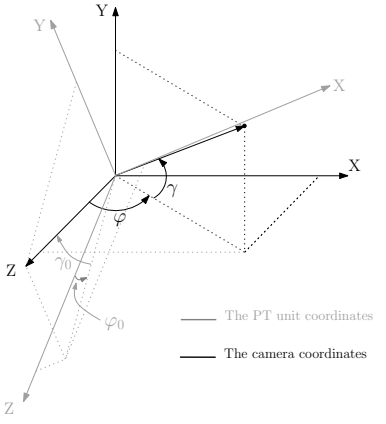


Fig. 5. Pan and tilt rotation angles with respect to the PT unit

of the camera with respect to the PT unit. The reference orientation for each camera, denoted (φ_0, γ_0), is saved once the global system calibration is done, see Fig 5.

The final rotation angles φ_f and γ_f are given by:

$$\varphi_f = \frac{\varphi + \varphi_0}{res} \qquad \gamma_f = \frac{\gamma + \gamma_0}{res} \tag{6}$$

where res is the PT resolution.

Each of the two angle is sent, thereafter, to the appropriate PT unit motor through a rotation command. The time spent by the PT unit to reach the target rotation, referred to as *control delay*, depends on the motor speed and the wideness of the angles. The rotation speed is as high as 70 rpm for our designed unit.

6 Experimental Results

In order to evaluate the performance of the proposed system, we conducted experiments whose results will be presented in this section. In this preliminary experiments, we considered two aspects, namely, the system operation, and the face 3D position accuracy.

As shown in Fig.6, The monitored area has a size of $2 \times 2m$ and located at the entrance of our experiment room. The cameras were fixed at the ceiling, high of about $3m$. The active cameras were fixed at different distances from the stereo camera and at opposite sides. The purpose is to test the influence of the angle between the projection axes of the stereo and active camera on the monitoring performance. First, we evaluate the system operation, then we discuss the accuracy of the 3D face position estimation.

6.1 System Operation

The goal of this experiment is to show how the different components of our system behave when a person is within the monitored area, and also to test the

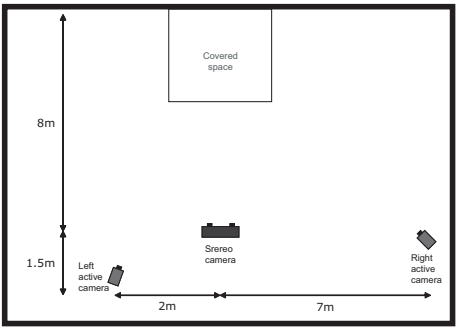


Fig. 6. View of the experiment setup



Fig. 7. Sample of the results taken during the preliminary experiment

influence of the spacial camera placement on the system performance. A person was invited to access and walk through the covered area while facing the stereo camera.

In Fig.7-b, the left stereo camera images on which are superimposed the respective bounding square indicating the detected faces, are shown. In (a) and (c) columns of this figure, the two active camera images taken at the same moment are shown. As for the last column, it shows the positions of the subject within the covered space. The difference in the position of the face with respect to the higher resolution images from one raw to another, is due essentially to the camera control delay while the subject is in motion, in addition to the resolution of the PT unit (0.37 *deg*). It can be also noticed that this variation is more important in the camera with longer distance from the stereo camera (right camera). This means that the control delay for an active camera becomes more important as the angle between its projection axis and this of the stereo camera increases.

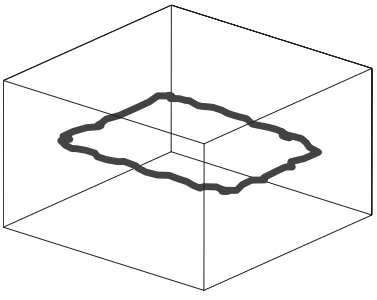


Fig. 8. Face position estimation

We can conclude that this angle should be taken into account in the spatial design of the proposed system. In general systems with several stereo and active camera, to one stereo camera should be associated the closest active cameras with the narrowest aforementioned angles.

6.2 3D Face Position Accuracy

In order to evaluate the accuracy of the 3D face position estimation, a person is invited to walk through a rectangular trajectory within the covered space. The purpose is to check if the estimated 3D positions of the subject's face corresponds to the trajectory. The estimated 3D positions are plotted in the 3D space, and the plotted points are joined together by interpolation. The result is shown in Fig.8 where it can be noticed that the resultant shape is rectangular as is the subject's trajectory. Further numerical experiments are planned in near future for more precise evaluations.

7 Summary

In this paper, a system platform for visual surveillance and security applications was introduced. Made up of fixed stereo cameras with wide FOV and active high resolution PT cameras, the system is designed to provide high resolution face images while covering a wide area. The preliminary experiments have shown a good performance of the proposed system in terms of 3D face position estimation.

As a future work, more evaluations and an extension of the system involving more cameras and covering wider areas, are planned. This extension will require further considerations such as:

- System architecture: depending on the wideness and nature of area to cover different architectures should be adopted. i.e. the system designed to cover a gate differs from the one to cover a waiting space.
- Active camera assignment: face orientation if known can be taken as a selection criterion to find the best camera facing the face.

Acknowledgments

This work is supported by the 21st Century COE program 'Ubiquitous Networked Media Computing' of Nara Institute of Science and Technology (NAIST-JAPAN). We would like to give a special thank to Hamid Laga for his help in the final stage of this paper.

References

1. M. Yang, D. Kriegman and N. Ahuja, "Detecting faces in images: A survey", *IEEE Transactions on Pattern Analysis and Machine intelligence,* Vol. 24, pp.34-58, January 2002.
2. "OpenCV : Open Computer Vision Library", *www.intel.com/research/mrl/research/opencv/.*
3. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 22(11), pp 1330-1334, 2000.
4. T. Ahmedali and J.J. Clark, "Collaborative Multi-Camera Surveillance with Automated Person Detection," in Proceedings of the Canadian Conference on Computer and Robot Vision, 2006, pp. 3-10.
5. A. Hampapur, S. Pankanti, A. Senior, Y-L. Tian, L. Brown, and R. Bolle, "Face Cataloger: Multi-Scale Imaging for Relating Identity to Location," in Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance, 2003, pp.13-20.
6. J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, "Multi-Camera Multi-Person Tracking for EasyLiving," in Proceedings of IEEE Workshop on Visual Surveillance, 2000, pp. 3-10.
7. T. Wada and T. Matsuyama, "Appearance Sphere: Background Model for Pan-Tilt-Zoom Camera," in Proceedings of IEEE International Conference on Pattern Recognition, 1996, pp. 718-722.
8. P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Proceedings of IEEE Computer Vision and Pattern Recognition, 2001, pp. 511-519.

A New Method for Approximating Optimal Parameterization of Polynomial Curves*

Fenghua Guo[1] and Caiming Zhang[1,2]

[1] School of Computer Science and Technology, Shandong University
250061 Jinan, P.R. China
[2] School of Computer Science and Technology, Shandong Ecnomic University
250014 Jinan, P.R. China
gfh@sdu.edu.cn, czhang@sdu.edu.cn

Abstract. Rational re-parameterizations of a polynomial curve that preserve the curve degree and [0,1] parameter domain are characterized by a single degree of freedom. The optimal re-parameterization in this family can be identified but the existing methods may exhibit too much residual parametric speed variation for motion control and other applications. In this paper, a new re-parameterization method to optimal parameterization is presented and the optimal parameterization in this family obtained by the new method satisfies that the maximum deviation from unit-speed is the minimum. Experiments for comparing the efficiency of this algorithm with other methods are also included.

Keywords: Parametric curves; Rational parameterization; Unit speed; Arc-length parameterization.

1 Introduction

The impossibility of constant parametric speed—i.e., of equal increments in the parameter defining equal increments in arc length [1]—is a fundamental limitation of the polynomial and rational curve parameterizations commonly used in computer-aided design, manufacturing, and inspection applications. Digital motion control is a typical context in which the impossible "ideal" of arc length parameterization is desirable[2], since it facilitates the realization of prescribed (constant or variable) speeds along curved paths.

The common practice in computer numerical control (CNC) machining is to attempt real-time compensation for non-uniformity of the parameterization. This is accomplished by *real-time interpolator* algorithms, which is usually based on Taylor series expansions [3][4].

In some applications, an explicit re-parameterization that brings a curve "close"(in the space of rational functions) to the arc-length parameterization is desired, rather than "on-the-fly" compensation for non-uniformity of the parametric speed. A number of authors [5-7] have proposed schemes to obtain approximations to the arc-length parameterization, based mostly on imposing constraints that are consistent with it at a

* Project supported by the National Nature Science Foundation of China (No. 60573180, No. 60533060).

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 98 – 105, 2006.
© Springer-Verlag Berlin Heidelberg 2006

set of discrete positions. However, the designation optimal parameterization has been used inappropriately in the context of such schemes, since they lack the following characteristics of a formal optimization problem:

- specification of a set P of admissible parameterizations;
- introduction of a metric $\left\| p - \bar{p} \right\|$ defining the distance between each member $p \in P$ and the arc-length parameterization $\bar{p}$;
- formulation of an algorithm that identifies a member p_* such that $\left\| p_* - \bar{p} \right\| \le \left\| p - \bar{p} \right\|$ for all $p \in P$

Optimal parameterizations that satisfy these criteria were introduced in Farouki[8] ,choosing P to be the set of rational parameterizations of a given degree, and $\|.\|$ to be L_2 norm of the deviation of the parametric speed from unity. This problem admits a single degree of freedom, whose optimal value can be determined [8]. But the parametric speed of the optimal parameterizations obtained by Farouki[8] can't guarantee to be close to unit speed in the whole parameter domain, i.e., the parametric speed in some domain is close to unit-speed while in the other domain may have a large deviation from unit-speed.

To achieve more accurate approximations of arc-length parameterization, Costantini [9] presents a piecewise-rational scheme and discuss its asymptotic behavior for large number of nodes. In this paper, a new quantitative measure of "closeness" to arc-length parameter is formulated. By new measure, the rational representation in this family satisfying that maximum deviation from unit-speed is the minimum is the optimal parameterization. The advantage of the new method is that the maximum deviation from unit-speed by new method is less than the one by Farouki[8]. Experiments for comparing the efficiency of this algorithm with the method of Farouki [8] are also included.

2 Re-parameterization of Polygonal Curves

An algebraic curve that admits a polynomial parameterization has a single point at infinity. In the customary representation, this point corresponds to an infinite parameter value. By a projective transformation of the parameter domain, however, the curve may also be expressed in rational form. In such a representation, the point at infinity corresponds to a finite parameter value, whereas an infinite value identifies some affine point of the curve. This gives a "geometrical" characterization of the family of parameter transformations that we shall invoke to optimize the parametric flow across a finite curve segment. In particular, Farouki [8] reduces the problem of finding the rational parameterization of a polynomial curve that is "closest" to the unit-speed or arc-length parameterization to the determination of the real root on (0,1) of a quadratic equation expressed in Bernstein form.

Given a polygonal Bézier curve

$$p(t) = \sum_{k=0}^{n} p_k b_k^{(n)}(t) \quad t \in [0,1] \quad \text{with } b_k^{(n)}(t) = \binom{n}{k} (1-t)^{n-k} t^k \tag{1}$$

The above curve may be re-parameterized by a fractional linear (Möbius) transformation. The most general form of Möbius transformation that maps the intervals $u \in [0,1]$ and $t \in [0,1]$ into each other is

$$t = \frac{(1-a)u}{a(1-u)+(1-a)u} \quad u \in [0,1] \tag{2}$$

where $0 < a < 1$. On substituting (2) into the curve (1) we obtain the rational representation

$$q_a(u) = \frac{\sum_{k=0}^{n}(1-a)^k a^{n-k} p_k b_k^{(n)}(u)}{\sum_{k=0}^{n}(1-a)^k a^{n-k} b_k^{(n)}(u)}$$

The subscript indicates the dependence on the parameterization variable a . Note that the denominator polynomial is simply $w(u) = \left[a(1-u)+(1-a)u\right]^n$, so the rational representation can be simply

$$q_a(u) = \frac{\sum_{k=0}^{n}(1-a)^k a^{n-k} p_k b_k^{n}(u)}{(a(1-u)+(1-a)u)^n} \tag{3}$$

$q_a(u)$ is a rational curve with "weights" of the form $w_k = (1-a)^k a^{n-k}$ ($k = 0,...,n$) can always be reformulated as a polynomial curve of the same degree, since by substituting the inverse of (2),namely

$$u = \frac{at}{(1-a)(1-t)+at}$$

into the rational curve (3), we recover the polynomial curve (1)[8].

3 Optimal Curve Parameterization of Farouki [8]

Similar to the work Farouki [8], our work also exercises the freedoms of re-parameterization of polynomial curve segments to achieve a "parametric flow" closeness to unit-speed or arc-length representation. Now we introduce the work of Farouki [8] first.

The total arc length of curve (1) is

$$S = \int_0^1 \left\| \frac{dp}{dt} \right\| dt \tag{4}$$

The function $\left\| \frac{dp}{dt} \right\|$ defines the parametric speed of the curve, i.e., the rate of change its arc length S with respect to the parameter t . The equation of $\left\| \frac{dp}{dt} \right\| \equiv 1$ identifies the unit-speed or arc-length parameterization.

Unfortunately, however, the only curves that admit polynomial or rational unit-speed parameterisations are straight lines [1]. For curve (1), our concern is to obtain re-parameterizations $q(u) = p(t(u))$, that are "as close as possible" to the unit-speed parameterization. We assume that the curve (1) is regular- i.e., it is at least of class C^1

and $\left\|\frac{dp}{dt}\right\| > 0$ for $t \in [0,1]$. The measure of "closeness" to the arc-length parameterization adopted by Farouki[8] is the following integral

$$I = \int_0^1 \left(\left\| \frac{dq_a}{du} \right\| - 1 \right)^2 du \,. \tag{5}$$

Farouki[8] minimizes the integral among the specified class of admissible reparameterizations to obtain the optimal parameterization.

The parametric speed of the optimal parameterization obtained by Farouki[8] can't guarantee to be close to unit speed in the whole parameter domain, i.e., the parametric speed in some domain is close to unit-speed while in the other domain may have a large deviation from unit-speed.

4 The New Optimal Curve Parameterization

The rational curve $q_a(u)$ dependents on the parameterization variable a . Given a value of variable a , there is a maximum deviation from unit speed. Obviously, one of the best choices is to make the maximum deviation from unit speed as small as possible. The new method has the property that the maximum deviation from unit speed of the curve determined by the new method is the smallest among the family of curve $q_a(u)$.

4.1 The Basic Idea

Our measure of "closeness" to arc-length parameterization is

$$E_a(u) = \max\left(\left\| \left\| \frac{dq_a}{du} \right\| - 1 \right| \right) \quad u \in [0,1], a \in (0,1) \tag{6}$$

which we seek to minimize over free parameterization variable a ,i.e.,

$$\min\left(\max\left(\left\| \left\| \frac{dq_a}{du} \right\| - 1 \right| \right) \right) \tag{7}$$

Clearly $E_a(u) \geq 0$ and $E_a(u) = 0 \Leftrightarrow p(t)$ is the straight line.

4.2 Implementation

Problem (7) is an asymmetric continuous minimax problem, because the solution (a^*, u^*) doesn't satisfy the following conditions

$$f(a^*, u) \leq f(a^*, u^*) \leq f(a, u^*)$$

where $f(a,u) = \left(\left\| \left\| \frac{dq_a}{du} \right\| - 1 \right| \right)$.

Many authors have studied the minimax problems. For example, the method of [10-11] can solve the symmetric continuous minimax problems. The method of [12-13] can solve asymmetric continuous minimax problem, but the method has a

relatively high computational load. Particle swarm optimization (*PSO*) is a population–based evolutionary algorithm originally developed by Kennedy and Eberhart[14]. It can solve many complicated problems, such as nonlinear, indifferentiable, mutimodal problems. The implementation of *PSO* is very simple. Eberhart [15] had made a summary about *PSO*. The method of [16] can effectively solve symmetric and asymmetric continuous minimax problems. The basic idea is that the method maintains two populations coevolving asymmetrically in two different search spaces, and a particle swarm optimization algorithm is employed to find a good solution to the minimax problems. So we adopt method of [16] to solve problem (7).

Let a and u belongs to domain A and U . $L(k)$ and $S(k)$ describes the generation of k in domain A and U . The algorithm is outlined as follows:

1) $k = 0$. Create initial generation $L(0)$ and $S(0)$ .Let $w = -Inf$ and $b = -Inf$.

2) for each $a \in L(k)$ do

{ for each $u \in S(k)$ do

{if $w < f(a,u)$ then { $w = f(a,u)$, $(a_w, u_w) = (a,u)$ }

$h(a) = \max_{u \in S(k)} f(a,u)$

}.

3) Let $-h(a)$ be the fitness function, i.e., when $h(a)$ is smaller, the value of fitness function becomes larger. Create the new generation $L(k+1)$ with *PSO* optimization algorithm.

4) for each $u \in S(k)$ do calculate $g(u) = \min_{a \in L(k)} f(a,u)$ .Let $g_m = \max(g(u))$, then u_m satisfies $g(u_m) = g_m$. if $b < g_m$,then $b = g_m$, $u_b = u_m$.

5) Create $S(k+1)$ by random, then replace a particle of $S(k+1)$ with u_w randomly.

6) $k = k+1$. if $k < k_{\max}$ then return to 2).

7) a_0 and u_b which corresponds to the optimal value of fitness function is the resolution to the original minimax problem.

Substitute a_0 into curve (3),we get the optimal parameterization $q(u)$.

Just like most of Genetic Algorithms, PSO can seek the optimal domain quickly, but has a poor performance in local optimization. We modify the step 7), perform a local optimization around a_0 and u_b , which can improve the local optimization largely.

5 Examples and Analysis

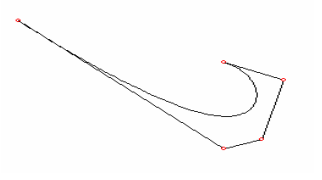
We adopt three test curves. These test curves are

Curve 1: (0.8 9.7) (5.6 5.4) (6.5 5.7) (7.0 7.7) (5.6 8.3)

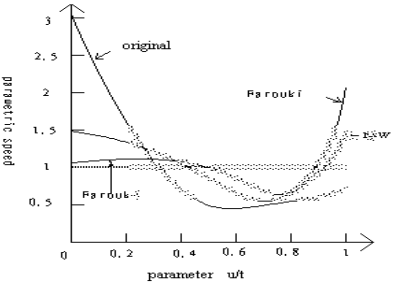
Curve 2: (2.8 3.7) (3.6 3.0) (5.0 4.5) (1.6 5.4) (3.2 8.6) (4.8 6.0)

Curve 3: (0 5.0) (1.0 2.0) (9.0 9.0)

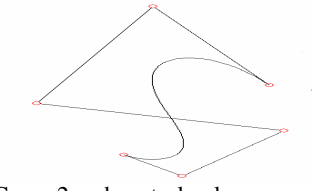
To every test curve, we adopt these two methods to reach different optimal parameterization. The comparison of different parametric speed is shown in Figure1. The comparison of different $E_a(u)$ (the maximal deviation from unit speed) is shown in Table 1.



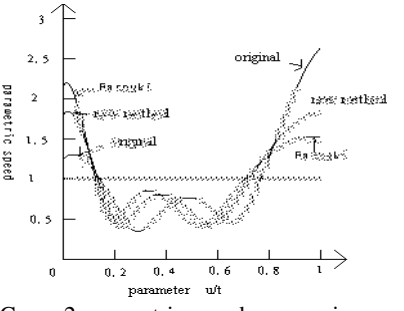
Curve 1 and control polygon



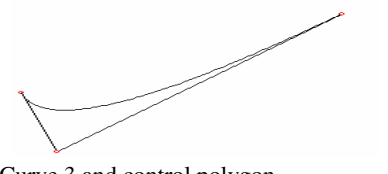
Curve 1 parametric speeds comparison



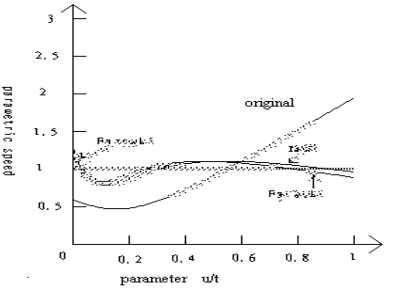
Curve 2 and control polygon



Curve 2 parametric speeds comparison



Curve 3 and control polygon



Curve 3 parametric speeds comparison

Fig. 1. Comparison between Farouki [8] and new method

Table 1. The different $E_a(u)$ reached by the method of Farouki [8] and new method

$E_a(u)$	Original	Farouki[8]	New method
Curve 1	2.03	1.06	0.47
Curve 2	1.63	1.188	0.833
Curve 3	0.94	0.27	0.22

We can conclude from Table 1 and Figure 1, the $E_a(u)$ of optimal parameterization of the same curve reached by new method is less than by the method of Farouki [8], and when the degree of the curve becomes higher, the difference is more obvious, recall that $E_a(u)=0$ is ideal arc-length parameterization. Compared with Farouki[8], the new method's fault is the high computation load. But in some applications, this doesn't matter, for we can compute the optimal parameterization in advance.

6 Conclusions

The impossibility of arc-length parameterization is a fundamental limitation of the polynomial and rational curve parameterizations commonly used in computer-aided design, manufacturing, and inspection applications. Exercising the freedoms of re-parameterization of polynomial or rational curve segments can achieve the optimal parameterization. We present a new quantitative measure the closeness of a parameterization to arc-length parameterization and adopt a new method to obtain the optimal parameterization satisfying that the maximum deviation from unit speed is the minimum in the re-parameterization family.

References

1. Farouki, R.T., Sakkalis, T. Real rational curves are not "Unit Speed". Computer Aided Geometric Design 8,151-157,1991.
2. Zhang, C., Cheng, F. and Miura, K., A method for determining knots in parametric curve interpolation, CAGD 15, 399-416,1998.
3. Farouki, R.T. Convergent inversion approximations for polynomials in Bernstein form. Computer Aided Geometric Design17, 179-196,2000.
4. Tsai,Y.-F., Farouki, R.T., Feldman,B. Perfomance analysis of CNC interpolators for time-dependent federates along PH curves. Computer Aided Geometric Design18,245-265, 2001.
5. Wang, F.C., Wright, P.K. Open architecture controllers for machine tools, Part2:A real time quintic spline interpolator. ASME J. Manufacturing Science and Engineering 120, 425- 432, 1998.
6. Wang, F.C., Wright, P.K., Barsky, B.A., Yang, D.C.H. Approximately arc-length parameterized C^3 quintic interpolatory splines. ASME J. Mech. Design 121,430-439, 1999.
7. Zhang, C., Han, H. and Cheng, F., Determining Knots by Minimizing Energy, Journal of Computer Science and Technology, 21(6), 261-264, 2006..
8. Rida T. Farouki. Optimal parameterizations. Computer Aided Geometric Design 14, 153-168, 1997.
9. Paolo Costantini, Rida T.Farouki, Carla Manni, Alessandra Sestini. Computation of optimal composite re-parameterizations. Computer Aided Geometric Design 18, 875-897, 2001.
10. Hermann J W. A genetic algorithm for minimax optimization Technical Report,TR 97-61,USA:The Institute for Systems Research,University of Maryland,1997,1-15.
11. Hermann J W. A genetic algorithm for minimax optimization problems. Proc 1999 Congress on Evolutionary Computation,Washington D.C.,1999,1099-1103.

12. Jesen M T. Robust and flexible scheduling with evolutionary computation. PhD Dissertation ,Denmark: University of Aarhus,2001,21-32.
13. Jesen M T.A new look at soving minimax problems with coevolution 4th Metaheuristics International Conference ,Porto,Portugal,2001,103-107.
14. Kennedy J, Eberhart R C. Particle swarm optimization. Proc IEEE Int. Conf. on Neural Networks, Perth, WA, Australia,1995,1942-1948.
15. Eberhart R C, Shi Y. Particle swarm optimization: developments, applications and resources. Proc. 2001 Congress on Evolutionary, Seoul, South Korea, 2001, 81-86.
16. Ke Jing, Qian Jixin,Qiao Yizheng. Particle Swarm Optimization Method for Solving Continuous Minimax Problems. Chinese Journal of Scientific Instrument, 2005,26(3): 267-271.(in Chinese)

Interpolation by Piecewise Quadric Polynomial to Scattered Data Points*

Shanshan Gao[1], Caiming Zhang[1,2], and Li Zhong[3]

[1] School of Computer Science and Technology, Shandong Economic University, Erhuandong Road #7366 Jinan, Shandong 250014, China
[2] School of Computer Science and Technology, Shandong University, Jingshi Road #73 Jinan, Shandong 250061, China
[3] School of Computer Science and Technology, Ludong University, Hongqi Road #186 Yantai, Shandong 264025, China
gsszxy@yahoo.com.cn, czhang@sdu.edu.cn, zlaugust@hotmail.com

Abstract. This paper presents a new method to construct C^1 polynomial surface to the scattered data points. The given data points are triangulated into triangle networks, and a C^1 piecewise quadric interpolant patch is constructed at the adjacent region of each point. For the different cases of the number of adjacent points, two different algorithms are used for determining the patches, respectively. The surface patch on each triangle is constructed by the weighted combination of the three patches corresponding to the vertices of the triangle. All of the surface patches are put together to form the whole surface with C^1 continuity. The construction of the surface by the new method is simple, and the surface has the shape suggested by the given data points and is more fairing. Comparison examples are included.

Keywords: Scattered data points; Interpolation; Polynomial Surface; Energy; Triangle.

1 Introduction

The surface modeling is used widely in reverse engineering, shape design, CAD system, Computer Graphics and other scientific research or engineer design. The problem to be studied here is that a set of data points are given, the goal is to construct a fairing surface interpolating to these data points. One of the general methods for constructing interpolant surface is to divide the given scattered data points into triangles, then on each triangular region, a interplant patch is constructed based on the boundary continuous conditions, the whole interpolant surface is combined by the all patches. At present the polynomial interpolant method is widely used for constructing surface on triangular regions because of the simple form and easy computing of polynomial.

* Project supported by the National Nature Science Foundation of China (No.60403036, No.60573114).

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 106 – 115, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Many methods have been applied for constructing interpolation surface with scattered data points [1-4]. Function values of triangular vertices and the first partial derivatives are used to construct four degrees polynomial surface in reference [1], however, the degree of each vertices of triangle is odd. In reference [2], besides the function values of triangular vertices and the first partial derivatives, the first normal derivatives at mid-point of triangle sides are needed, the total interpolation conditions are twelve. In reference [3], the interpolation conditions are only the function values of triangle vertices, but the whole surface is constructed at the local region of each data points, the surface can't has the shape suggested by the given data points. The method [4] works on triangular networks, C^1 surface is determined by solving system of equations. The methods [8-14] can also be used to construct triangular patch by constructing boundary interpolation conditions using the scattered data points.

The constructed surface is expected to achieve certain fairness. Principle of minimum energy is widely used to fair surface. References [5-6] combine designing of surface and minimizing energy subjected to certain constraints to reach the purpose of fairness. Reference [7] studies an interpolation method for scattered data points based on the minimizing energy and the data approximation theory. The Hermite element is used in this paper to minimize objective function. The constructed curved surface can achieve C^1 and C^2 continuity.

A new method for constructing interpolation surface is presented in this paper. New method divides the space data points into triangular networks, and constructs a quadric piecewise polynomial patch corresponding to each vertex of the triangles. For the different number of the adjacent points of each data points, two different methods based on the least square method and minimum of energy method are used to determine the unknown quantities of the patch, respectively. The surface of each triangle is constructed by weighted combination of the three patches.

2 Constructing Interpolation Surface

Let the N given scattered space data points be $p_l = (x_l, y_l, F_l)$, $l = 1, 2, \cdots, N$. The goal is to construct a surface which interpolates to these points. The projection of p on the xy plane is denoted as P_l. The basic idea of constructing interpolation is: Triangulate the given data points in xy plane; construct a piecewise quadric polynomial patch satisfying the interpolation condition on the adjacent regions of each data point P_l, the surface patch on each triangle is constructed by the weighted combination of the three quadric patches at the vertices of the triangle. All the triangle patches form the whole surface with C^1 continuity.

2.1 Constructing Local Polynomial Patch at P_l

For the data point P_l on xy plane, P_1', P_2', $P_3' \ldots P_j'$ is said to be the adjacent points of P_l, and the region of triangle networks constructed by these points and P_l is called the adjacent region of P_l, see **Fig. 1**. Let $f_l(x, y)$ denote local piecewise polynomial patch at P_l. $f_l(x, y)$ is constructed as follows: for each triangle t on the

adjacent region of P_l, construct a quadric polynomial interpolation surfaces based on boundary continuity conditions, then combine the triangle patches on the adjacent regions directly to obtain $f_l(x, y)$.

Firstly, the expression of curved surface $F_t(x, y)$ on each triangle is given. Let t be a triangle with the vertices P_i, P_j and P_k, where (P_i, P_j, P_k) satisfies the principle of right hand coordinate system. Let (L_i^t, L_j^t, L_k^t) be the barycentric coordinate of P in t.

Let $(F_x)_m$ and $(F_y)_m$ denote the first partial derivative of vertex P_m on t, let N_m^t denote the unit outer normal vector on the opposite side of P_m. On each triangular region t, the quadric polynomial curved surface $F_t(x, y)$ interpolating to the vertices function value $F_m, (m = i, j, k)$ and $(F_x)_k$, $(F_y)_k$ of point P_k is:

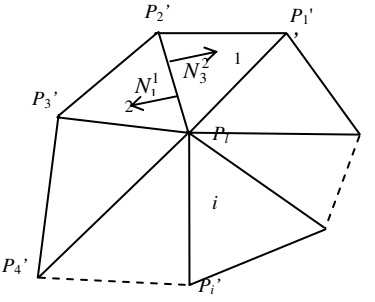


Fig. 1. The adjacent region of P_l

$$F_t(x, y) = [(2 - L_k^t)F_k + (x - x_k)(F_x)_k + (y - y_k)(F_y)_k]L_k^t + (L_i^t)^2 F_i + (L_j^t)^2 F_j + A_t S_t L_i^t L_j^t \quad (1)$$

where S_t is the area of triangle t, and $(F_x)_k$, $(F_y)_k$ and A_t are unknown quantities.

Secondly, the C^1 continuity conditions of the two adjacent triangles are given. The adjacent region of P_l is rounded by the triangles that denote with 1, 2…2n+1(or 2n), see **Fig. 1**. For the triangles 1 and 2, the expressions of patches of the two adjacent triangles are defined as $F_1(x, y)$ and $F_2(x, y)$ by (1). The two patches have the same vertices function value at P_2' and P_l which are $F_{P_2'}$ at P_2' and F_{P_l} at P_l, respectively. Furthermore, the first partial derivatives at the common vertex P_l are the same, so $F_1(x, y)$ and $F_2(x, y)$ are C^0 continuous on the common border. And the unit normal vector on the common border is linear polynomial, so the C^1 continuous condition of the two adjacent triangles $F_1(x, y)$ and $F_2(x, y)$ is defined by:

$$\partial F_1 / \partial N_1^1 \Big|_{\substack{x=x' \\ y=y'}} = -\partial F_2 / \partial N_3^2 \Big|_{\substack{x=x' \\ y=y'}} \quad (2)$$

where (x', y') is the midpoint of $P_2' P_l$. By computing we have:

$$[\partial F_1 / \partial N_1^1]\Big|_{\substack{x=x' \\ y=y'}} = F_l(\partial L_l^1 / \partial N_1^1) + (F_x)_l(\partial x / \partial N_1^1)/2 + (x_2 - x_l)(F_x)_l(\partial L_l^1 / \partial N_1^1)/2 + (F_y)_l(\partial y / \partial N_1^1)/2 + (y_2 - y_l)(F_y)_l(\partial L_l^1 / \partial N_1^1)/2 + 2L_2^1(\partial L_2^1 / \partial N_1^1)F_2 + A_1 S_1 L_2^1(\partial L_l^1 / \partial N_1^1)$$

$$[\partial F_2 / \partial N_3^2]\Big|_{\substack{x=x' \\ y=y'}} = F_l(\partial L_l^2 / \partial N_3^2) + (F_x)_l(\partial x / \partial N_3^2)/2 + (x_2 - x_l)(F_x)_l(\partial L_l^2 / \partial N_3^2)/2 + (F_y)_l(\partial y / \partial N_3^2)/2 + (y_2 - y_l)(F_y)_l(\partial L_l^2 / \partial N_3^2)/2 + 2L_2^2(\partial L_2^2 / \partial N_3^2)F_2 + A_2 S_2 L_2^2(\partial L_3^2 / \partial N_3^2)$$

The continuity function is defined by equation (2) as follows:

$$A_1 + A_2 = q(1,2,l,3) \tag{3}$$

where:

$$q(1,2,l,3) = 2\{[2F_l + (x_2 - x_l)(F_x)_l + (y_2 - y_l)(F_y)_l](\partial L_l^1 / \partial N_1^1 + \partial L_l^2 / \partial N_3^2) + 2F_2(\partial L_2^1 / \partial N_1^1 + \partial L_2^2 / \partial N_3^2)\} / \sqrt{(x_2 - x_l)^2 + (y_2 - y_l)^2}$$

For the each triangle in **Fig. 1**, it interpolates $F_m, (m = i, j, l)$, $(F_x)_l$ and $(F_y)_l$, the quadric polynomial surface $F_t(x, y)$ is defined by (1). The adjacent triangles satisfy C^1 continuity conditions on the common border, namely, satisfy (3), so it is necessary to resolve the system of equations constructed by the continuous conditions. For the two cases that the number of adjacent points of P_l is odd and even, systems of equations about the unknown quantities based on the continuous conditions (2) are defined by:

$$\begin{aligned} &A_1 + A_2 = q[1,2,l,3] \\ &A_2 + A_3 = q[2,3,l,4] \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &A_{2n} + A_{2n+1} = q[2n, 2n+1, l, 1] \\ &A_{2n+1} + A_1 = q[2n+1, 1, l, 2] \end{aligned} \tag{4}$$

$$\begin{aligned} &A_1 + A_2 = q[1,2,l,3] \\ &A_2 + A_3 = q[2,3,l,4] \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &A_{2n-1} + A_{2n} = q[2n-1, 2n, l, 1] \\ &A_{2n} + A_1 = q[2n, 1, l, 2] \end{aligned} \tag{5}$$

Where $(F_x)_k, (F_y)_k$ and $A_t, (t = 1, 2, 3\cdots)$ are unknown quantities. Then the solution for above system of equations is given. For convenience, denotes $q[i, j, k, l] = q_i$.

(i) The number of vertices is odd

Solving the system of equations (4), we have:

If t is odd: $A_t = (\sum_{i=1}^{t-1} (-1)^i q_i + \sum_{i=t}^{2n+1} (-1)^{i+1} q_i) / 2$;

If t is even: $A_t = (\sum_{i=1}^{t-1} (-1)^{i+1} q_i + \sum_{i=t}^{2n+1} (-1)^i q_i) / 2$.

Suppose $(F_x)_l$ and $(F_y)_l$ are known, then $A_t, (t = 1, 2, 3\cdots)$ has one and only one root. So A_t can be represented as a function with $(F_x)_k$ and $(F_y)_k$ defined by:

$$A_t = a_{t1}(F_x)_l + a_{t2}(F_y)_l + a_{t3} \quad t = 1, 2, \cdots 2n+1 \tag{6}$$

Substitute (6) into every expression of triangular curved surface, then the unknown variables are only $(F_x)_k$ and $(F_y)_k$.

(ii) The number of points is even

It's be known from the form of system of equations (5) that A_t ($t = 1, 2, \cdots 2n$) can't be expressed with $(F_x)_k$ and $(F_y)_k$ in the same form as (6) from the above system. Then the solution to system equations (5) is given as follows:

$$\begin{aligned} &A_1 + A_2 = q[1,2,l,3] \\ &A_2 + A_3 = q[2,3,l,4] \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &A_{2n-1} + A_{2n} = q[2n-1, 2n, l, 1] \\ &A_{2n} + A_{2n-1} = q[2n, 1, l, 2] - q[1,2,l,3] + q[2,3,l,4] - \cdots q[2n-2, 2n-1, l, 2n] \end{aligned} \tag{7}$$

In order to get the exclusive solution of (5), constraint function is given as follows:

$$q[2n,1,l,2]-q[1,2,l,3]+q[2,3,l,4]-\cdots q[2n-2,2n-1,l,2n]=q[2n-1,2n,l,1] \quad (8)$$

Then the following expression can be obtained by the system of equations (7) with constraint (8)

If t is odd: $A_t=\sum_{i=t}^{2n-1}(-1)^{i+1}q_i-A_{2n}$;

If t is even: $A_t=\sum_{i=t}^{2n-1}(-1)^{i}q_i+A_{2n}$.

The relationship between $(F_x)_k$ and $(F_y)_k$ can be obtained by equation (8): $(F_x)_l=b_1(F_y)_l+b_2$, substitute it into A_t, then take $(F_y)_l$ and A_{2n} as variables, A_t is expressed as a function of A_{2n} and $(F_y)_k$:

$A_t=a_{t1}(F_y)_l+a_{t2}A_{2n}+a_{t3}$, $t=1,2,\cdots 2n-1$

Therefore the unknown variables are just A_{2n} and $(F_y)_k$.

The resolving method for the unknown quantities is discussed in the following. Taking the solution progress about $(F_x)_l$ and $(F_y)_l$ as example because the solutions about the $(F_x)_l$ and $(F_y)_l$ in i) and $(F_y)_l$ and A_{2n} in ii) are similar. For the case of less number of triangular patches on the adjacent region of P_l, the absolute error of the approximation to the nearest points is smaller, so the least square method is applied to approximate these adjacent points. However, when the number of adjacent triangles is big, more points are needed to approximate. The absolute error of the approximation to the nearest points is big, and it may affect the fairness of the surface. The minimum energy principle is used to optimize the fairness of the surface. Then the above two methods are used to determine the unknown quantities based on the two different case of the number of the adjacent triangles, namely the number is less than five and the number is equal to or more than five.

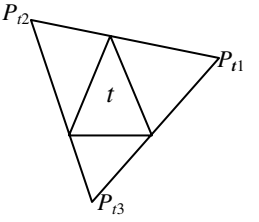


Fig. 2. the adjacent vertices on triangle t

(1) The number of triangle on adjacent region of P_l is less than five:

For the triangle t on the adjacent region of P_l in **Fig. 1**, let P_{t1}, P_{t2} and P_{t3} denote nearest points of t, see in **Fig. 2**. From the idea of establishing $F_t(x,y)$ to approximate the nearest points of $F_t(x,y)$, namely, P_{t1}, P_{t2} and P_{t3}, in order to determine $(F_x)_l$ and $(F_y)_l$, it is necessary to make the surface $F_t(x,y)$ defined on t satisfy that $\left|F_t(x_{ti},y_{ti})-F_{ti}\right|$ $(i=1,2,3)$ is minimum, so least square method is used to determined $(F_x)_l$ and $(F_y)_l$. However, different from usual method, distance weight is used in this paper. The weight is defined by the distance between the nearest points and P_l, which make that the nearer the distance between P_{ti} and P_l is, the bigger the effect on the construction of F_t. Let:

$$r_t((F_x)_l,(F_y)_l) = | F_t(x_{ti}, y_{ti}) - F_{ti} |^2 / d$$
$$= (k_1(x_{ti}, y_{ti})(F_x)_l + k_2(x_{ti}, y_{ti})(F_y)_l + k_3(x_{ti}, y_{ti}))^2$$

Where $1/d$ is the distance weight, and d is the distance between P_{ti} and P_l.

Suppose the number of triangles with vertices P_l is M, and then accumulate r_t determined on these M triangles to obtain the objective function:

$$R((F_x)_l,(F_y)_l) = \sum_{t=1}^{M} r_t((F_x)_l,(F_y)_l)$$

Minimizing $R((F_x)_l,(F_y)_l)$, then $(F_x)_l$ and $(F_y)_l$ can be determined by the following system of equations:

$$\begin{aligned} \partial R / \partial (F_x)_l &= 0 \\ \partial R / \partial (F_y)_l &= 0 \end{aligned} \tag{9}$$

The system of equations (9) has and only has one set of solutions of $((F_x)_l,(F_y)_l)$ to satisfy the approximation conditions.

(2) The number of triangle on adjacent region of P_l is more than five:

From the rule of minimum energy, it's be known that the fairness of the constructed curved surface can be improved, if the strain energy of the curved surface is minimized. If a piece of curved surface F is considered as an ideal elastic thin plate, the energy can be expressed approximately as follows:

$$\Pi = \int_s (\kappa_1^2 + \kappa_2^2) ds \tag{10}$$

where κ_1, κ_2 are curvatures. In order to avoid solving non-linear equations, the following expression replaces (10) [7].

$$\Pi = \int_s D\{(\nabla^2 F)^2 - 2(1-\mu)[F_{xx}F_{yy} - F_{xy}^2]\} ds$$

where D is the board flexible rigidity which is constant, it is determined by the board material parameter and thickness; μ is material Poisson ratio, F_{xx} and F_{yy} are the second partial derivatives, F_{xy} is the second mixed partial derivative. Usually take $D = 1, \mu = 0$, then the deformation energy of every triangle t is:

$$\Pi = \int_s (F_{xx}^2 + 2F_{xy}^2 + F_{yy}^2) ds \tag{11}$$

From the above description, F_{xx}, F_{yy} and F_{xy} are all the functions with $(F_x)_l$ and $(F_y)_l$. Supposes the number of triangles of vertices P_l is M, the objective function corresponding to P_l is:

$$E = \sum_{i=1}^{M} \int_s (F_i(x,y)_{xx}^2 + 2F_i(x,y)_{xy}^2 + F_i(x,y)_{yy}^2) ds$$

Minimizing $E((F_x)_l,(F_y)_l)$, then the unknown variables are the solution of the equations: $\begin{matrix}\partial E/\partial(F_x)_l=0\\ \partial E/\partial(F_y)_l=0\end{matrix}$.

The equations have and only have one solution $((F_x)_l,(F_y)_l)$ or $((F_x)_l,A_1)$ to satisfy the approximation conditions. $(F_y)_l$ and A_{2n} can be obtained by the same method. After the unknown variables are obtained, the triangular patches expression can be determined. Combine the triangular patches $F_t(x,y)$, $t=1,2,\cdots,M$ on the adjacent regions of point P_l directly to obtain the polynomial surface $f_l(x,y)$ with C^1 continuity. From the construction method of $f_l(x,y)$, whichever data point is adjusted, only the triangular patches on its adjacent region are influenced, therefore the whole surface has better local adjustability.

2.2 Constructing Final Triangular Patch F_{final} on Triangle

For each triangles of the whole triangle network, three quadric polynomial surfaces with C^1 continuity at vertex P_i , P_j and P_k of *s*. Then the final surface patch $F_{final,s}(x,y)$ on *s* is obtained by weighted combination of the three quadric patches at the vertices of the triangle *s*:

$$F_{final,s}(x,y)=w_i(x,y)f_i(x,y)+w_j(x,y)f_j(x,y)+w_k(x,y)f_k(x,y)$$

Where $f_i(x,y),f_j(x,y)$ and $f_k(x,y)$ are quadric polynomial surfaces at the vertices of triangle *s*, respectively, $w_i(x,y),w_j(x,y)$ and $w_k(x,y)$ are weight functions. In order to ensure that the constructed curved surface achieves C^1 continuity, the weight functions must satisfy the following conditions:

I) The weighted triangular patch also passes through the three vertices.
II) The first degree partial differentials at P_i , P_j and P_k of each weighted patch are also ($\partial f_i/\partial x$, $\partial f_i/\partial y$), ($\partial f_j/\partial x$, $\partial f_j/\partial y$) and ($\partial f_k/\partial x$, $\partial f_k/\partial y$), respectively.
III) The weighted triangular patch is C^1 continuous on the boundary.
So the weight functions are expressed as follows:

$$w_i(x,y)=\{10L_i-15L_i^2+6L_i^3+30L_jL_k(Q_kL_k/D_{ik}+Q_jL_j/D_{ij})\}L_i^2$$
$$w_j(x,y)=\{10L_j-15L_j^2+6L_j^3+30L_kL_i(Q_iL_i/D_{ij}+Q_kL_k/D_{kj})\}L_j^2$$
$$w_k(x,y)=\{10L_k-15L_k^2+6L_k^3+30L_iL_j(Q_jL_j/D_{jk}+Q_iL_i/D_{ik})\}L_k^2$$

where:

$$Q_i=(x_j-x_i)(x_k-x_i)+(y_j-y_i)(y_k-y_i) \qquad D_{ij}=(x_j-x_i)^2+(y_j-y_i)^2$$
$$Q_j=(x_i-x_j)(x_k-x_j)+(y_i-y_j)(y_k-y_j) \qquad D_{ik}=(x_k-x_i)^2+(y_k-y_i)^2$$
$$Q_k=(x_i-x_k)(x_j-x_k)+(y_i-y_k)(y_j-y_k) \qquad D_{jk}=(x_j-x_k)^2+(y_j-y_k)^2$$

3 Experiments

In this section, the six bi-variate functions proposed by Franke[15] are used in the comparison experiments, which are $F_1(x, y)$, $F_2(x, y)$, $F_3(x, y)$, $F_4(x, y)$, $F_5(x, y)$ and $F_6(x, y)$ respectively. These experiments are used to compare the new method with the method proposed in [3]. For the triangular network shown in **Fig.3**, the new method and the method in reference [3] are used to construct interpolant surface, respectively. The surfaces shown in **Fig.4~Fig.8** are combined by the patches of the triangles, where the surfaces in (a) are linear interpolant patches on each triangles, and in (b) are the surfaces constructed by the new method, the surfaces in (c) are constructed by the method proposed in reference [3]. For the former two functions of the six functions, the surfaces constructed by the new method and the method in [3] are unsatisfactory because the two functions are both complicated, and the interpolation conditions on the triangular network is not enough(for example, see **Fig.4** the interpolant surfaces defined by F_1). For the latter four functions, namely:

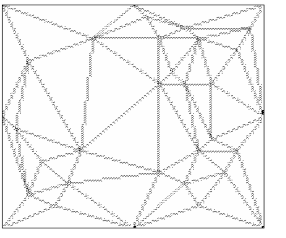


Fig. 3. Triangulation of 33 points

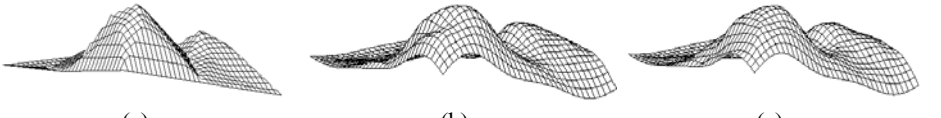
$$F_3(x, y) = 5.2(1.25 + \cos(5.4y))/(6 + 6(3x-1)^2)$$

$$F_4(x, y) = 5.2\exp(-81((x-0.5)^2 + (y-0.5)^2)/16)/3$$

$$F_5(x, y) = 5.2\exp(-81((x-0.5)^2 + (y-0.5)^2)/4)/3$$

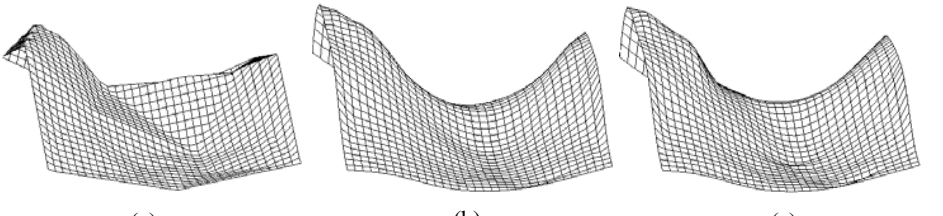
$$F_6(x, y) = 5.2sqrt(64 - 81((x-0.5)^2 + (y-0.5)^2))/9 - 2.6$$

The surfaces interpolated to $F_3(x, y) \sim F_6(x, y)$ are shown in **Fig.5~Fig.8**. From the comparison of the Figures we can see, the interpolant surfaces constructed by the new method are more fairing and have the shape suggested by the original data points.



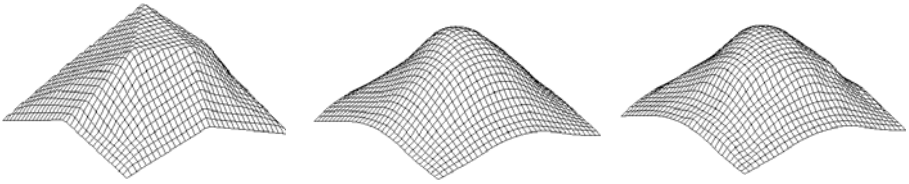
(a) (b) (c)

Fig. 4. The interpolant surfaces defined by F_1



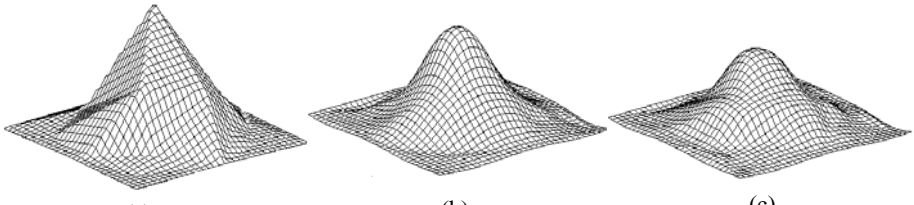
(a) (b) (c)

Fig. 5. The interpolant surfaces defined by F_3



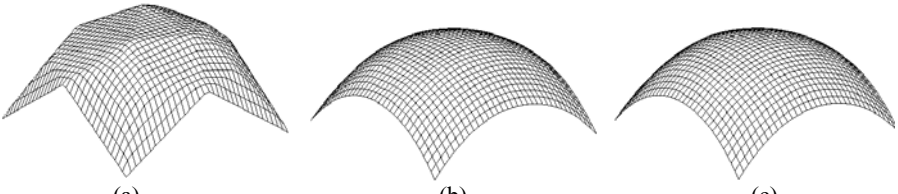
(a) (b) (c)

Fig. 6. The interpolant surfaces defined by F_4



(a) (b) (c)

Fig. 7. The interpolant surfaces defined by F_5



(a) (b) (c)

Fig. 8. The interpolant surfaces defined by F_6

Based on the triangulations shown in Fig3, the maximum error of the two methods for interpolating $F_3(x, y) - F_6(x, y)$ are given in table 1.

Table 1. Maximum errors generated by the two methods

Maximum Error	$F_3(x, y)$	$F_4(x, y)$	$F_5(x, y)$	$F_6(x, y)$
New method	0.272	0.151	0.532	0.057
Method in [3]	0.370	0.164	0.843	0.061

4 Conclusion

The new method can construct surface which interpolates the given scattered data points. The surface patch on the adjacent region of each vertex of the triangle is C^1 piecewise quadric interpolation patch. The solution to the quadric piecewise polynomial surface is classified as two cases by the number of triangles of the adjacent

region. In the first case, the method of least square combining distance weight is used to resolve the unknown variables of the polynomial surface. In the second case, objective function of minimum energy is established to determine the unknown variables. The surface has the properties that it has better shape suggested by the given data points and the shape is adjustable over the local region. The polynomial precision set of the methods includes all the polynomials of degree two.

References

1. C.Y. Wang, Zhang C.M, Polynomial of degree four interpolation on triangles. Journal of Computational Mathematics, (1991), 19(2):155-161
2. Powell M JD, Sabin M A. Piecewise quadratic approximations on triangles. ACM Transaction on Mathematical Software, (1977), 3(4):316-325
3. Wang jiaye, Zhang Caiming. Interpolation From Scatter Data Points. Applied Mathematics, A journal of Chinese Universities, (1987), 2(4): 456-461
4. Grandine T A. An iterative method for computing multivariate C1 piecewise polynomial interpolants. Computer Aided Geometric Design, (1987), 4:307-319
5. D Terzopoulos, J Platt, A Barr, er al Elastically deformable models[J]. Computer Graphics, 21(4), (1987), 205-214
6. G celniker, D Gossard Deformable curve and surface finite elements for free-form shape design[J]. Computer Graphics, 25(4), (1991), 257-266
7. Cai Zhongyi, Reconstruction and Smoothing of Surface Based on Minimum Energy Principle, Journal of Computer-Aided Design & Computer Graphics, 14(8), (2002), 758-762
8. Barnhill R E, Birkhoff G, and Gordon W J. Smooth interpolation in triangles. J. Approx. Theory, (1973), 8:114-128
9. Nielson G M. The side vertex method for interpolation in triangles. J. Approx. Theory, (1979), 25:318-336
10. Gregory J A. Smooth interpolation without twist constraints. In: Barnhill R E and Riesenfeld R F, eds. Computer Aided Geometric Design, Academic Press. New York, (1974). 71-88
11. C.Y. Wang. C1 Rational Interpolation over an Arbitrary Triangle. Computer-Aided Design, (1983), 15:33-36
12. C Zhang, T Agui, H Nagahashi and Tomoharu Nagao, “A New Method for Smooth Interpolation without Twist Constraints", IEICE Trans, INF & Syst.,(1993), Vol. E76-D(2), 243-250,
13. C Zhang and Fuhua Cheng, “Triangular patch modeling using combination method”, Computer Aided Geometric Design, (2002), Vol. 19(8), 645-662;
14. Zhang C, Ji X and Yang X, Constructing triangular patch by basic approximation operator plus additional interpolation operator, Science in China F, (2005), 48(2), 263-272.
15. Franke, R., A critical comparison of some methods for interpolation of scattered data. Naval Postgraduate school, Tech. Report NPS-53-79-003, (1979)

Detection and Localization of the Top Object in the Stack of Objects

Hernsoo Hahn and Youngjoon Han

School of Electronics, Soongsil University,
511, Sangdo-Dong, Dongjak-Ku, Seoul, 156-743 Korea
{hahn, young}@ssu.ac.kr
http://visionlab.ssu.ac.kr

Abstract. This paper proposes a new algorithm that detects the top object in a stack of arbitrarily overlapped objects, using the fact that the boundary edges of an occluding surface are not disconnected by other edges. To determine whether a surface is an occluding one or occluded one, this paper represents the objects in an input image using the attributed relation graph where a node represents a surface and an arc connecting two nodes shows the adjacency of the two surfaces in the image. An arc is weighted by two relation values, each of which is weighted on a node and tells the number of edges shared with the surface connected in the opposite side of the edge. Based on the magnitudes of the relation values in the attributed graph, all surfaces are classified into either occluding or occluded ones and grouped as a node set. The top object is selected as the result of the merging process of the node sets. The experimental results have shown that the proposed algorithm efficiently separates the top object in the various images of object stacks, with the time complexity of O(n) where n is the number of edges in the image.

1 Introduction

Occlusion among objects frequently occurs in the process of part assembly, part feeding, or inspection lines. Many researchers have proposed various solutions for this problem which can be categorized into two approaches: one is of using 2D information[1,2] and the other one is of using 3D information[3,4] from the image. Most of those approaches are based on the model matching technique which searches for the features of the object model in the given 2D or 3D images and decides the object as the model if it contains them.

In case of using 3D information, the geometric and spatial features such as volumetric primitives, normal vector of a plane and distances between faces are being used to decide the occlusion and to recognize the objects. Those works using 3D information simplify the recognition process compared to the works using 2D images, since the representation of 3D objects using 3D features is natural and much easier than using 2D features. However, these approaches cost too much to process 3D data and to construct 3D image.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 116 – 125, 2006.
© Springer-Verlag Berlin Heidelberg 2006

On the other hand, 2D image is easier to obtain and less sensitive to noise than 3D image, though it loses the depth information. When dealing with a 2D image, boundary image is popularly used to represent the object and to find the best matching object model. For example, Edwards and Murase[5] assumed the possible types of occlusions between objects and made 13 binary masks corresponding to individual types. In the first step, they used the model primitives with lower resolution to obtain the correlation with the objects in the image and thus generated more than one hypothesis on the object's model and pose. The resolutions of the primitives are increased and the matching process is iterated until the number of hypotheses converged to one. Pikaz and Dinstein[6] proposed the matching algorithm between two curves, which have the relation of similarity transformation with each other. Each curve is represented by a total curvature graph, which is the relation graph showing the distance and the change of gradient from any point on the contour, and then the gradient axis of both graphs are sampled periodically. The matching process between the model and the test object uses the rate of two parameters. One is the length of distance axis that corresponds to the sampled value on the gradient axis and the other is the cumulative length on the distance axis at that point. The performance of these types of methods varies depending on the amount of occlusion, so it can bring about lower performance in recognition.[7] These approaches tend to increase computation time as the number of model increases.

This paper proposes a new approach for separating the occluding and occluded objects to simplify the recognition process of the overlapped objects by selecting the occluding one before the matching process. The basic idea was originated from the fact that human detects the occluding object first and then assumes the model of the occluded object based on the shape of the contour of edges, when recognizing the overlapped objects. For this purpose, this paper uses an attributed relation graph and proposes a new method of analyzing the relation graph to determine which object is the top one in the stack of objects. In this paper, the surfaces are assumed to be planar, cylindrical, or spherical.

2 Attributed Relation Graph

3D objects are projected on a 2D image plane where the objects appear as a combination of the 2D surfaces located forwarding the optical center of a camera. To decide which one is occluding or occluded, the objects are represented by an attributed relation graph where the surfaces are represented by the nodes and the adjacent surfaces are connected by the arcs. Here each arc has two relation values on both ends enumerating the type of adjacency of the nodes.

The relations between two faces in a 2D image are categorized into three types: 'connected', 'separated' and 'overlapped'. 'Connected' is defined as the case that two faces are sharing one common edge, such as (A,B), (A,C), and (B,C) in Fig. 1(a). 'Separated' is defined when two faces are sharing no edge, such as (A,E), (A,D) and (F,E). When two faces are not 'separated' and the number of shared edge is not 1, then 'overlapped' is defined. For example, the edge of the occluding face (A) is intersected

by that of the occluded face (F) to generate a 'T' junction. These relations are described by the relation graph. Two examples of relation graph are given in Fig. 1(b) and (c). Fig. 1(b) shows the relation graph of three surfaces (A,B,C) where the relation values on three arcs are all 1 since each two of them are sharing one edge. When a relation value is one, then it is not written for convenience. Fig. 1(c) shows the relation graph of three surfaces (C, F, D) where the both ends of the edge between C and F are weighted with a relation value 2 since two faces are sharing two edges each other.

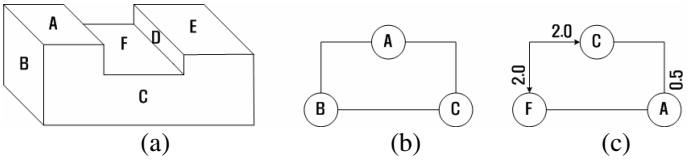


Fig. 1. Types of relations between two faces

Formally, relation value *m* to be weighted on one end of an arc is defined as follows:

$$m = E + S \tag{1}$$

where *E* is the number of edges sharing with the surface at the other end of the arc, and *S* is the inverse of the number of intersected points on the boundary edge with the face at the other end of the arc added by 1. For example, any two of three surfaces (A,B,C) in Fig. 1(a) are sharing one edge without intersected point so that all arcs in Fig. 1(b) are weighted by 1. In the case of two surfaces (A,F), the edge of A adjacent to F is intersected by the edge of F, and thus the relation value on the end of the arc near A becomes 0.5. It can be interpreted here as surface A is occluding surface F. Some examples showing how to determine the relation values are given in Fig. 2.

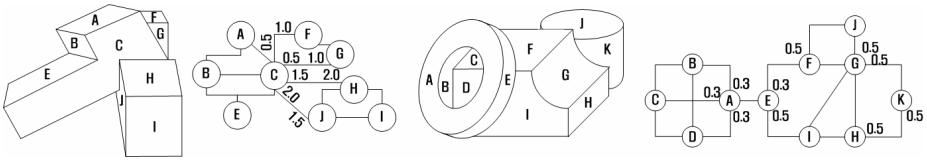


Fig. 2. Examples of determining the relation values between two faces

When two surfaces are in the occlusion relation, then the arc between two nodes representing these two surfaces are changed to an arrow. The node with a smaller relation value occludes the node with a larger one. Thus the arrow forwards from the node with the smaller relation value to the node with the larger relation value. If two relation values are same then both are occluding each other.

3 Separation of Occluding and Occluded Objects

Once the attributed relation graph of the surfaces included in an input image is obtained, then the surfaces pertained to the same object should be segmented and also the occlusion relation between the objects should be determined. This process is illustrated by the block diagram in Fig. 3.

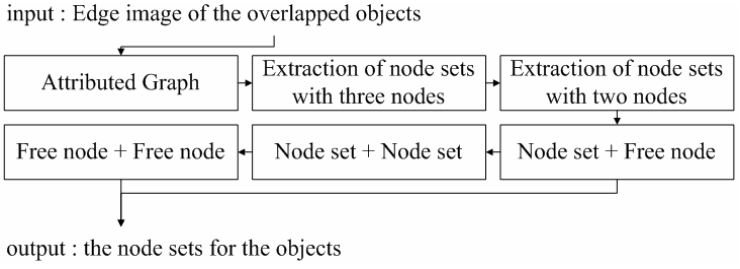


Fig. 3. Flow of the algorithm

For determining the surfaces pertaining to the same object, a '3-node set' is defined first as the set of three nodes where three arcs connecting two of them are all 'connected' types as shown in Fig. 4(a). This is to consider the fact that an object can be a polyhedra. The surfaces included in a 3-node set are considered to pertain to the same object. In the next step, a '2-node set' is defined for two nodes with a relation of 'connected' type if they are adjacent to the same third node and they have the same type of relation with the third node as shown in Fig. 4(b). Those nodes not included in any node set are called 'free nodes'.

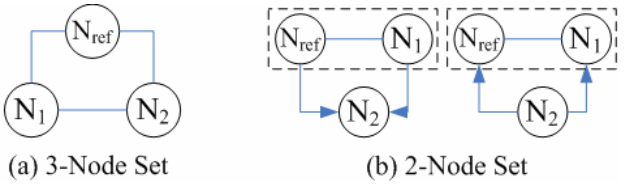


Fig. 4. Extraction of the node sets from attributed graph

If all the node sets are found in the attributed relation graph, then the merging operation begins to find the nodes that should have been included in any node set but failed, among the free nodes. They can be merged either with one of the node sets already found or with other free node(s).

Merging a free node(N_{ref}) with a 3-node set (Refer to Fig. 5)

Step 1: Among the nodes connected to N_{ref}, eliminate those nodes connected by the arcs representing T- or $\prod$-junctions those nodes are occluded by N_{ref} .

Step 2: Count the number of arcs connected to the nodes included in a 3-node set NS_T already extracted.

Step 3: Test whether N_{ref} is connected to NS_T with a single arc and it is not connect to any other node. If so, merge N_{ref} to NS_T.

Step 4: If the number of arcs connected to NS_T is equal to or greater than 2, test whether any three nodes including N_{ref} and two from NS_T make a 3-node set. If so, merge N_{ref} to NS_T. If they make the relation graph given in Fig. 8(a), then N_{ref} is merged to NS_T too.

Step 5: If the number of arcs connected to NS_T is equal to 1, make a group of three nodes including N_{ref}, one node N_1 from NS_T, and the third node N_2 which is connected to both of N_{ref} and N_1. If these three nodes form one of the four relation graphs in Fig. 5(b)-(e), then N_{ref} and N_1 are merged to NS_T.

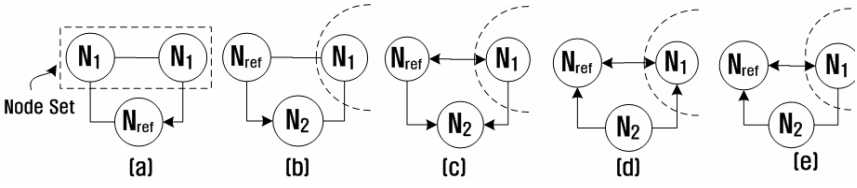


Fig. 5. Merging a 'free' node to a node set

If there lefts no 'free' node and node sets to be merged, test whether different node sets are pertained to the same object. Here each node set becomes reference node set NS_{ref} in turn.

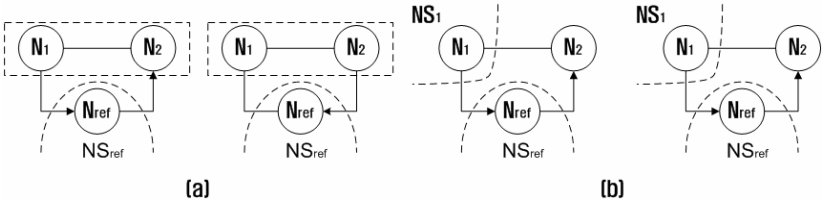


Fig. 6. Merging of two node sets

Merging two node sets (Refer to Fig. 6)

Step 1: Count the number of arcs connecting two node sets, NS_{ref} and NS_1. If both relation values of an arc are larger than 1 and their difference is 0.5, then it is not counted in, since such nodes are pertained to different objects.

Step 2: If the number is equal to or greater than 2, test whether any three 'connected' nodes construct a 3-node set as shown in Fig. 6(a). If passed, then merge those two node sets.

Step 3: If the number of arcs connecting NS_{ref} and NS_1 is 1, then make a group of three nodes, one node N_1 from NS_{ref}, one node NS_2 from NS_1, and the third one from other node set which is adjacent to both of NS_1 and NS_2. Test

whether these three nodes construct a 3-node set as shown in Fig. 6(b). If passed, then merge those two node sets.

In the last step, it is tested whether the remained 'free' nodes can be merged to construct a node set.

Merging two 'free' nodes (Refer to Fig. 7)

Step 1: Count the number of arcs connecting a 'free' node N_{ref} to other 'free' nodes N_i.

Step 2: If both N_{ref} and N_1 are connected to only one other node respectively, then construct a new node set with these two nodes.

Step 3: If the number of arcs connected to N_{ref} is equal to or greater than 2, then make a group of three nodes including N_{ref} and two connected nodes. If these three nodes construct any pattern given in Fig. 7, then merge those nodes to make a new node set.

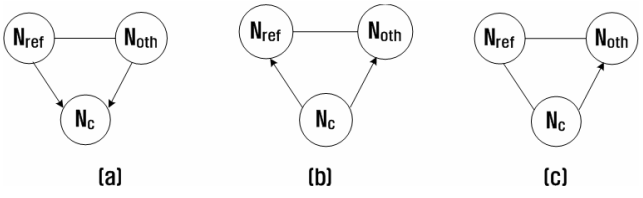


Fig. 7. Merging of two 'free' nodes

The process of separating the top object from a stack of object using the relation graph is illustrated using the image given in Fig. 8. It extracts the node sets which have 'basic' pattern or 'equal-position' pattern in the attributed relation graph. The node set {C, D, E} is an example that forms a 'basic' pattern. In the first iteration, it found that node A is 'isolated' with reference to node B and they satisfy the merging condition to form a two node set. In the second iteration, it found that node G is 'isolated' with reference to the node set {A, B} and they also satisfy the merging condition to form a three node set {A, B, G}. In the same way, node F is merged to the node set to form a new node set {A, B, F, G}. It could also be found that three nodes, E, B, and F, formed a connected graph where the relation values of the arcs connecting two node pairs, F and E, and F and B, satisfied the merging condition. However, the node pair, F and E, do not have a third node which is adjacent to both nodes and thus they do not satisfy the condition to make a node set. Therefore, the arc connecting node F and the node set {C, D, E} could be eliminated. In other words, node F was isolated from the node set {A, B} and satisfied the sufficient condition to be 'self-occluded' pattern with the node set {A, B}, these three nodes could be merged as a new node set. Since node B in the node set {A, B, F, G} is occluding the surfaces C and E included in the node set {C, D, E}, it is determined that the object including the node set {A, B, F, G} is the top object.

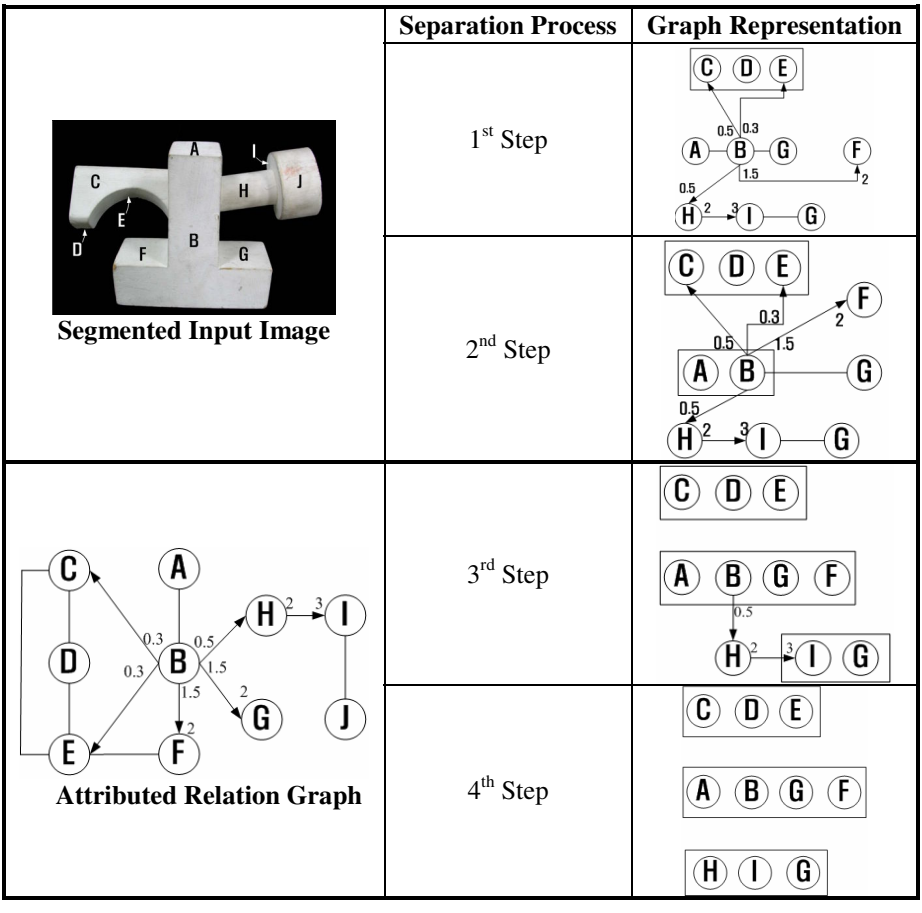


Fig. 8. The process of separating the top object from a stack of objects using the attributed relation graph

4 Experiment and Discussion

For the experiment, the proposed algorithm has been implemented using Visual C^{++} in the Pentium-IV PC. For the experiments, total 40 input images have been used where some of 8 different objects are arbitrarily overlapped. Eight object models used in the experiments are shown in Fig. 9. The algorithm process an input image following the flow illustrated in Fig. 3.

In the first step, the boundaries of the objects in an input image are extracted, using a region growing algorithm. The segmented regions are merged based on the Mahalanobis distance, proximity between them, and edge existence. Mahalanobis distance is defined as follows:

$$d_M(X_A, X_B) = \frac{d_E(C_R(X_a), C_R(X_B))}{\sqrt{\sigma_R(X_A) + \sigma_R(X_B)}} \tag{2}$$

where X_A and X_B represent segmented regions A and B, respectively, and $C_R(X)$ is the averaged grey level of region X. d_E and $\sigma_R(X)$ represent Euclidean distance and the standard deviation of region X, respectively. The segmented boundary images of the input images are given in Fig. 10(a).

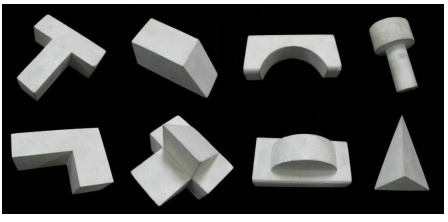


Fig. 9. Eight object models used in the experiments

In the second step, the attributed relation graph is extracted from the boundary image as shown in Fig. 10(c) to which the surface separation procedure is applied. The separated node sets are also included in Fig. 10(d). Fig. 10(e) shows the finally separated occluding object.

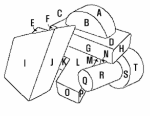
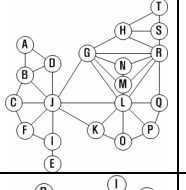
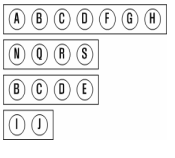
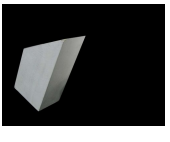
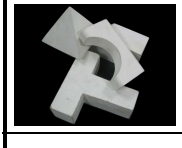
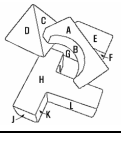
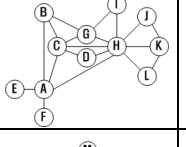
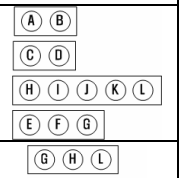
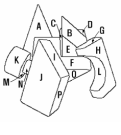
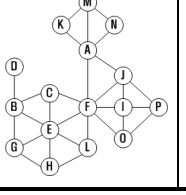
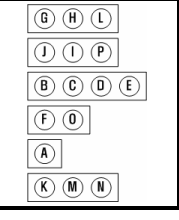
(a) Original image	(b) Edge image	(c) Attributed relation graph	(d)Node sets	(e)Selected top object
				
				
				

Fig. 10. Intermediate and final results of the proposed algorithm

For evaluation of the performance of the algorithm, the time complexity is measured. The proposed algorithm selects every surface as a reference one and tests its relationship only with its adjacent surfaces. Thus, the time complexity of the algorithm becomes $O(nm)$, where n is the number of surfaces in the image and m is the average number of adjacent edges of a surface.

To analysis how much time is spent more in the preprocessing of an image by an inclusion of this process, the times spent for all procedures in the preprocessing step are measured and summarized in Table 3. As can be found in Table 3, the total preprocessing time increase less than 40% on average by an addition of the proposed algorithm.

Table 3. The times spent for individual procedures of the preprocessing step, applied to the three images in Fig. 10(unit: seconds)

Preprocessing Step		1st image	2nd image	3rd image
Common Procedures	Edge Detection	13.61	20.13	15.08
	Line finding	0.97	0.8	0.73
subtotal		14.58	15.81	20.21
Added (proposed) Procedure	ARG construction	7.14	6.92	4.64
	ARG Segmentation	0.015	0.015	0.025
subtotal		7.16	4.67	6.94
Total time		21.72	20.7	30.865

5 Conclusion

This paper proposed a noble approach of separating an occluding object from the occluded one(s). By separating the occluding object from the occluded objects, the matching problem of multiple overlapped objects will be simplified as a matching task of individual single object. The proposed algorithm proposed for this purpose represented the object image using an attributed relation graph and the occlusion relation between two surfaces is enumerated by the relation values. By analyzing the relation values based on the topological features of an occluding and occluded edge set, the attributed relation graph can be segmented so that the nodes pertained to the same objects may be included in the segmented relation graph.

The performance of the proposed algorithm has been tested with various 2D images where some of 8 objects are arbitrarily overlapped. The experimental results have shown that it efficiently separates the top (occluding) object in most cases and that the addition of the proposed algorithm increases less than 40% of the conventional preprocessing algorithm on average.

Acknowledgement

This work has been supported partially by Soongsil University and Center for Intelligent Robot.

References

1. Dockstader S.L., Tekalp A.M.: Tracking multiple objects in the presence of articulated and occluded motion. Human Motion, 2000 Proceedings, Workshop on. (2000) 88-95.
2. Stan Z. Li: Recognizing Multiple Overlapping Objects in Image: An Optimal Formulation. IEEE Transactions on Image Processing, Vol. 9, No 2. (2000) 273-277.
3. A. Johnson and M. Hebert: Efficient Multiple Model Recognition in Clustered 3-D Scenes. In Proceedings of IEEE Coference on Computer Vision and Pattern Recognition. (1998) 671-677.
4. Umberto Castellani, Salvatore Livatino, Robert B. Fisherge: Improving Environment Modelling by Edge Occlusion Surface Completion. 3D Data Processing Visualization and Transmission, 2002 Proceedings, First International Symposium on. (2002) 672-675.
5. J. Edwards and H. Murase: Appearance Matching of Occluded Objects Using Coarse-to-fine Adaptive Masks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1997) 533-539.
6. A. Pikaz and I. Dinstein: Matching of Partially Occluded Planar Curves. Pattern Recognition, Vol. 28, No. 2. (1995) 199-209.
7. R. Dinesh and D.S. Guru: Recognition of Partially Occluded Objects Using Perfect Hashing: An Efficient and Robust Approach. In Proceedings of the second Canadian Conference on Computer and Robot Vision, (2005) 1-8.
8. Wen-Jing Li, Tong Lee: Object recognition by sub-scene graph matching. Robotics and Automation ICRA, IEEE International Conference on, Vol. 2. (2000) 1459-1464.

Recognition of 3D Object Using Attributed Relation Graph of Silhouette's Extended Convex Hull

Hernsoo Hahn and Youngjoon Han

School of Electronics, Soongsil University,
511, Sangdo-Dong, Dongjak-Ku, Seoul, 156-743 Korea
{hahn, young}@ssu.ac.kr
http://visionlab.ssu.ac.kr

Abstract. This paper presents a new approach of recognizing a 3D object using a single camera, based on the extended convex hull of its silhouette. It aims at minimizing the DB size and simplifying the processes for matching and feature extraction. For this purpose, two concepts are introduced: extended convex hull and measurable region. Extended convex hull is consisted of convex curved edges as well as convex polygons. Measurable region is the cluster of the viewing vectors of a camera represented as the points on the orientation sphere from which a specific set of surfaces can be measured. A measurable region is represented by the extended convex hull of the silhouette which can be obtained by viewing the object from the center of the measurable region. Each silhouette is represented by a relation graph where a node describes an edge using its type, length, reality, and components. Experimental results are included to show that the proposed algorithm works efficiently even when the objects are overlapped and partially occluded. The time complexity for searching the object model in the database is $O(N)$ where N is the number of silhouette models.

1 Introduction

In recognition of 3D object, the most important problem is how to represent the object models so that they can be recognized with only partial information. Various representation schemes can be used depending on which types of sensors are used. A stereo camera or a single camera with structured light looks the proper sensor to deal with 3D data. However, these types of sensors cannot be a practical solution when the sensor positions and the background changes frequently.[1] Although a single camera may cause many difficulties for performing this task since one dimension of data is reduced, it has still the strong advantage that it is simple to use and to process the data. This paper deals with the recognition problem using a single camera to emphasize these advantages.

The object presentation schemes, introduced thus far to recognize 3D objects using a single camera, can be categorized into two groups. The first one is based on the appearance of an object which uses the shape information itself. It collects the 2D images of a 3D object from all possible viewing directions and uses those images as the templates which include the contour and texture information. Those information can be used as the advantageous features in the matching process since they can be

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 126 – 135, 2006.
© Springer-Verlag Berlin Heidelberg 2006

acquired from the input image with little processing, as well as the weak points since too many contours should be reserved in the database and the texture information is influenced too much by the lighting conditions.[2] Another one is based on the structural features of an object. They are usually described by a relation graph where the edges or surfaces are represented by the nodes and their connectivity or adjacency are represented by the arcs.[3,4] Since this type of representation scheme describes an object using the relative relations of the features, it is less sensitive to the rotation and size variation. It is also appropriate for the case where the objects are partially occluded. However, the complexity of relations among the nodes increases in a combinatorial way and thus the searching time when the nodes in a relation graph increases. To solve this problem and utilize the strong points of the relation graph representation scheme, the graph matching algorithms based on nonlinear approximation[5,6] and eigen-vector analysis[7,8] have been proposed. The matching scheme based on an eigen-vector analysis has shown a fast matching time and also a robust performance when there exist partial variations in shape, distortion and noise in the extracted features.[8] One major problem which hinders this scheme from applying to general recognition problem is that the number of nodes of the objects in the database and that of the test object should be same. Other approaches based on a relation graph suffer from the similar difficulties so that the application becomes very restricted.

This paper aims at achieving two goals. The first one is to minimize the DB size and thus to reduce the matching time by using simple features. The second one is to reduce the processing time necessary for extracting the features in a 2D image. For these purposes, this paper represents a 3D object with a minimum number of silhouettes, each of which is consisted with a unique set of surfaces. A silhouette is refined by the extended convex hull which is represented by a circular relation graph where a node corresponds to an edge in the extended convex hull and an arc illustrates the adjacency of two nodes. Each node is weighted by four features of the edge such as the type, length ratio, reality, and the number and types of the arcs covered by the edge if its type is 'virtual'. An arc is also weighted by the angle between adjacent edges. To find the minimum number of silhouettes to represent an object, the measurable region of an object is defined on the orientation sphere and segmented by the measurable regions of the different sets of surfaces.

2 Extended Convex Hull

Convex hull is the smallest polygon or polyhedra which includes a given 2D or 3D object, defined as follows:

Definition 1: The convex hull of a finite point set S = {P} is the smallest 2D polygon W (or polyhedron in 3D) that contains S. That is, there is no other polygon (or polyhedron) L smaller than W while containing the set S.

By the definition, the conventional convex hull has to enclose the object with a polygon (or polyhedra) even when the object has convex curved edge (or surface) as shown in Fig. 1(a). The edges or surfaces covering concave edges or surfaces are called virtual edges or surfaces here. Since the areas covered by the virtual edges in a convex hull are unique if the object is so, convex hull representation has been used for

object recognition. However, since the conventional convex hull asks to enclose an object with only a polygon(or polyhedra), it should spend time to find the polygon even for enclosing convex curved edge(or surface). To solve this problem, extended convex hull is defined here as follows.

Definition 2: The extended convex hull of a finite point set S = {P} is the shortest 2D convex closed line W_E (or convex object in 3D) that contains S. That is, there is no shortest convex line (or convex object) than W_E.

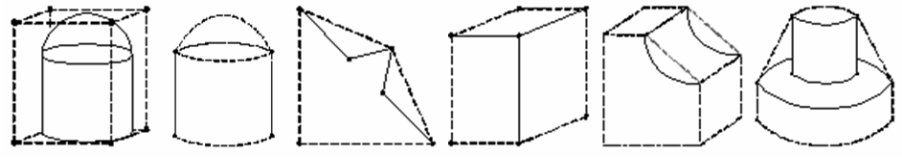


Fig. 1. Convex hull and extended convex hull representatins of 2D and 3D objects

The extended convex hull may have the same shape with the conventional convex hull if an object is consisted of polygons or concave surfaces, as shown in Fig. 2(c)-(e). However, if an object includes convex curved surfaces, the extended convex hull has a different shape from its conventional convex hull, as shown in Fig. 2(b) and (f).

3 Measurable Region of a Surface in Orientation Sphere

A 3D object projects different sets of surfaces and thus generates different silhouettes to the 2D image plane normal to a viewing direction. A silhouette is consisted of those edges of the projected object in the 2D image plane making the boundary with the background. This means that if all the silhouettes acquired from all possible viewing directions are provided then the object can be completely reconstructed. However, acquisition of silhouettes from all possible viewing directions is impossible, since the number of silhouettes will be at least 360^2 even dividing the viewing direction by one degree distance. Some practical programs used in industry reduce this number to a couple of thousands silhouettes and they have shown reasonable performance.[9]

To significantly reduce this large number of silhouettes needed to represent a 3D object, this paper proposes to represent those silhouettes having the same set of surfaces by the representative silhouette. For this purpose, the measurable regions of individual surfaces included in an object are defined on the orientation sphere and then the orientation sphere is segmented by the measurable regions so that each region may have a different set of surfaces. Each region is represented by the center point of the region on the orientation sphere. The silhouette acquired from that center point (viewing direction) is considered as the representative silhouette of the measurable region.

Here the measurable region of a surface is defined as the collection of those viewing directions on the orientation sphere from which even a part of the surface can be detected. A viewing direction on the orientation sphere is defined by a set of two

angles (α, β) where α is the angle apart from the X axis in the counter clockwise in the X-Y plane and β is the angle apart from the X-Y plane in the counter clockwise. It is assumed that the viewing position is located far enough from the object. Fig. 3 shows the measurable region of a plane described as a hemisphere. Then if the normal of a surface is n, then the surface can be measured by the camera whose optical axis is c when the camera is located on the orientation sphere satisfying the following condition:

$$\begin{aligned} & ax + by + cz - 1 > 0 \\ & \alpha = \tan^{-1} \frac{y}{x} \\ & \beta = \tan^{-1} \frac{z}{\left(x^2 + y^2\right)} \end{aligned} \tag{2}$$

where *n=(a, b, c)*. The collection of all points on the orientation sphere satisfying this condition forms a hemisphere.

Measurable regions of curved surfaces are also determined in the same way. Curved surface is either convex or concave. For a convex curved surface, the measurable regions for all surface normal vectors on all points of the surface are calculated and overlapped as illustrated in Fig. 2(a) and (b). Convex curved surface can be seen by a camera when the surface normal of any point on the surface is located within 90^o with reference to the viewing direction (optical axis) of the camera. The measurable regions of concave curved surfaces are derived, differently from the cases of convex ones, from the extended convex hull of the surfaces. The silhouettes of concave curved surfaces are formed by actual boundary points. Fig. 2(c) shows that the measurable region of a concave spherical surface is defined as the one of the plane covering the spherical surface. Fig. 2(d) shows that the convex hull of a concave cylindrical surface needs three more virtual surfaces, one is to cover the concave surface and two are to cover the circular edges. Since two virtual surfaces covering the circular edges do not contribute to form the silhouette, the surface normal of the virtual surface covering the concave surface is used to determine the measurable region of the concave cylindrical surface.

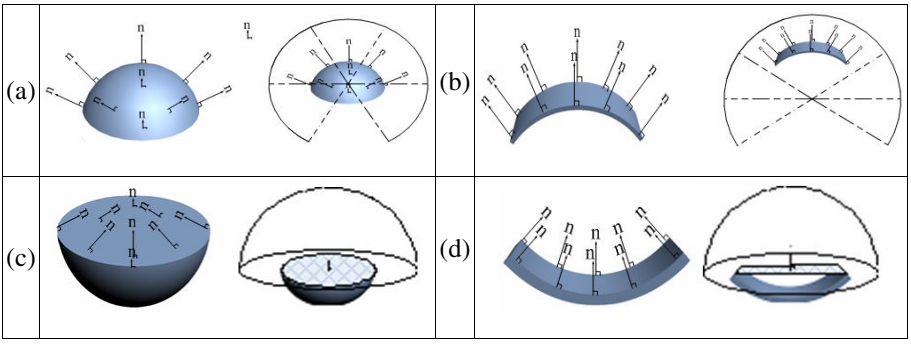


Fig. 2. Measurable regions of convex and concave curved surfaces using extended convex hull

The process of determining the measurable region of an arbitrary surface can be summarized as follows:

Determination of the measurable region of a surface on the orientation sphere

Step 1: Find the normal vectors at all the points of a surface. If all the normal vectors are identical, go to step 2. If the surface is convex curved one, go to step 3. If the surface is concave curved one, go to step 4.

Step 2: If all the normal vectors are identical, then the surface is a plane. In this case, those points on the orientation sphere satisfying Eq. (1) with the surface normal are labeled with the surface index.

Step 3: For every normal vector, its measurable region is determined using Eq. (1) with the normal vector. All points on the orientation sphere are labeled with the same surface index. Any point on the orientation sphere labeled with the index of the surface even once is included in the measurement region.

Step 4: Find the extended convex hull of the surface first and then the virtual surface forming the silhouette. Those points on the orientation sphere satisfying Eq. (1) with the surface normal of the selected virtual surface are labeled with the surface index.

4 Determination and Segmentation of Measurable Region of a 3D Object

A 3D object used in industry is constructed by combining a set of known surfaces following a specific combination rule. It may have convex or concave shape(s) in it. When an object contains only convex shapes, then the measurable region of the object can be obtained by overlapping the measurable regions of individual surfaces included in the object. If an object contains concave shapes, then its extended convex hull is generated and the measurable regions of the surfaces forming the concave shape are represented by the measurable region of the virtual surface covering them.

As shown in Fig. 3(a) and (b), the measurable region of a convex object is obtained by overlapping the individual measurable regions of the surfaces. To determine the measurable region of a 3D object having concave shapes, its extended convex hull is derived first. For example, S7 and S8 of the object in Fig. 3(c) making a concave shape are covered by the virtual surface S7&8 in the extended convex hull. In Fig. 3(d), the concave shape formed by two surfaces S2 and S3 is eliminated by the virtual surface S2&3 in the extended convex hull. Then the measurable regions of these objects are generated in the same way that has been done for a convex object case.

Once the measurable regions of all surfaces of an object are decided on the orientation sphere, then the orientation sphere can be segmented by the measurable regions in such a way that each one should have different set of surfaces. Since a point (viewing direction) on the orientation sphere is addressed by (α, β), the process of this segmentation is to select the addresses containing the same set of labels. Fig 4 shows

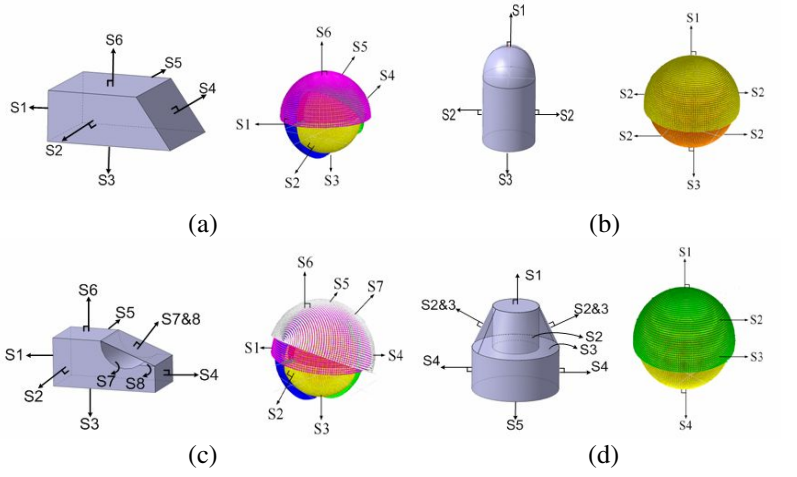


Fig. 3. Measurable regions of 3D objects

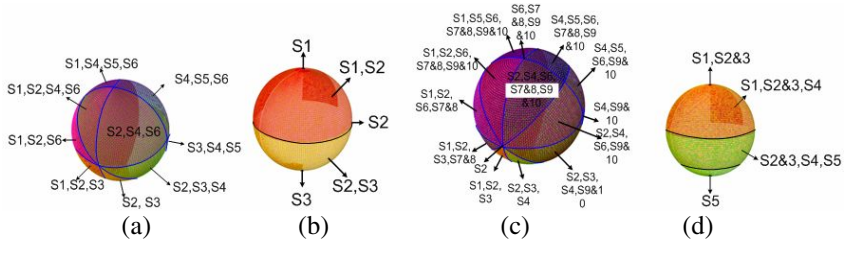


Fig. 4. The orientation sphere segmented by the measurable regions containing different sets of surfaces

the measurable regions segmented on the orientation spheres that are obtained in Fig. 3. Now the object is represented by the silhouettes of the object acquired from the viewing directions which are the centers of the measurable regions.

5 Object Representation and Recognition Using Silhouettes

Silhouette is the boundary contour of a 2D image with respect to the background which is projected on the 2D plane whose normal is the viewing direction of a camera. It has a unique shape for every viewing direction on the orientation sphere and thus it can provide the type and pose information of the object. To estimate the type and pose of a 3D object using its silhouette obtained on the 2D image plane of a single camera, here a new representation scheme is illustrated. It generates the extended convex hull if a silhouette is obtained and then represents it using the feature relation graph, as shown in Fig. 5.

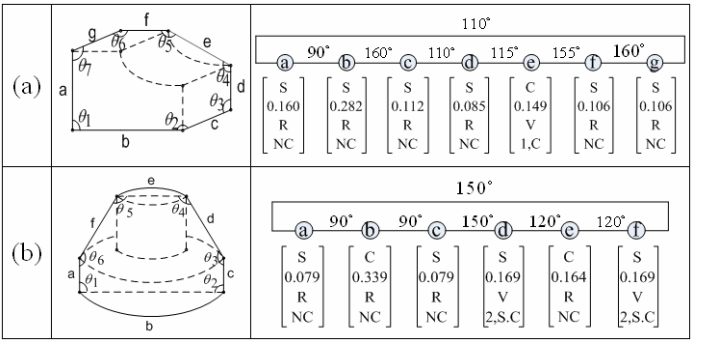


Fig. 5. Graph representation of the extended convex hull of a silhouette

A feature relation graph consists of the nodes illustrating the features of an edge and the arcs illustrating the adjacency as well as the inside angle between the neighboring edges. Four features are used for illustrating a node(an edge): 1) Edge type (Straight/Curved), 2) Edge length over total length of silhouette(L), 3) Convexity(Virtual/Real), and 4) Number and Types of the actual edges covered by the virtual edge(N, S/C..). The 4th feature is included when the third feature is Virtual. An arc is weighted by the inside angle between two nodes(edges). If an edge is curved one, then it is approximated by the straight line connecting two end points of the curved edge. In Fig. 5(a), the edge e is a virtual edge covering one curved edge, and the edge d and f in Fig. 5(b) are the virtual edges covering one straight and one curved edges respectively.

Based on this object representation scheme, the recognition process follows the flow given as a block diagram in Fig. 6. The database, representing an object using a set of silhouettes where each silhouette is described by the feature relation graph of its extended convex hull, is constructed first. Once a test image is obtained, then the silhouette of the object included in the image is derived and its extended convex hull is generated. Using the features of the extended convex hull, [S/C, R/V, NI&T], those model silhouettes having those features in the database are searched. The model silhouettes, partially matching with the test one, are also selected when the objects are allowed to be overlapped. The other features (L, θ) not used for selecting the

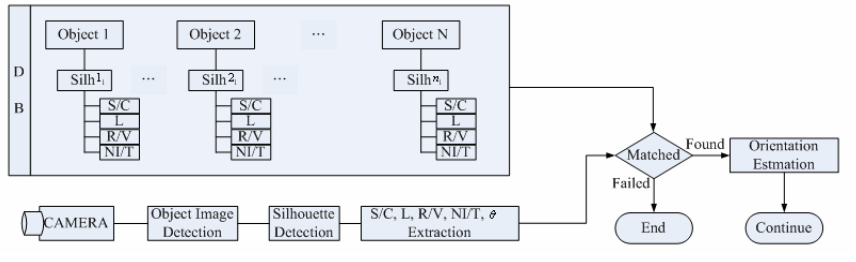


Fig. 6. Recognition procedure

silhouette model are used for determining the object's pose. Since each silhouette model has its representative viewing direction(α , β), the viewing direction of the test silhouette can be determined by adjusting (α , β) with (L, θ).

6 Experiments

The proposed algorithm has been implemented on Pentium IV PC with Visual C^{++} to evaluate the performance in terms of the accuracy of object recognition and pose estimation. The experiment has been performed following the flow explained in Fig. 6.

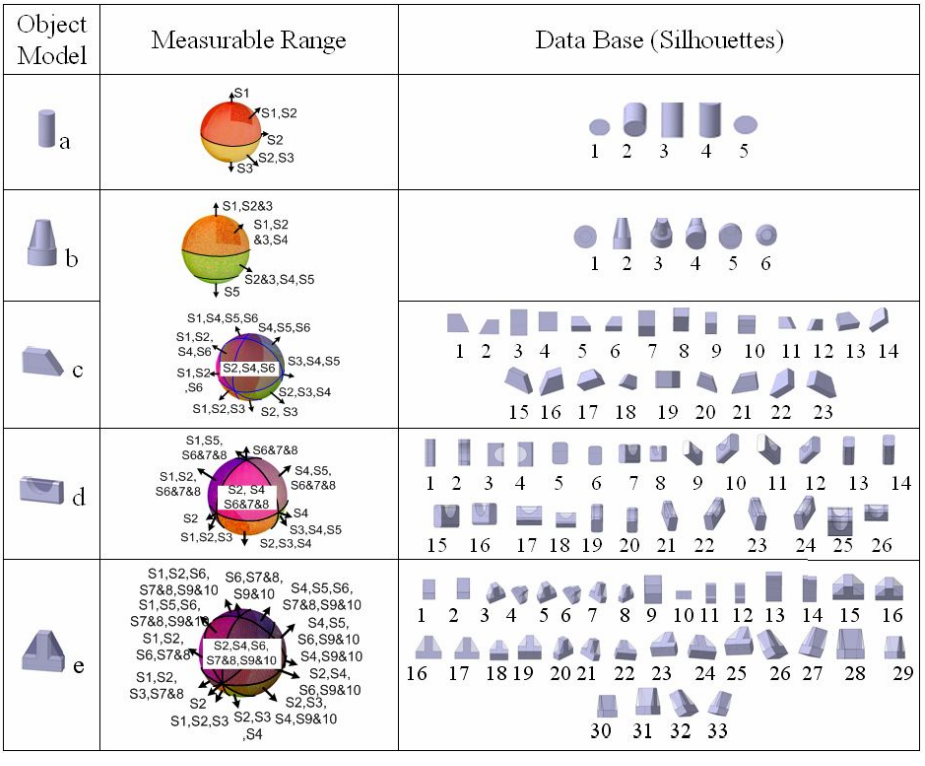


Fig. 7. Five object models used for the experiment and their silhouettes whose features are included in the database for recognition

In this experiment, five object models including convex and concave shapes are selected. Their size is about 10x10x10cm^3. To construct the database with these objects, all possible measurement regions are determined and the silhouettes of individual measurement regions are derived to generate the feature relation graphs. The silhouettes of five object models are illustrated in Fig. 7.

The performance of the algorithm has been tested with 30 images. In 10 of them, the objects are positioned without contact to each other. In 20 of them, the object

models are arbitrarily overlapped. In Fig. 8, five test images are given for the purpose of illustration. For each input image, the same procedure used to construct the database is applied to extract the feature relation graph for each object in the input image. For this purpose, the silhouette of each object image is extracted and then its extended convex hull is formed. With this feature relation graph of a test object, the database is searched to find the best matching silhouette and the object model having this matched silhouette is determined as the model of the test object.

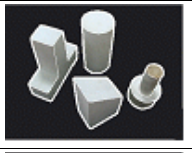
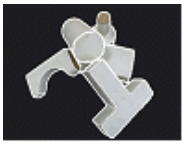
Input Image		Extracted Silhouettes and Matching Results				
IN5		 1	 2	 3	 4	 5
		e-1 e-2	c-15	b-4	a-2 a-4	d-8 d-16
IN7		 1	 2	 3	 4	
		e-8 e-22	a-2 a-4	c-18	b-3	
IN15		 1	 2	 3		
		a-2 a-4	e-15 e-16	c-15		
IN18		 1	 2	 3		
		c-17	e-27 e-33	a-2 a-4		
IN25		 1	 2	 3	 4	 5
		d-22 d-24	a-2 a-4	b-3	NC	e-8 e-21

Fig. 8. Test Images composed of 5 object models

Fig. 8 includes the silhouettes of the objects included in the input image and also shows the results (the index of the silhouette models) of searching the database with these silhouettes. The proposed algorithm has found the matching object models successfully when the objects in an input image are separated with no contact (IN5, IN7). When an object is symmetric, then multiple matches are found. When the objects in an input image are overlapped, only those objects on the top whose silhouettes are completely appeared successfully found their matching silhouette models. For the partially occluded objects (IN15-2, IN18-2, IN18-3, IN25-3, IN25-4), the matching

models are selected if their graphs match with that of the test silhouette more than 60%.

7 Conclusion

To efficiently recognize a 3D object and estimate its orientation using a single camera, this paper presented a new representation scheme that describes a 3D object using the set of unique silhouettes, each of which is represented by the relation graph of the silhouette's extended convex hull. By representing an object using the minimal set of silhouettes that the object can generate and by describing each silhouette with a small number of features using the relation graph, the DB size can be reduced and thus the time complexity spent for the matching process is also significantly reduced. Since the silhouettes of an object model are acquired on a specific viewing orientation, once the matching object model is selected its pose can also be estimated. Although the proposed algorithm has shown outstanding performance even when the objects are overlapped, it still needs to enhance the edge detection scheme to expect a better performance in the case where the objects are overlapped.

Acknowledgement

This work has been supported partially by Soongsil University and Center for Intelligent Robot.

References

1. Philip Quick and David Capson, Analysis of Determining Camera Position Via Karhunen-Loeve Transform. IEEE Southwest Symposium on Image Analysis and Interpretation, 88-92, 2000.
2. P, J. Besl and R. C. Jain, Three dimensional object recognition. Comput. Surveys, vol. 17, No.1 77-145, 1985
3. S. Joshi and T. C. Chang. Graph-based heuristics for recognition of machined features from a 3D solid model. Computer-Aided Design, Vol. 20, No.20 58-66, 1988.
4. DeFloriani L. Feature Extraction from Boundary Models of Three-Dimensional Objects. IEEE Trans. Pattern Matching and Machine Intelligence, Vol. 11, No 8, 785-598.
5. Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. IEEE Trans. Pattern Analysis and Machine Intelligence, 18(4): 377-388, April 1996.
6. R. C. Wilson, E.R. Hancock. Structural matching by discrete relaxation, IEEE Trans. Pattern Analysis and Machine Intelligence, 19(6): 634-647, June 1997.
7. B. Luo, Richard C. Wilson, E. Hancock. Spectral embedding of graphs. Pattern Recognition, 36: 2213-2223, October 2003.
8. Terry Caelli and Serhiy Kosiov. An eigenspace projection clustering method for inexact graph matching. IEEE Trans. Pattern Analysis and Machine Intelligence, 26(4): 515-519, April 2004.
9. FANUC LTD.
http://www.fanuc.co.jp/en/product/new_product/2003/0311/newbinpicking.html.

Image Retrieval by Local Contrast Patterns and Color

M.K. Bashar and N. Ohnishi

Nagoya University, Graduate School of Information Science,
Furo-cho, Chikusa-ku, Nagoya, 464-8603, Aichi, Japan
{khayrul,ohnishi}@ohnishi.m.is.nagoya-u.ac.jp
http://www.agusa.i.is.nagoya-u.ac.jp/person/khayrul/

Abstract. Despite simplicity of the Local binary patterns (LBP) or local edge patterns (LEP) for texture description, they do not always convey complex pattern information. Moreover they are susceptive to various image distortions. Hence we propose a new descriptor called Local Contrast Patterns(LCP), which encode the joint difference distribution of filter responses that can be effectively computed by the higher order directional Gaussian derivatives. Though statistical moments of the filter responses are typical texture features, various complex patterns (e.g., edges, points, blobs) are well captured by the proposed LCP. Observation shows that anyone of the first few derivatives can produce promising results compared to LBP(or LEP). To extract more improved outcome, two sub-optimal descriptors (LCP1, LCP2) are computed by maximizing local bit frequency and local contrast-ratio. Global RGB color histogram is then combined with the proposed LCP descriptors for color-texture retrieval. Experiments with the grayscale (Brodatz album) and color-texture (MIT VisTex) databases show that our proposed LCP (LCP+RGB) produces 8 % and 2.1 % (1.4 % and 1.9 %) improved recall rates compared to LBP and LEP (LBP+RGB and LEP+RGB) features. The achievement of the lowest rank ratio, i.e., 2.789 for gray images (1.482 for color images) also indicates the potentiality of the proposed LCP2(LCP2+RGB) feature.

Keywords: Retrieval, local contrast patterns, texture , color histogram.

1 Introduction

With the rapid proliferation of the internet and the world-wide-web, the amount of digital image data accessible to users has grown enormously. Image databases are becoming larger and more widespread, and there is a growing need for effective and efficient image retrieval system.

While choosing feature descriptors for image retrieval we have several choices to begin with. The most common categories of descriptors are based on, color, texture, and shape, and there are many alternatives in each of these. Some popular descriptors are color histograms [12], color correlograms [4], MPEG color and texture descriptors [7], Gabor and wavelet features [8], [2], [13].

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 136–145, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Algorithms based on statistical feature distributions [10], [15], [14], [9], [3] have also proved to be efficient in texture analysis and retrieval. Among them, LBP method has recently drawn a lot of attention because of its efficient characterization of micro textures, e.g.,edges, points, constant areas etc. However its application to raw images is susceptive to noise, illumination, viewpoint and other image distortions like blurring.

According to Taylor series expansion, a better local pattern can be described by higher order derivatives, which are obtained by unique natural kernel of Gaussian derivatives. We thus propose a new pattern descriptor called Local Contrast Patterns (LCP) that uses the directional higher order image derivatives and encode the pattern contrast with the following advantages.

1. LCP is simple but efficient operator for local image description.
2. It encodes the contrast of patterns, represented by various image derivatives.
3. Robustness to various image distortions (i.e., noise, blurring, illumination, and view point changes etc.)

The remaining of this paper is organized as follows: section 2 gives the details of our proposed approach including feature extraction and similarity measures. Section 3 explains experimental details including discussion, while the work is concluded through section 4.

2 Color and Gray Texture Retrieval Using Feature Distribution

2.1 Overview of the System

Our proposed retrieval system first transforms the original image into the pattern domain using directional Gaussian derivatives. The various responses obtained by directional filtering can be used to extract LCP value. LCP values are obtained at every pixel using the contrast of response values in a small neighborhood (3×3). LCP histogram is then obtained for every region of interest or image. This pattern distribution is finally used for retrieval using histogram intersection similarity measure.

2.2 Feature Extraction

We proposed LCP features in this study and combined them with the color histogram. RGB space is chosen for simplicity. Color histogram is then obtained by partitioning RGB color space into 64 disjoint regions.

Local Contrast Pattern. In a recent approach [9], T. Ojala et al. proposed a simple LBP approach, which is based on the joint difference distribution of gray levels. However, we observe that the partial directional derivatives of gray level images may have sufficient pattern information. Surprisingly, even a single partial derivative can produce better or comparable results than LBP. We thus

propose to use new binary patterns called LCP, which is obtained by convolving image function with various order directional Gaussian derivatives $G_i^{\theta_j}$, where i, j are order and orientation parameters, respectively. Fortunately, we have a nice scale space theory [6] that uses unique Gaussian kernel, given by

$$G(x, y; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{x^2 + y^2}{2\sigma^2}). \tag{1}$$

However the exact number and type of the Gaussian derivative basis functions is motivated by the requirement of rotation invariance [11], which can be achieved by the minimal basis set: G_1^0,G_1^{90},G_2^0,G_2^{60},G_2^{120},G_3^0, G_3^{45}, G_3^{90}, and G_3^{135}. We therefore represent these nine filters in our current study by

$$G_i^{\theta_j}, i = 1, 2, 3; \theta_j = j\pi/(i+1), j = 0, 1, ..., i. \tag{2}$$

Note that we have omitted using σ for the sake of simplicity. The response of an image patch I centered at (x_0, y_0) to a particular basis filter $G_i^{\theta_j}$ can be obtained by convolving the image patch with the filter:

$$r_{i,j}(x_0, y_0) = \int\int G_i^{\theta_j}(x_0 - x, y_0 - y) I(x, y) \partial x \partial y \tag{3}$$

This response $r_{i,j}(x_0, y_0)$ can be extended to multiscale representation by $r_{i,j,s}(x_0, y_0)$, where $s = s_{min}, ..., s_{max}$.

To compute LCP, the above responses from different derivative masks are adapted in the following equations.

$$LCP_{i,j}(x, y) = \sum_{(k,l) \in I or R} bw(k, l) \times u(r_{i,j}(k + x, l + y) - r_{i,j}(x, y)), \tag{4}$$

$$u(z) = \begin{cases} 1 \ (z \geq 0) \\ 0 \ (z < 0) \end{cases}$$

and

$$bw(k, l) = \begin{pmatrix} 1 & 2 & 4 \\ 8 & 0 & 16 \\ 32 & 64 & 128 \end{pmatrix}$$

Here $r_{i,j}(x, y)$ is a 3×3 block from any response image, and $bw(k, l)$ is the binary weight mask, and $LCP(x, y)$ is the output LCP value at the pixel located at (x, y). Note that the LCP value ranges between 0 and 255. For color-texture image, the proposed LCP feature can be computed from Y-component in YIQ transformation [15] by

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B, \tag{5}$$

where RGB corresponds to color components in the original image. LCP histogram can be obtained using LCP values ranges from 0 to 255 by

$$Ht_i = \frac{n_i}{N}, \tag{6}$$

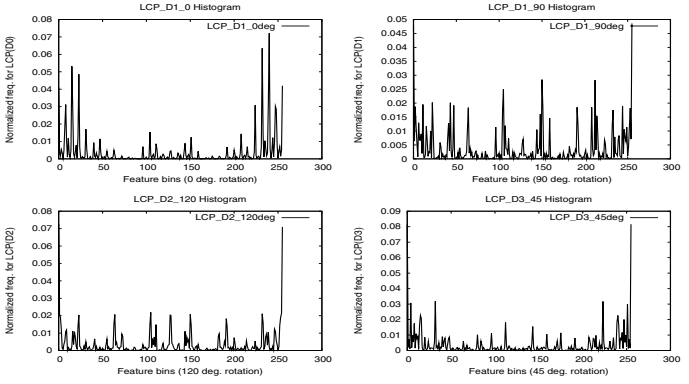


Fig. 1. LCP Histograms. LCP histograms with (a) first derivative with 0^o, (b) first derivative with 90^o, (c) second derivative with 120^o, and (d) third derivative with 45^o

where n_i is the number of pixels with LCP value i and N is the number of total pixels in the image I or region R.

For clarity $LCP_{i,j}$ can be denoted by LCP(D1), LCP(D2), LCP(D3) without explicit orientation parameter. Likewise LCP for zero order derivative and joint mean-variance features are denoted by LCP(D0), MV(D0), MV(D1), MV(D2), and MV(D3). Fig. 1 shows how LCP histograms represent various patterns at 0^o, 90^o, 120^o, and 45^o orientations for D35 image. We also attempted to combine multiple responses by two simple algorithms.

Algorithm 1

1. Compute nine directional responses by using Gaussian derivative filters as mentioned.
2. Select a small 3×3 block from each of the nine responses and compute LCP patterns.
3. For each bit position, count the number of '1' and '0' bits as $sum1$ and $sum0$. If $sum1 > sum0$, assign '1' for that position. Assign '0' otherwise. This will create a new pattern, called LCP1.

Algorithm 2

1. Compute nine directional responses by using Gaussian derivative filters as mentioned.
2. Select a small 3×3 block from each of the nine responses $r(x, y)$ and compute contrast-ratio (CR) for the block, where $CR(x, y) = \frac{SAD(x,y)}{AOS(x,y)}$. Here $SAD(x, y) = \sum_{(k,l)\in R} |r(k + x, l + x) - r(x, y)|$ and $AOS(x, y) = |\sum_{(x,y)\in R} r(x, y)|$ and (x, y) is the block center.
3. Select the response that maximizes nine contrast-ratio, (CR).
4. Compute LCP, which is the LCP2 for that block.

2.3 Similarity Measurement and Performance Evaluation

In our study, we used histogram intersection technique [15] to measure the degree of matching between two texture regions on the basis of the distributions.

If the similarity measures for the color and texture are SMc and SM_t, the overall similarity can be given by

$$SM = w_{color} \times SM_c + w_{contrast} \times SM_t. \tag{7}$$

Currently, we set equal weights, i.e., w_{color} and $w_{contrast}$ are to 1.0.

Performance evaluation in this study is done by using two well-known metrics, namely *rank-ratio* [15] and *precision-recall* [5].

3 Experimental Results

The proposed system is implemented in an IBM compatible PC with a Single Pentium(R)D 2.66 Ghz CPU and 448 MB with the operating system of Windows Xp English version Service Pack 2. The program is developed in the web-based "Struts" framework that uses java servlets including other java classes for business logic with html/jsp in the front end. The experimental results include two parts, one for gray level images, and one for color texture images.

3.1 Retrieval for Gray Level Texture Images

The texture database used in the experiments consists of 40 different textures classes. These are gray-scale images of 640 × 640, collected from the Brodatz texture album [1]. Each of the 40 images is divided into twenty five 128 × 128 non-overlapping sub-images, called related images. Consequently, we obtain a database of 1000(= 25 × 40) grey texture images. Each texture image is then regarded as a whole region from which LCP histograms are stored into the database. Some texture images from the database are shown in Fig. 2.

Fig. 3 shows the retrieval results of the proposed LCP1 feature for a single query. Results for other features are not shown for space problem. Clearly LCP1 feature is able to retrieve almost all 25 related images. Table 1 show comparative performance of the proposed and conventional features. An increase in retrieval performance (8% (= (0.784 − 0.704) × 100) to LBP, 2.1% to LEP, 15.2 % to MV(D0)) is achieved by our LCP(D1) compared to the LBP, LEP, and MV(D0)

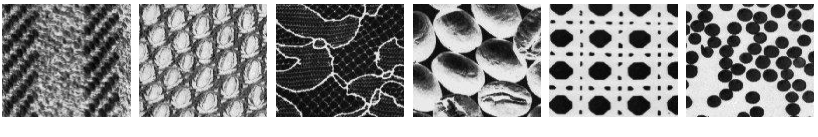


Fig. 2. Sample images from gray level texture database (Brodatz album). These are 128 × 128 images, designated (row-wise) as D11, D35, D41, D47, D101, D67 according to Brodatz notation.

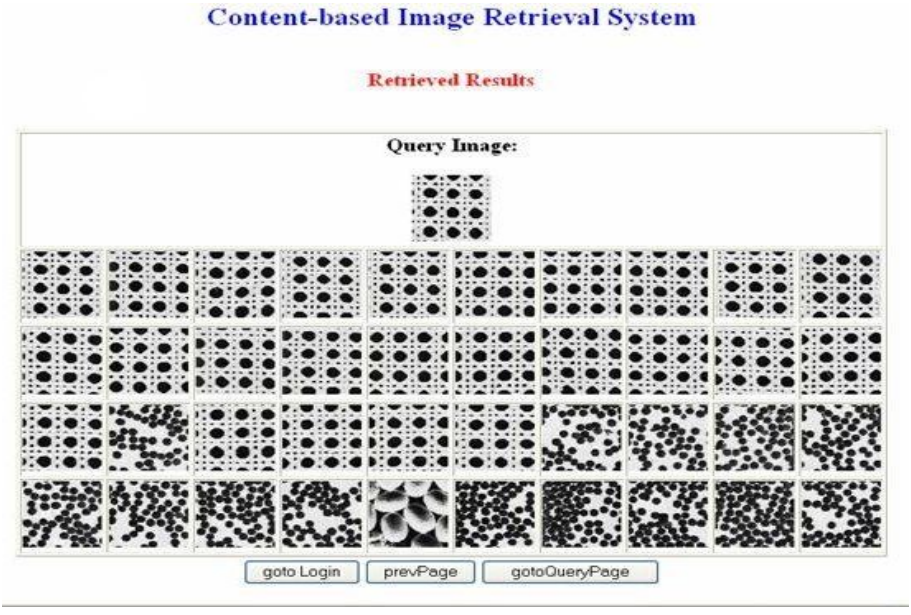
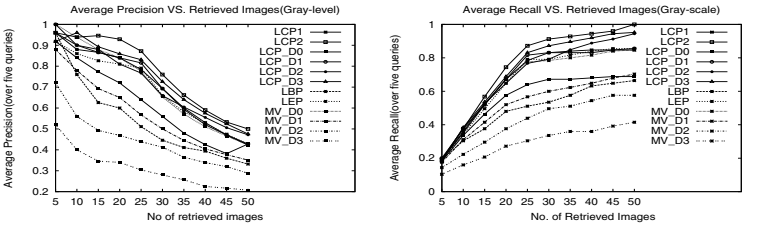


Fig. 3. Retrieved results . Results are due to our proposed LCP1 (Other result are not shown for space limitation). Note this result is obtained by a single Gaussian scale ($\sigma = 0.85$) with (7×7) filter kernel.

Table 1. Comparison of our LCP features with LBP, LEP, MV(D0), MV(D1), MV(D2), MV(D3) in average retrieval performance (40 gray-level query images, one per category)

Features	Retrieval Performance			
	$R^{(10)}$	$R^{(25)}$	$R^{(50)}$	RR
LCP1	0.334	0.749	0.832	3.307
LCP2	0.357	0.769	**0.867**	**2.789**
LCP(D0)	0.341	0.751	0.830	3.886
LCP(D1)	**0.364**	**0.784**	0.860	2.938
LCP(D2)	0.347	0.727	0.845	3.424
LCP(D3)	0.326	0.682	0.820	3.66
LBP	0.341	0.704	0.79	3.528
LEP	0.358	0.763	0.861	2.889
MV(D0)	0.327	0.632	0.778	4.305
MV(D1)	0.257	0.461	0.590	6.916
MV(D2)	0.216	0.381	0.510	10.038
MV(D3)	0.166	0.298	0.426	11.889



(a) (b)

Fig. 4. Performance analysis for gray image retrieval. Average (over 5 query images) plots for (a) $P^{(T)}$, (b) $R^{(T)}$, at 5 interval, i.e., $T = 5, 10, ..., 50$. Note this results is obtained by a single scale ($\sigma = 0.85$) with (7×7) filter kernel.

features. Though sub-optimal LCP1, LCP2 features are inferior to LCP(D1), they also show better performance (i.e., 6.5%, 4.5 % to LBP) compared to LBP with the lowest (2.789) rank-ratio. Since the mean-variance features (MV(D0), MV(D1), MV(D2), and MV(D3)) appear very weak, we relaxed their further adoption in our study.

In another experiment (detail omitted),we observe that the proposed LCPs works well with 29 out of 40 texture categories in terms of average precision, recall, and rank ratio metrics. The rest 11 classes (D27, D30, D48, D66, D111, D42, D51, D74, D88, D99, D107), for which LEP, LBP, and MV(D0) works well, contains retatively larger texture primitives. Using larger σ with LCPs can solve this problem. However this new setting may not work well with the mention 29 classes. Hence determining optimal size for the texture primitive is necessary. We must also maintain a suitable relationship between the chosen σ and kernel size of derivative filters.

3.2 Color Texture Retrieval for Texture Images

For color-texture retrieval, we used VisTex database from MIT Media Lab. We used 32 color images of natural textures, each of which is 512×512. Each of the 512×512 images is divided into sixteen 128×128 non-overlapping sub-images. Consequently, we obtain a database of $512 (= 32 \times 16)$ color-texture images. Fig. 5 shows some sample images. During similarity computation, we combined the LCP and color histograms by Eq. 7 in section 2.3.

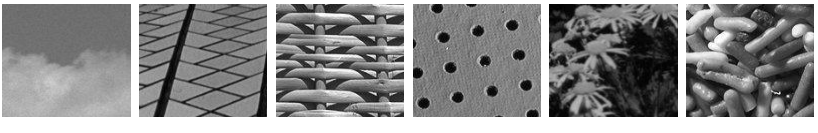


Fig. 5. Sample images from color-texture database (MIT VisTex). These are all 128×128 images.

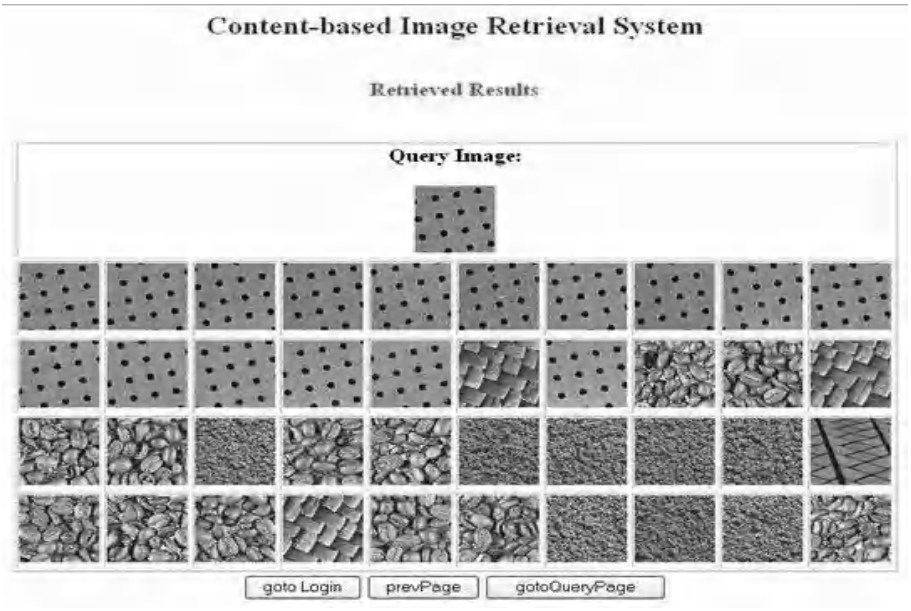
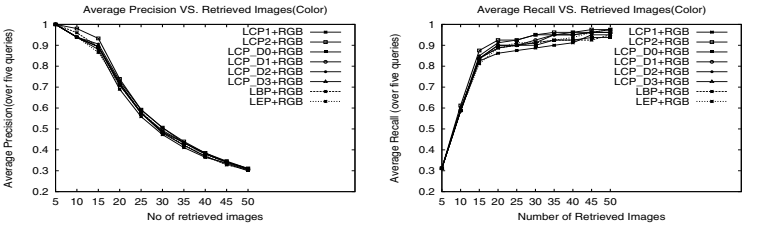


Fig. 6. Retrieved results . Results are due to our proposed LCP2+RGB (Other results are not shown for space limitation). Note this results is obtained by a single Gaussian scale ($\sigma = 0.5$) with (7×7) filter kernel. RGB indicates RGB global color histogram.

Table 2. Comparison of our combined LCP+RGB features with LBP+RGB, LEP+RGB in average retrieval performance (32 color query images, one per category)

Features	Retrieval Performance			
	$R^{(10)}$	$R^{(25)}$	$R^{(50)}$	RR
LCP1+RGB	0.603	0.914	0.960	1.610
LCP2+RGB	**0.607**	0.927	0.962	**1.482**
LCP(D0)+RGB	0.605	0.931	**0.968**	1.454
LCP(D1)+RGB	0.605	**0.935**	0.962	1.489
LCP(D2)+RGB	0.603	0.919	0.960	1.521
LCP(D3)+RGB	0.605	0.933	0.966	1.490
LBP+RGB	0.601	0.921	0.953	1.605
LEP+RGB	0.605	0.916	0.960	1.766

Fig. 6 shows retrieval results for one query image. It is obvious that almost all 16 related images were retrieved by our LCP2 feature. Table 2 shows average retrieval performance for 32 queries. All of the LCP features produce more than 90% recall rates within the first 25 retrieved images. However we found here a very minor improvement (1.4% to LBP, 1.9% to LEP) compared to LBP+RGB and LEP+RGB features. We also obtained the minimum rank ratio for our



(a) (b)

Fig. 7. Performance analysis for color-texture retrieval. Average (over 5 query images) plots for (a) $P^{(T)}$, (b) $R^{(T)}$, at 5 interval, i.e., $T = 5, 10, ..., 50$. Note this results is obtained by a single scale ($\sigma = 0.5$) with (7×7) filter kernel.

LCP2 feature compared to LBP, LEP features with RGB combination. Note that in both experiments, LCP(D1), LCP(D2), and LCP(D3) are computed for 0^o orientation.

3.3 Discussion

We combined various derivative responses by maximizing local contrast-ratio and bit frequency. Despite better results, we should explore more efficient combining strategy for optimal description. Currently, texture and color features are combined with equal weighting values. An optimal weighting scheme may provide better results.

4 Conclusion

We have proposed a new texture extraction approach called local contrast patterns(LCPs) and combined it with the global RGB histogram for color-texture retrieval. The method represents the contrast of the various image derivative patterns. Two simple algorithms that combines multiple responses by local contrast-ratio and binary bit frequency maximization are also proposed. Promising outcome is obtained by LCPs in retrieving gray-scale and color texture images. However, we have yet to explore rotation and scale invariant analysis including the robustness to noises and other image distortions. Finally, we will pay our attention for object based retrieval in our future study.

Acknowledgements

This work was supported by the Science Research Foundation and the Center of Excellence(COE), Nagoya University, Japan. We would like to thank my laboratory friends for their miscellaneous cooperations.

References

1. Brodatz P.: Textures: A Photographic Album for Artists and Designers. Dover, New York, (1966).
2. Bashar M. K., Ohnishi N., Matsumoto T., Takeuchi Y., Kudo H.: Image Retrieval by Pattern Categorization Using Wavelet Domain Perceptual Features with LVQ Neural Network. Pattern Recognition Letters (PRL) **26(15)** (2005) 2315-2335
3. Bashar M. K., Matsumoto T., Ohnishi N.: Wavelet Transform-based Locally Orderless Images for Texture Segmentation. Pattern Recognition Letters **24(15)** (2003) 2633–2650
4. Huang J., Kumar S. R., Mitra M., Zhu W.-J., Zabih R.: Image Indexing Using Color Correlograms. In: IEEE Conference on Computer Vision and Pattern Recognition (1997) 762-768
5. Jones K. S.: Information Retrieval Experiment. Butterworth and Co. (1981)
6. Lindeberg T. : Discrete derivative approximations with Scale-Space Properties: A Basis for Low-Level Feature Extraction. Journal of Mathematical Imaging and Vision **3** (1993) 349-376
7. Manjunath B. S., Ohm J.-R., Vasudevan V., Yamada A.: Color and Texture Descriptors. IEEE Transactions on Circuit and Systems for Video Technology **11** (2001) 703-715
8. Manjunath B. S., Ma W. Y.: Texture Features for Browsing and Retrieval of Image Data. IEEE transactions on Pattern Analysis and Machine Intelligence **18** (1996) 837-842
9. Ojala T., Pietikainen M., Maenpaa T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns **24(7)** (2002) 971-987
10. Park S. J., Park D. K., Won C. S.: Core Experiments on MPEG-7 Histogram Descriptors ISO/IEC JTC1/SC29/WG11-MPEG2000/M5984 (2000)
11. Rajesh P.N. Rao and Dana H. Ballard: Natural Basis Functions and Topographic Memory for Face Recognition. Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI), Montral, Qubec, Canada, August 20-25, (1995) 10-17
12. Swain M., Ballard D.: Color Indexing. In: Third International Conference on Computer Vision (1990) 11-32
13. Suematsu N., Ishida Y., Hayashi A., and Kanbara T.: Region-based Image Retrieval using Wavelet Transform. In: 15th Int. Conf. on Vision Interface, Calgary, Canada, May 27-29, (2002) 9-16
14. Takala V., Ahonen T., Pietikainen M.: Block-Based Methods for Image Retrieval Using Local Binary Patterns. **LNCS 3540** (2005) 882–891.
15. Yao C.-H., Chen S.-Y.: Retrieval of translated, rotated and scaled Color Textures. Pattern Recognition **36** (2003) 913–929

A VR Game Platform Built Upon Wireless Sensor Network

Doo-seop Eom[1], Jungshik Jang[2], Taeyoung Kim[1], and JungHyun Han[2,⋆]

[1] Department of Electronics and Computer Engineering
Korea University, Seoul, Korea
[2] Game Research Center, College of Information and Communications
Korea University, Seoul, Korea

Abstract. This paper presents a VR game platform built upon wireless sensor network (WSN). The WSN is composed of multiple beacons and a receiver. Each beacon transmits both an ultrasonic pulse and the RF message at the same time. Then, the receiver attached to a moving device uses the standard time difference of arrival technique to estimate its distance from each beacon. The estimated distances are passed to the context-aware game server, which computes the location of the receiver. The location data are fed into VR games, and determine the virtual device's motion. The prototype implementation proves that WSN and VR games can be neatly integrated, and can lead to a new type of natural interface for the next-generation game.

1 Introduction

Mobility and context-awareness play key roles in ubiquitous computing environments. Specifically, we have seen rapidly growing interest in location-aware applications and systems such as Active Badge [1], Active Bat [2], RADAR [3], and Cricket [4,5]. They provide location information in indoor environments, where Global Positioning System (GPS) shows poor performances.

Active Badge and Active Bat proposed passive ceiling-mounted receivers that obtain information from active transmitters carried by users. Active Badge uses infrared while Active Bat uses both radio frequency (RF) and ultrasound. In contrast, RADAR uses 802.11 RF, and is not as accurate as the systems based on RF and ultrasound. However, RADAR does not require any infrastructure other than 802.11 access points. The Cricket architecture can be taken as an inverse of the Active Badge and Active Bat systems in that ceiling- or wall-mounted active beacons send RF and ultrasound to passive receivers.

We have designed and developed a location-aware system based on wireless sensor network (WSN), and built VR game prototypes upon it. In the VR game prototypes, the physical and virtual worlds are mixed and integrated, and the game player uses a simple and natural interface rather than the traditional game interfaces such as keyboard, mouse and joystick. This paper presents the WSN-based VR game platform.

⋆ Corresponding author.

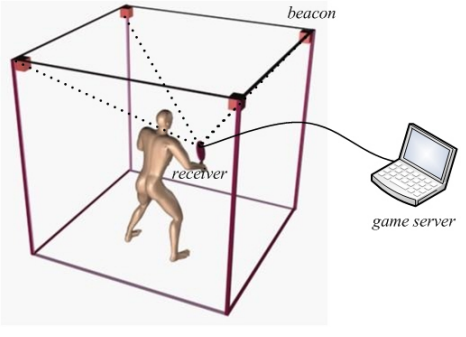
G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 146–155, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The organization of this paper is as follows. Section 2 presents the WSN configuration, its sensor nodes, and the trilateration method for location-aware computing. Section 3 presents the prototypes and interfaces of VR games based on the WSN. Section 4 presents the experiment results and performance analysis, and finally Section 5 concludes the paper.

2 Wireless Sensor Network and Location Computation

2.1 Workspace Configuration

We have built a WSN for the workspace shown in Fig. 1-(a), where each beacon sends both an ultrasonic pulse and the RF message at the same time. The receiver uses the standard *time difference of arrival* technique by observing the time lag between the arrival of the RF and ultrasonic signals, and estimates its distance from each beacon. The estimated distances are passed to the *context-aware game server*, which computes the location of the receiver using the distances, as will be described in Section 2.3.



(a) ***workspace configuration***



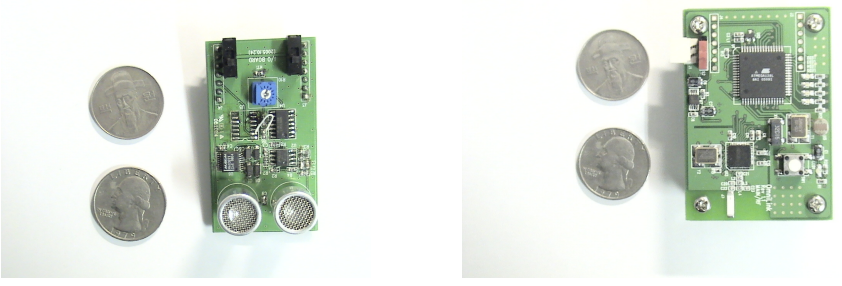
(b) ***beacons attached to ceiling***

Fig. 1. Workspace

Our system is different from Cricket [4,5] in that the receiver is separated from the host. The receiver can then send context information to an authenticated server looking for a specific service. In the current proof-of-concept implementation, game service is requested by providing the host (game server) with the user's location information.

2.2 Wireless Sensor Node

We have developed an ultrasonic sensor module with a pair of transmitter and receiver, shown in Fig. 2-(a). The ultrasonic sensor module is plugged into the main node, shown in Fig. 2-(b). The main node consists of the 8-bit AVR MCU, a mobile transceiver of 2.4GHz ISM bandwidth, 128KB memory, etc. Table 1 shows the detailed specifications of the main node.



(a) ***ultrasonic sensor module*** **(b)** ***main node***

Fig. 2. Ultrasonic sensor module and main node

Table 1. Specifications of the wireless sensor main node

Component	Performance	Remark
CPU	ATMega128 8bit	Micro Controller
Speed	8MHz	
A/D	10bit	>4ch
Processor Current Draw	10mA	Active
	<15uA	Power Down Mode
Flash Memory	128Kbyte	
Radio Freq.	2.4GHz	ISM Band
Data Rate	<256Kbps	
Power	3.3V	
Radio Current Draw	<17.4mA	Tx.
	<19.7mA	Rx.
	<3uA	Stanby
Radio Range	<100m	Open Space
Power Source	2 AA(1.5V)	
Other I/O	8bit	Digital Input/Output

The wireless sensor nodes are used both for the beacon nodes and for the receiver node. The light-weight beacon nodes are easy to deploy. They can be placed with few constraints in an indoor environment such as rooms and corridors. Little physical infrastructure is required. The light-weight receiver node can be easily attached to a physical object. In the current experiments, the receiver node communicates with the game server through serial communication, but can be seamlessly replaced by RF communication.

2.3 Trilateration

The context-aware game server computes the location (x,y,z) of the receiver using the distances through the *trilateration* method, as illustrated in Fig. 3. Suppose

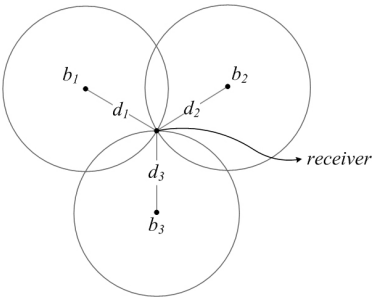


Fig. 3. Trilateration

the location of the i-th beacon is denoted by b_i, and the estimated distance between the i-th beacon and the receiver is d_i. Then, we have the following equation for a beacon:

$$(x - b_{ix})^2 + (y - b_{iy})^2 + (z - b_{iz})^2 = d_i^2 \tag{1}$$

The system of equations is solved using Newton-Raphson method. In the current implementation, the initial location (x_0,y_0,z_0) of the receiver is set to the center of the workspace.

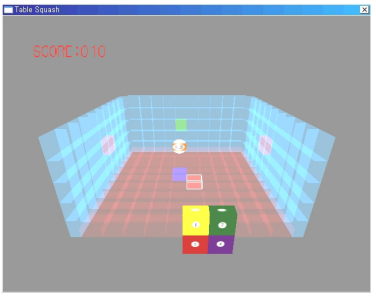
3 VR Game Prototype and Interface

The wireless sensor network can provide a new type of natural interface for many applications, which require tracking the position of a user or moving device. Obvious applications include virtual reality (VR) or augmented reality games.

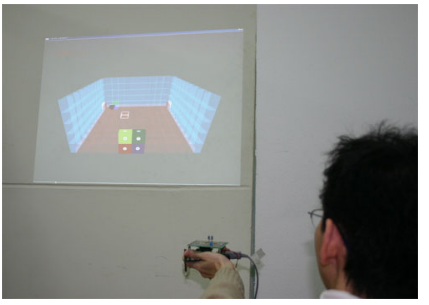
As proof-of-concept implementations, we have developed two VR games: 'table squash' and 'billiard.' Fig. 4-(a) shows the virtual space of 'table squash,' which is bounded by front, left and right planes. The ball and block-shaped racquet are also illustrated. In contrast, the physical space of the real world is open, as shown in Fig. 4-(b). The receiver node of the sensor network is attached to the racquet, which is the only physical device for 'table squash.' The motion of the physical racquet is tracked, and determines the virtual racquet's motion. Fig. 4-(c) and -(d) show the virtual and physical space of 'billiard.'

Fig. 5 describes the modularized frameworks for both location awareness and game service. In addition to the modules for computing and transferring location data (LCM and LTM), the Location-aware Framework includes Virtual Location Module (VLM), which makes the game development process independent of the sensor network. Even with the sensor network disabled, location-aware game can be tested by running VLM, which takes mouse input in the current implementation.

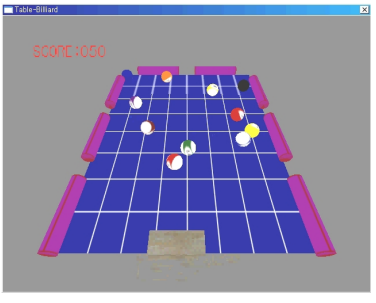
The two frameworks communicate with each other through TCP/IP. Even though they are located in a game server in the current implementation, we



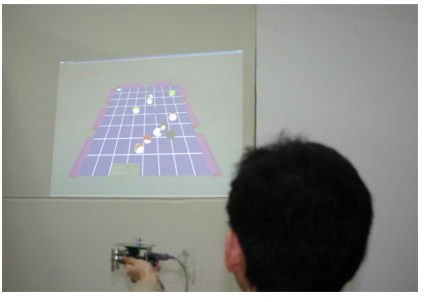
(a) *table squash: virtual space*



(b) *table squash: physical space*



(c) *billiard: virtual space*



(d) *billiard: physical space*

Fig. 4. Virtual and physical spaces for VR game prototypes

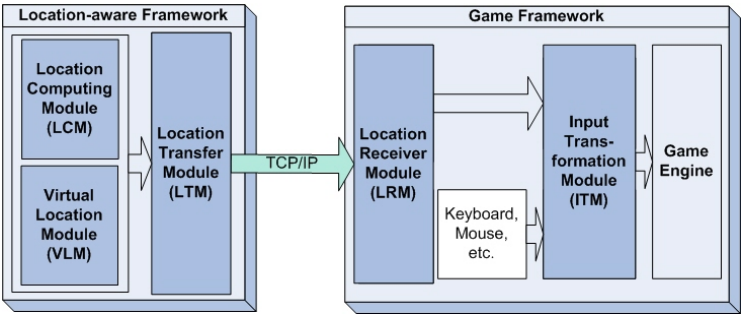
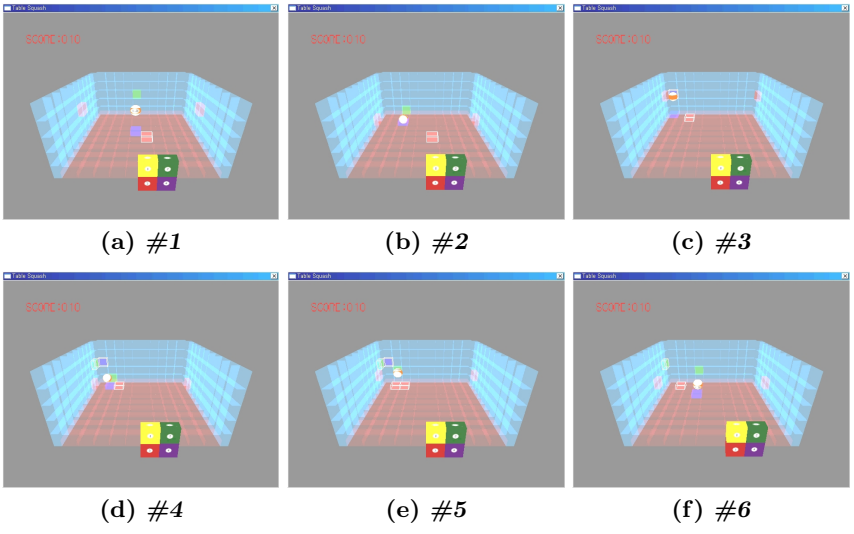


Fig. 5. Location-aware and game frameworks

envision that they can be separated in the future especially for a networked or online game.

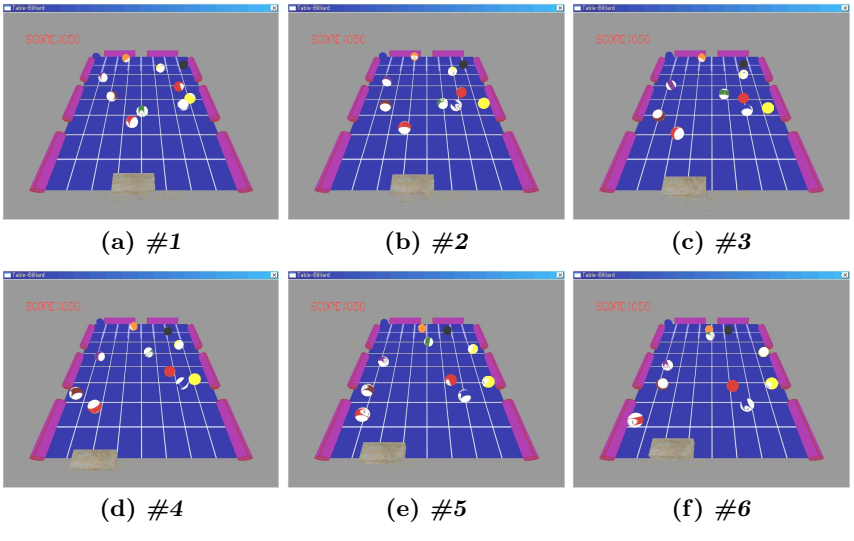
The location data are passed to Location Receiver Module of the Game Framework, and then to Input Transformation Module (ITM). ITM is responsible for processing the location data to be suitable for specific game applications. We are extending 'table squash' such that not only the location but also the orientation of the physical racquet are considered. To determine the racquet orientation, multiple receiver nodes need to be attached to the physical racquet.



(a) *#1* (b) *#2* (c) *#3*

(d) *#4* (e) *#5* (f) *#6*

Fig. 6. Table squash: snapshots



(a) *#1* (b) *#2* (c) *#3*

(d) *#4* (e) *#5* (f) *#6*

Fig. 7. Billiard: snapshots

Their locations are computed and passed to ITM. Then, ITM finally computes the racquet's orientation by considering the relative locations of the receivers. Many 3D applications can benefit from such pose-awareness.

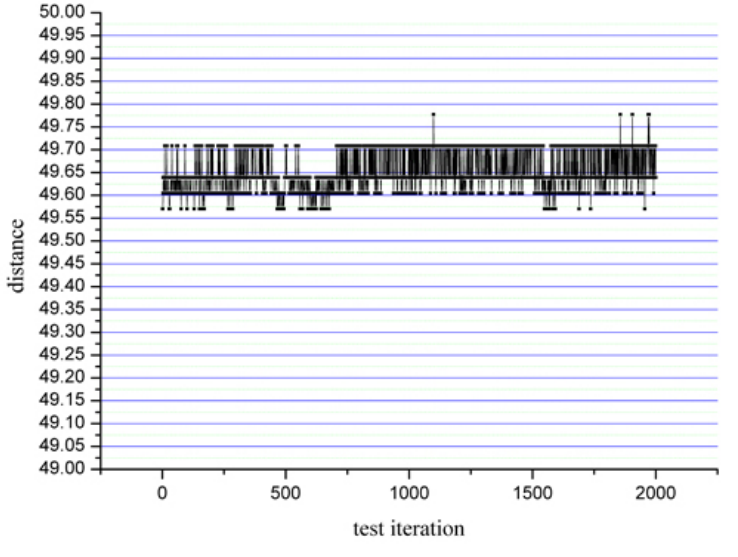
Fig. 6 shows the snapshots of 'table squash.' When the ball hits a wall or floor, the contacting area of the wall/floor is highlighted. DirectX APIs are used for rendering. Fig. 7 shows the snapshots of 'billiard.'

Distance (cm)	Frequency (2,000 times in total)
49.5704	71
49.6048	454
49.6392	1004
49.708	471

(a) ***estimated distances***

$$\begin{aligned} \text{mean } \bar{X} &= 49.64515 \text{ cm} \\ \text{variance } S^2 &= 0.001517 \text{ cm} \\ \text{standard deviation } S &= 0.038945 \text{ cm} \\ \text{standard error } SE &= 0.000871 \text{ cm} \end{aligned}$$

(b) ***statistical analysis***



(c) ***graph***

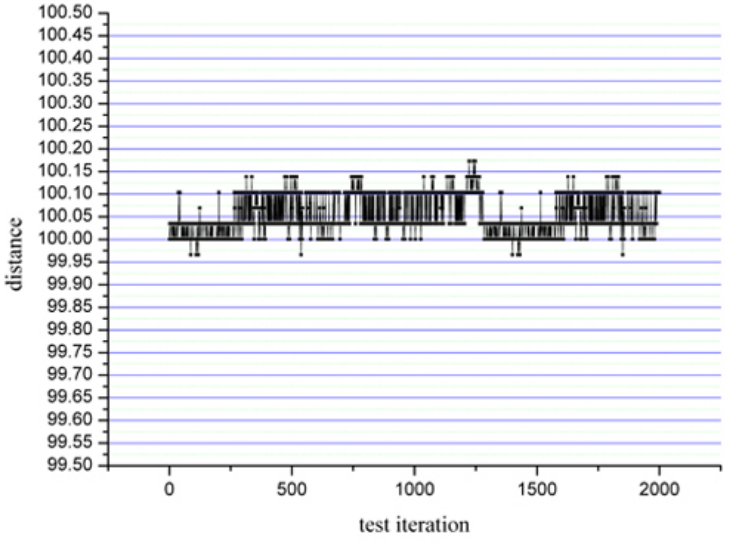
Fig. 8. Test Result #1

Distance (cm)	Frequency (2,000 times in total)
99.9664	10
100.0008	325
100.0352	949
100.0696	65
100.104	571
100.1384	74
99.072	6

(a) ***estimated distances***

$$\begin{aligned} \text{mean } \bar{X} &= 100.0541 \text{ cm} \\ \text{variance } S^2 &= 0.00168062 \text{ cm} \\ \text{standard deviation } S &= 0.04099536 \text{ cm} \\ \text{standard error } SE &= 0.00091668 \text{ cm} \end{aligned}$$

(b) ***statistical analysis***



(c) ***graph***

Fig. 9. Test Result #2

4 Implementation and Performance Analysis

The distances between the beacon nodes and the receiver node are obtained in an *asynchronous mode*. The main node uses a 16-bit timer, and the location of the receiver is computed at 10Hz. Both of the Location-aware and Game Frameworks are coded in C++, and run in 1.4GHz Pentium M CPU and 512MB RAM.

The accuracy of the distance estimation has been tested. For a fixed position of the receiver, the distance from a beacon is measured by hand. Then, the distance is estimated in the proposed framework. Such estimation is done for 2,000 times. Fig. 8 and Fig. 9 show the statistics for the measured distances of $49.6cm$ and $100cm$, respectively. The distance test proves that the wireless sensor network can be successfully integrated with VR games.

5 Conclusion

This paper presents a wireless sensor network and VR game prototypes based on it. The implementation results prove that they can be well integrated, and lead to a new type of natural interface for the next-generation ubiquitous game [6][7].

In the current implementation, the location data are computed at 10Hz. For fully supporting real-time game applications, more efforts should be made to increase the performances. Further, as discussed in Section 3, many applications may benefit not only from location-awareness but also from orientation-awareness. To fulfill such needs, the overall performances should be continuously upgraded.

For both 'table squash' and 'billiard,' single-player versions have been developed as proof-of-concept implementations. However, the proposed Game Framework can be extended to support networked games, for example, a one-to-one squash game. The current system is being extended along the direction.

Acknowledgements

This research is accomplished as the result of the research project for culture contents technology development supported by Korea Culture and Content Agency. This research is also supported by the Ministry of Information and Communication, Korea under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment, IITA-2005(C1090-0501-0019).

References

1. R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. ACM Transactions on Information Systems, 10(1):91-102, January 1992.
2. A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. In Proc. 5th ACM MOBICOM Conf., Seattle, WA, August 1999.

3. P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In Proc. IEEE INFOCOM, Tel-Aviv, Israel, March 2000.
4. N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In Proc. 6th ACM MOBICOM Conf., Boston, MA, August 2000.
5. H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. B. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang. Lessons from Developing and Deploying the Cricket Indoor Location System, November 2003. http://citeseer.ist.psu.edu/balakrishnan03lessons.html
6. Björk, S., Holopainen, J., Ljungstr, P. and Mandryk, R. Personal and Ubiquitous Computing (Special Issue on Ubiquitous Games) Vol. 6, 2002.
7. C. Randell, P. Duff, M. McCarthy, and H. Muller. HeadRacer: A Head Mounted Wearable Computer with Ultrasonic Position Sensing. In: Seventh International Conference on Ubiquitous Computing - Demonstration, September 2005.

Feature Extraction and Selection for Recognizing Humans by Their Gait

Jang-Hee Yoo[1] and Mark S. Nixon[2]

[1] ETRI-Information Security Research Division
161 Gajeong-Dong, Yuseong-Gu, Daejeon 305-700, South Korea
jhy@etri.re.kr
[2] School of Electronics and Computer Science, University of Southampton
Southampton SO17 1BJ, United Kingdom
msn@ecs.soton.ac.uk

Abstract. We describe an efficient and effective feature extraction and selection method for identifying humans by their gait. A sequential set of 2D stick figures is extracted from gait silhouette data by determining the joint angles and body points, and it is used to represent the gait signature that is primitive data for extracting motion parameters. The motion parameters in the gait signatures are stride length, cycle time, speed, and joint angles, and the gait features are extracted from these motion parameters. By measuring a class separability of the extracted features, important features are selected from original feature sets for classifying human in the gait patterns. Then, a *k*-NN classifier is used to analyze the discriminatory ability of the selected features. In experiments, higher gait classification performances, which are 96.7%, have been achieved.

1 Introduction

The ability to recognize humans and their activities by computer vision is a very important task, with many potential applications [1]. Human gait is known to be one of the most universal and complex of all human activities [2][3][4][5]. It has been studied in medical science, psychology, and biomechanics for decades. In addition, each person appears to have his or her own characteristic gait pattern. There is much evidence from psychophysical experiments [6][7] and medical analysis [2][3] that gait patterns have characteristics of uniqueness for each individual. In recent years, with advances in computing hardware, gait has become a potential practical biometric, and techniques have been developed [8]. As a biometric, human gait is defined as a means of identifying individuals by the way they walk. Using gait has many advantages over other biometrics such as fingerprints and face recognition, most notably that it is non-invasive and available at low resolution.

Currently, there are two major approaches to gait recognition in computer vision [8]. The first is model-based where the subject's movement is described by a body model. In this approach, a body model is fitted to the human in every frame of the walking sequence, and kinematic parameters are generally measured on the body model as the model deforms over the walking sequence. An alternative method is to

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 156 – 165, 2006.
© Springer-Verlag Berlin Heidelberg 2006

apply a model-free (or holistic) description to the set of images. Model-free approaches used features based on the motion or shape of subjects. However, a set of human gait data may consist of kinematic, kinetic, electromyography, metabolic and anthropometric variables [9]. Thus, extracting features which truly discriminate among groups is a critical task in identification or recognition of humans, because improperly extracted features frequently lead to low classification rate and require complex classification algorithms.

In this paper, we propose a feature extraction and selection method for recognizing humans by their gait. To describe gait movement, the human body and its contour is extracted from the image sequences. Then, a 2D stick figure is used to represent the human body structure, and it is extracted from body contour by determining the body points. The features are extracted by analyzing the sequence of gait figures during a time period and selected by statistical analysis. Also, a simple *k*-NN classifier is used in attempt to reveal the genuine discriminatory capability of the selected feature sets. In the experiments, the proposed methods have been demonstrated on the largest available database. As such, the proposed method aims to derive features of biometrical significance, in part demonstrated by recognition capability.

2 Human Gait Analysis in Image Sequence

Human walking is the most common means of gait motion, and all normal people follow the same basic bipedal pattern, namely all humans walk in the same basic way [10][11]. Also, gait motion can be decomposed into temporal and spatial components, and these components should be included in gait description. A very important aspect in the fundamental study of gait motion is the analysis of the kinematics of human body segments. Kinematics is the study of movements, or more specifically the geometric description of motion, in terms of displacements, velocities and accelerations [5][11]. Here, a 2D stick figure with six joint angles is used to represent the human body structure for recovering and interpreting the human movement. The gait motion is analyzed by measurement of temporal parameters such as stride, cadence and walking velocity, and kinematics dealing with the analysis of joint movements.

2.1 Extracting Human Gait Signatures

The study of human motion analysis from image sequences is related to several research areas of computer vision such as the motion capture, detection, tracking and segmentation of people. Here, the SOTON indoor gait database [8], which is currently the largest gait database under laboratory conditions, is pre-processed, and the human body and its contour is extracted from the image sequences of the database. An image sequence in the database contains only a single subject walking a specially designed continues track at normal speed. Also, the subjects in the database are filmed in front of a green background and fronto-parallel (where the walking path is normal to the camera view) or at an oblique angle, thus human body extraction from the image sequences is easily achieved through background subtraction. To estimate the position of a human body, the histogram projection profiles are analyzed, and the body region is

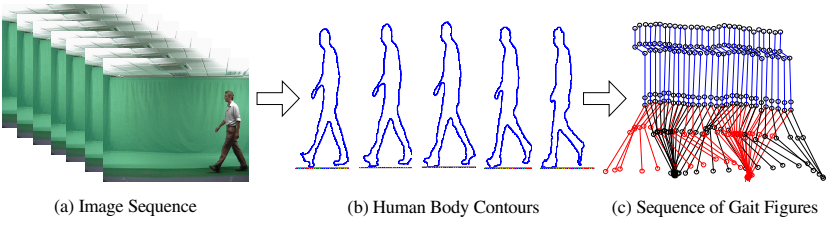
verified by prior knowledge such as size and shape. Then, thresholding method based on similarity measures between the background and the object image is used, and the body contour is extracted by subtraction followed by dilation and erosion [12].

To extract body points in a contour image, a skeleton data with body segment properties is used. For a body height H, an initial estimate of the vertical position of the neck, shoulder, waist, pelvis, knee and ankle was set by study of anatomical data to be $0.870H$, $0.818H$, $0.530H$, $0.480H$, $0.285H$, and $0.039H$, respectively [13]. The gait skeleton can be simply calculated by two border points of each body part p with a range constraint. The angles θ_p of body part p from skeleton data can be approximated by using the slope of the lines in linear regression equation. The body angles around three double support phases [10] are clearly extracted, but the angles around single support phases are not as accurate as those estimated for the double supports. Therefore, a motion tracking method between double supports is used to extract body points at the lower limbs as well. Here, each body point can be also calculated by

$$x_p, y_p = \left[x_i + L_p \cos(\phi + \theta_p) \quad y_i + L_p \sin(\phi - \theta_p) \right] \tag{1}$$

where ϕ is the phase shift, x_i and y_i are the coordinates of a previously established position, and L_p is the length of body segments guided by anatomical knowledge [13].

Also, the horizontal center of mass in the upper body is used as a gait symmetry point to detect the gait cycle. Now we can extract a 2D stick figure with the nine body points from the skeleton data of each body segment. The extracted stick figures from an image sequence are shown in Fig. 1(c). The gait signature can be defined as a sequence of the stick figures obtained from gait silhouette data during one gait cycle. The stick figure model is the most effective and well-defined representation method for kinematic gait analysis. Moreover, the stick figure is closely related to a joint representation, and the motion of the joints provides a key to motion estimation and recognition of the whole figure.



(a) Image Sequence (b) Human Body Contours (c) Sequence of Gait Figures

Fig. 1. Example of Extracted Gait Motion

2.2 Analyzing the Gait Parameters

Human gait contains numerous parameters, and these parameters can be categorized into general and kinematic parameters. The general gait parameters (also known as the temporal and spatial parameters) are the stride length, cycle time (or cadence) and speed and provide a basic description of the gait motion [11]. These parameters

present essential quantitative information about a human gait and give a guide to the walking ability of subject [14]. In addition, the general parameters tend to change together in most locomotor disabilities, so that a subject with a long cycle time will usually have a short stride length and a low speed. Each parameter may be affected by such factors as age, sex, height, muscle strength, etc.

In the previous section, the gait signatures were extracted from the SOTON indoor database. The trajectories of gait signatures contain the general gait parameters on human movement. The period of the gait is determined by number of frames during one gait cycle in image sequence, and frame rate of the SOTON database was 1/25 seconds. The cycle time can be calculated by

$$cycle_time(\text{sec}) = \frac{gait_period(\text{frames})}{frame_rate(\text{frames}/\text{sec})}. \tag{2}$$

The gait speed is the rate of linear forward motion of the body, which can be measured in meters per second. The speed serves as one of the most important factors in determining gait characteristics, and it can be calculated from the cycle time and stride length as

$$speed(\text{m/sec}) = \frac{stride_length(\text{m})}{cycle_time(\text{sec})} \tag{3}$$

where the stride length can be directly estimated from the physical dimensions of the image plane. Namely, the stride length is determined by the coordinates of the forward displacements of the gait signatures during one gait cycle [11][14].

On the other hand, kinematics is the geometric description of motion and does not consider the forces that cause the actions. Kinematic analysis of human gait usually characterizes the joint angles between body segments and their relationships to the events of the gait cycle [2][4][11]. The joint angles of the hip, knee and ankle have been considered as the most important kinematics of the lower limbs. The hip angle is defined by the relationship between the thigh and trunk or pelvis, and the pattern of the rotation is nearly the same as that of thigh rotation. The angle between the thigh and shin is the knee angle, and the ankle angle is the angle between the shin and foot. However, the joint angles of a stick figure in the gait signature can be determined by the coordinates of the body segments. Also, kinematics explicitly defines the state vector of a stick figure at a specific time. The state vector is defined by

$$\Theta_t = (\theta_1, ..., \theta_n). \tag{4}$$

The state vector is the set of joint angles including independent parameters defining the positions and orientations of all joints belonging to the gait figure. In the gait signature, a sequence of the state vectors contains all the geometrical and time-related properties of the gait motion. Therefore, the time series of the joint angles provide much kinematic information such as linear and angular velocities, and accelerations, which are derived from the displacements with time interval. Moreover, they have basically the same characteristic with respect to the gait cycle.

3 Feature Extraction and Selection

The gait signatures have been extracted from image sequences in the SOTON indoor database, and their general and kinematic motion parameters were calculated. Also, several kinds of gait features are extracted from the motion parameters of the sequence of gait signatures. The gait features may contain information that is redundant or superfluous, in which case it is usually required to select a subset to reduce extraneous noise. This process of removing irrelevant and redundant features is known as feature selection. Using feature selection often improves the accuracy and reduces the computational effort involved in classification. Thus, feature selection is an essential step to enhance correct classification in the presence of many irrelevant features and a small number of samples.

3.1 Features Derived from Motion Parameters

We can extract trajectory-based features from the sequence of gait signatures. The trajectories are basically vector valued functions at each frame of a sequence of gait signatures, and they can be parameterized by topological and parametric feature vectors. The topological feature vector consists of geometry data of human gait motion such as height, width, and position. The parametric feature vector such as cycle time, stride length, linear and angular velocities has variable length since not every component is always visible [15]. In addition, the trajectory-based features have less dimensionality than optical flow. The nine coordinates that construct the gait signature are the most important topological features, because other parametric features such as the joint angles are calculated based on these coordinates. Therefore, the nine coordinates of a gait signature can be considered as an original feature space. The gait features are basically based on sequences of these spatial-temporal data, and they are extracted by determining an appropriate subspace of dimensionality m in the original feature space of dimensionality d ($m \leq d$).

However, the general parameters are a basic description of gait motion and can be calculated from each of the gait signatures. Also, the kinematic parameters are obtained from the joint angles, which are interpolated by 4^{th}-order trigonometric polynomials, during one gait cycle. In general, the kinematic parameters are time series data during the gait cycle, thus mean and standard deviation values of the time series can be used as gait features. Table 1 shows the gait features based on motion parameters of the sequence of gait signatures. In addition, the standard deviations (σ) in the

Table 1. Gait Features based on the Motion Parameters

Type	Features
General Parameters	*amplitude, body height*
	cycle time (or cadence), stride length, speed
Kinematic Parameters	$mean(\lvert\theta_{neck}\rvert)$, $mean(\lvert\theta_{back}\rvert)$, $mean(\lvert\theta_{hip}\rvert)$, $mean(\lvert\theta_{knee}\rvert)$
	$mean(\lvert\omega_{hip}\rvert)$, $mean(\lvert\omega_{knee}\rvert)$, $\sigma(\lvert\omega_{hip}\rvert)$, $\sigma(\lvert\omega_{_knee}\rvert)$
	$\sigma(\lvert\theta_{neck}\rvert)$, $\sigma(\lvert\theta_{back}\rvert)$, $\sigma(\lvert\theta_{hip}\rvert)$, $\sigma(\lvert\theta_{knee}\rvert)$, γ_{knee}, μ_x , μ_y

kinematic parameters are calculated by a general statistical method, and the angular velocities ω_i at frame i, given an inter-frame time Δt (1/25 sec), are calculated by

$$\omega_i = (\theta_i - \theta_{i-1})/\Delta t. \tag{5}$$

To determine temporal dependence, we generally need periodic data, a sequence of images of human gait covering a whole gait cycle. As a feature, the correlation γ_{knee} between θ_{knee1} and θ_{knee2} is calculated. Symmetry is one of the characteristics of a walking gait [16], so the correlation between the leading and the following leg is an important factor in gait. Also, the centre coordinates μ_x, μ_y of the hip-knee cyclogram [17], which allow interpretation of the relationships between the two angles, are calculated as feature values. The hip-knee cyclograms represent the movements of nearly the entire body, thus they can be representative of the subject's gait pattern.

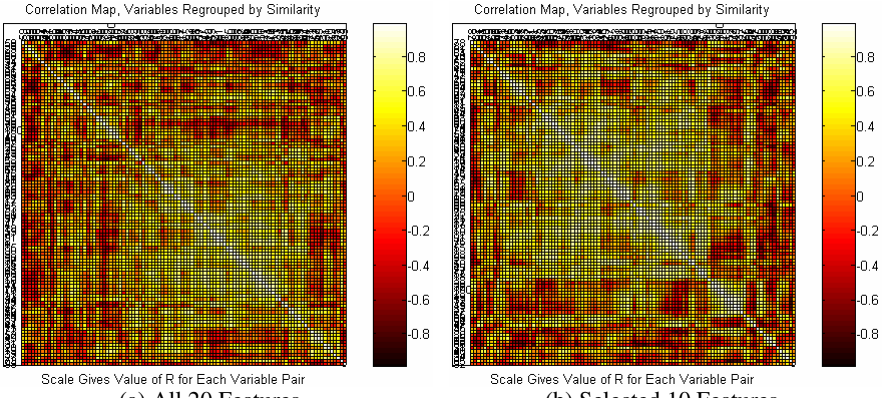
3.2 Feature Selection and Evaluation

The goals for feature selection are to retain as much of the original information as possible, remove as much as possible of the redundant and irrelevant information that could degrade the classification performance, and reduce the measurement data to variables that are more suitable for discrimination [18]. Accordingly, the task of feature selection is to improve the classification rate, so a natural selection rule seems to be to choose the features with lowest individual classification error values. However, feature selection is often relegated to classification so that usefulness of each feature can be evaluated using an evaluation function in order to select the optimal subset. An evaluation function typically tries to measure the discriminatory ability of a feature or a subset to distinguish the different class labels [19]. Here, a statistical distance measure that describes the distribution of subjects or classes in the feature space is employed. That is, inter-class separation due to mean-difference with respect to the class covariances is measured by a variation of the Bhattacharyya distance [20][21][22]. The separation between the two classes i and j, for a given feature, is given by

$$\boldsymbol{S}_{i,j} = \left[\boldsymbol{m}_i - \boldsymbol{m}_j\right]^T \left[\frac{\Sigma_i + \Sigma_j}{2}\right]^{-1} \left[\boldsymbol{m}_i - \boldsymbol{m}_j\right] \tag{6}$$

where $\boldsymbol{m}_i$ is the class mean and $\sum_i$ is the covariance matrix of class i, with equivalent terms for class j. To measure a class separability of the given feature, a mean and standard deviation value of $\boldsymbol{S}_{i,j}$ for each feature is calculated, and the larger values of the ratio of mean to standard deviation ($\hat{S}^2/\sigma$) imply a good class separability.

However, treating features individually as scalars has the advantage of computational simplicity but may not be effective for complex problems and for features with high mutual correlation [22]. Therefore, the discrimination effectiveness of feature vectors is evaluated by a classification error rate for each combination of features, although the best features are selected by the class separability measures. In practice, the relationship between feature vector and classification error rate in a classifier can be analyzed by removing relevant input features. In addition, the classification probability for the feature vector combinations may increase the complexity requirements depending on the classifiers. In order to reduce complexity, a number of efficient



(a) All 20 Features (b) Selected 10 Features

Fig. 2. Correlation Maps for Feature Vectors of 100 Subjects

searching methods have been suggested [19], and the sequential forward selection method [22] is used here with *k*-NN classifier. As a result, 10 features are removed from 20 gait features shown in Table 1, and 10 features such as *amplitude*, *body height*, *cycle time*, *stride length*, *speed*, *mean*($|\theta_{neck}|$), $\sigma(|\theta_{back}|)$, $\sigma(|\theta_{knee}|)$, γ_{knee}, μ_x are selected as gait features.

Correlation and covariance matrices provide the basis for all classical multivariate techniques, because they provide sufficient statistics under multivariate normal linear models [24]. Accordingly, the relationships among the variables may be more readily understood. Fig. 2 shows the correlation maps for feature vectors of 100 subjects. The correlation map of the correlation matrix $\boldsymbol{R}$ is drawn using Corrgrams [24] and re-ordered by magnitudes of correlation in the correlation map. In the figure, each cell is shaded with the intensity of colour scaled 0 to 100, in proportion to the magnitude of the correlation. Thus, a grey cell shows better discriminatory capability, and the diagonal is the brighter cell reflecting the zero distance between the same feature sets. As shown in Fig. 2(b), the inter-class correlation decreased when the selected features are used, hence selected feature vectors can lead to performance improvement of the classification system.

4 Experimental Results

The SOTON indoor database, which equals in size the largest contemporaneous gait database, is used in the experiments. In gait classification, 100 different subjects with seven gait signatures of each subject, a total of 700 gait signatures (≅19,534 images), are used. A total of the 500 feature vectors extracted from the front four of the seven signatures and their means for each of 100 different subjects are used as the training samples. Also, a total of the 100 feature vectors extracted from the means of the remaining three of the seven signatures for each of 100 different subjects are used as the test samples. To classify the gait features, a simple *k*-NN algorithm is employed as a classifier. Undoubtedly, a more sophisticated classifier would be prudent, but the interest here is to examine the genuine discriminatory ability of the features.

In the *k*-NN with values of k = 1, 3, and 5, the results of CCR (Correct Classification Rate) for using different kinds of the feature vectors are shown in Table 2. When not considering the number of features, the features based on kinematic parameters show the best performance in discriminatory capability. By combining two feature vectors, the performance improves to nearly 78% CCR. Also, the best performance comes from only 10 selected features, and the result shows an 84% CCR which is 6 percentage points higher then the result by 20 features as shown in the table. As can be seen in the table, a CCR offer declines only slightly when the value of k increases, so the selected feature vector shows a very good performance. However, all general parameters (100%) and five of the fifteen kinematic parameters (33%) are selected as important gait features.

Table 2. Classification Rate by Selected Feature Vectors

Feature Vector (Number of Features)	Correct Classification Rate (CCR)		
	k=1	*k=3*	*k=5*
General Parameters (5)	44.0%	39.0%	45.0%
Kinematic Parameters (15)	53.0%	47.0%	50.0%
All Parameters (20)	78.0%	74.0%	78.0%
Selected Features (10)	84.0%	80.0%	84.0%

This result confirms that individuality by gait concerns a variety of components. Intuitively, classification by gait concerns the human body shape and its motion. That is confirmed here by selections of the parameters *amplitude*, *body height*, μ_x that describe shape and *cycle time*, *stride length*, *speed*, *mean*($|\theta_{neck}|$), $\sigma(|\theta_{back}|)$, $\sigma(|\theta_{knee}|)$, γ_{knee}, which describe motion factors. Other studies [8] have clearly confirmed classification capability on this database, using more features by area masks, symmetry operator, velocity moments, and Principal Component Analysis (PCA) than given here to achieve a similar classification rate (97.3% CCR for 28 subjects) to our approach gives. However, their approaches are driven purely by biometric capability whereas here we sought to derive confidence in our biomechanical measure indirectly by classification assessment. Since we derive classification capability by these biomechanical measures this gives further confidence in the validity of these marker-less derived measures.

On the other hand, the quality levels of the pre-processed image sequences, which are body contour data of the SOTON indoor database, were graded as *A* (good), *B* (fair), and *C* (bad) to be 30%, 40%, and 30% of subjects, respectively. To analyze the relationship between the quality level and a CCR, 30 subjects are selected from each of the quality levels and classified separately by using the *k*-NN algorithm. Table 3 shows the classification results for each of the quality levels. In the experiments, seven image sequences for each subject with the 10 selected features are also used. As can be seen in the table, a CCR is considerably affected by the quality level of the pre-processed image sequences, namely the better performance of a CCR comes from the better quality level of pre-processed images. Note also, that the poorly extracted

Table 3. Classification Results by Pre-processed Image Quality

Body Contour Quality	# of Subjects	Correct Classification Rate (CCR)		
		k=1	*k=3*	*k=5*
Class A (good)	30	96.7%	93.3%	96.7%
Class B (fair)	30	93.3%	90.0%	86.7%
Class C (bad)	30	90.0%	80.0%	90.0%
Class A + B	60	91.7%	86.7%	85.0%
Class A + C	60	90.0%	83.3%	90.0%
Class B + C	60	83.3%	80.0%	80.0%
All Subjects	90	85.6%	81.1%	84.4%

silhouettes appear to reduce recognition capability (rather than the good extraction can improve that by poor extraction) since the recognition rate of the well and the poorly extracted silhouettes follows that of the poorly extracted silhouettes more closely.

Also, a back-propagation neural network algorithm is applied to the SOTON indoor database as in Table 3 for identifying humans by their gait. The 10 selected gait features for each subject are used as input data, and the numbers of hidden nodes and output nodes are set to 28 and 13 respectively. The neural networks are trained until recognition on the training data reached 100 percent, thus the classification rates for each group of the training sets were 100%. The neural network approach achieved a recognition rate of up to 90% for 30 subjects. Although the recognition rate does not reach 100%, this preliminary study does suggest the possibility of this gait biometric. Naturally we seek to improve the recognition rate in future.

5 Conclusions

We have described a feature extraction and selection method for recognizing humans by their gait. To recognize the human gait, two groups of motion parameters, a total of 20 parameters, are considered as gait features. These are including general and kinematic parameters. By measuring a class separability of the given feature, only 10 important features for classifying the gait are selected from these feature sets. To analyze the discriminatory capability of the selected features, a *k*-NN algorithm is employed as classifier. Here, higher gait classification performances, which are 96.7% CCR for 30 subjects and 84% CCR for 100 subjects, have been achieved on the larger SOTON database. The results achieved give promising performance and higher recognition rates than those of an earlier gait recognition approach. As such, a marker-less gait recognition system confirms uniqueness – as earlier suggested in biomechanical studies, thus confirming validity of the new methods. In future, we require more efficient features of gait motion for classifying and recognizing the gait activities, and a more sophisticated recognition algorithm.

References

1. Gavrila, D. M.: The Visual Analysis of Human Movement: A Survey. Computer Vision and Image Understanding. **73**(1). (1999) 82-98
2. Murray, M. P.: Gait as a Total Pattern of Movement. American Journal of Physical Medicine. **46**(1). (1967) 290-333
3. Inman, V. T., Ralston, H. J., Todd, F.: Human Walking. Williams & Wilkins (1981)
4. Winter, D. A.: The Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological. 2nd edn. Waterloo Biomechanics. (1991)
5. Aggarwal, J. K., Cai, Q., Liao, W., Sabata, B.: Nonrigid Motion Analysis: Articulated and Elastic Motion. Computer Vision and Image Understanding. **70**(2). (1998) 142-156
6. Johansson, G.: Visual Perception of Biological Motion and a Model for Its Analysis. Perception and Psychophysics. **14**(2). (1973) 201-211
7. Stevenage, S. V., Nixon, M. S., Vince K.: Visual Analysis of Gait as a Cue to Identity. Applied Cognitive Psychology. **13**. (1999) 513-526
8. Nixon, M. S., Tan, T., Chellappa, R.: Human Identification based on Gait. Springer (2006)
9. Chau, T.: A Review of Analytical Techniques for Gait Data. Part 1: Fuzzy, Statistical and Fractal Methods. Gait and Posture. **13**(1). (2001) 49-66
10. Perry, J.: Gait Analysis: Normal and Pathological Function. Slack (1992)
11. Whittle, M. W.: Gait Analysis: An Introduction. 3rd edn. Butterworth Heinemann. (2002)
12. Kim, W.-J., Kim, S.-D., Kim, K.: Fast Algorithms for Binary Dilation and Erosion Using Run-Length Encoding. ETRI Journal. **27**(6). (2005) 814-817
13. Dempster, W. T., and Gaughran, G. R. L.: Properties of Body Segments Based on Size and Weight. American Journal of Anatomy, **120**. (1967) 33-54
14. Norkin, C. C., Levangie, P. K.: Joint Structure and Function: A Comprehensive Analysis, 2nd edn. F. A. Davis Company (1992)
15. Chang, C., Huang, C. L.: The Model-Based Human Body Motion Analysis System. Image and Vision Computing. **18**(14). (2000) 1067-1083
16. Sadeghi, H., Allard, P., Prince, F., Labelle, H.: Symmetry and Limb Dominance in Able-Bodied Gait: A Review. Gait and Posture. **12**(1). (2000) 34-45
17. Goswami, A: A New Gait Parameterization Technique by Means of Cyclogram Moments: Application to Human Slop Walking. Gait and Posture. **8**(1). (1998) 15-36
18. Looney, C. G.: Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists. Oxford University Press (1997)
19. Dash, M., Liu, H.: Feature Selection for Classification. Intelligent Data Analysis. **1**(3). (1997) 131-156
20. Guorong, X., Peiqi, C., Minhui, W.: Bhattacharyya Distance Feature Selection. Proceedings of the 13th International Conference on Pattern Recognition. (1996) 195-199
21. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd edn. Academic Press (1990)
22. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press. (1999)
23. Hwang, T. H., Cho, S. I., Park, J. H., Choi, K. H.: Object Tracking for a Video Sequence from a Moving Vehicle: A Multi-modal Approach. ETRI Journal. **28**(3). (2006) 367-370
24. Friendly, M.: Corrgrams: Exploratory Displays for Correlation Matrices. The American Statistician. **56**(4). (2002) 316-325

Rectification of Illumination in Images Used for Shape from Focus

Mannan S.M., Aamir Saeed Malik, Humaira Nisar, and Tae-Sun Choi

Gwangju Institute of Science and Technology,
1-Oryong Dong, Buk Gu, Gwangju, 500-712, Korea
{mannan, aamir, humaira, tschoi}@gist.ac.kr

Abstract. Shadows occurring in images often lead to undesirable results in computer vision. An inherent weakness of 3D reconstruction from image focus is that, they require the imaged scene to have significant textures. In many real-world applications, surfaces can be smooth and lacking in detectable texture. In such cases, Shape-form-Focus (SFF) generates inaccurate and sparse depth maps. For accurate calculation of depth map, we consider the effects of illumination. In SFF the images are taken with one light source and the shadows occur in each frame. The more bright regions of the images give more accurate depth map, whereas, the less bright regions give less accurate depth map. In this paper we propose an algorithm that removes the shadows from the image sequence which are used for SFF methods. We show the results and compare them with the previous results. From simulation results, the depth maps of objects are improved when the shadows are removed.

Keywords: Shape from Focus (SFF), shadow removal, illumination.

1 Introduction

Several active and passive methods have been proposed for recovering 3D shape of objects from their two-dimensional images. In active methods sonar, laser range finders etc. are included; and, in passive methods, Binocular Stereo, Motion Parallax, Structure from Motion, Shape/Depth from Focus/Defocus, Shape from Shading, etc. are included.

An important advantage of shape/depth from focus is that, unlike stereo and motion, it is not confronted with correspondence problem. We have used the SFF algorithms to recover 3-D shape of the object. A lot of research has been done on the image focus analysis to automatically focus the imaging system or to obtain the sparse depth information from the observed scene [6], [7], [8], [9], [10], [11], [12], [13], [16], [17]. Most previous research on SFF concentrated on the developments and evaluations of different focus measures.

In Shape from Focus (SFF) methods, a sequence of images are used, taken by a single camera at different focus levels to compute the depth of the object in the scene. Then, the entire image sequence is searched to find the best focused image frame for a

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 166 – 175, 2006.
© Springer-Verlag Berlin Heidelberg 2006

particular point in the image space; camera parameter settings for that image frame are used to compute the distance of corresponding object point by using the Lens-Formula (equation 1). Where f is focal length, u and v, are the object and image distance from the lens, respectively.

$$\frac{1}{f}=\frac{1}{u}+\frac{1}{v} \tag{1}$$

Since this method involved focusing the object for finding 3-D shape, so it is called Shape from Focus (SFF). The equation of the blur radius caused by the focusing of the object surface is given by the equation 2, where R is the radius of blur circle, f is the focal length of the lens, D is the diameter of the lens; u and s, are the object and actual image distance from the lens, respectively. If $s=v$, then $R=0$, and equation 2 becomes equal to equation 1.

$$R=\frac{qD}{2}=s\frac{D}{2}(\frac{1}{f}-\frac{1}{u}-\frac{1}{s}) \tag{2}$$

SFF methods are inherently slow and computationally intensive. S. K. Nayar and Y. Nakagawa [16], [17] suggested using interpolation to compute more accurate depth estimates. M. Subbarao and T. S. Choi [11] proposed an accurate method for the recovery of 3-D shape from image focus using Focused Image Surface. This method is called SFF.FIS, gives much more accurate shape estimates than the traditional SFF methods. Another method searches the FIS using neural networks proposed by Muhammad Asif and T. S. Choi [8]. The method represents the 3-D shape of FIS in a window in terms of Neural Network Weights. One of the other methods proposed by Muhammad Bilal Ahmed and Tae Sun Choi [9] is based on Dynamic Programming.

All these techniques use focus measures to calculate depth maps. Famous Focus-Measures (FM) are Laplace Operator, The Tenengad FM, Grey Level Variance, Mean Method FM, Curvature FM, Point FM and Sum Modulus Difference [6], [11], [17].

2 Effects of Illumination on sequence of images used for SFF

An inherent limitation of focus-based methods is that they require the imaged scene to have significant texture. In many real-world applications, surfaces can be smoothly shaded and lacking in detectable texture. In such cases, Shape form Focus generates inaccurate and sparse depth maps. For accurate calculation of depth map, we consider the effects of illumination [16], [17].

The interaction between light and surface is complex and introduces many unwanted artifacts into an image. For example, shading, shadows, and inter-reflections, all make it more difficult to achieve basic visual tasks such as image segmentation, object recognition, and tracking.

In SFF the images are taken with one light source and the shadows occur in each frame. Thus, the more bright regions of the images give more accurate depth map, where as, the less bright regions give less accurate depth map. Fig. 1 shows how light

affects the acquired images. Each frame has more brightness on upper side and less brightness on the lower side. When the edges are extracted from these images the upper side gives good edges and lower side gives almost no edges. Therefore, the shadows in these images must be removed and the images must be enhanced.

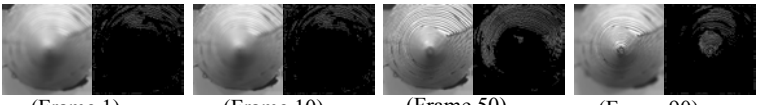


Fig. 1. Different frames of real cone with different focal lengths, and their extracted edges

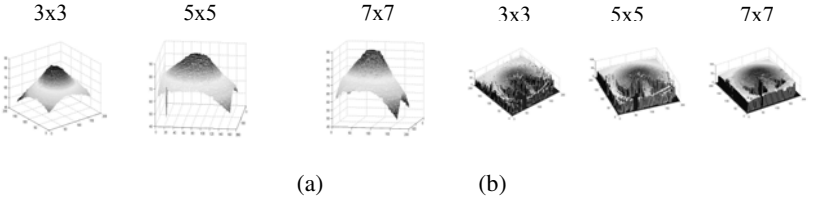


Fig. 2. Depth Maps computed from (a) Grey Level; and (b) Sum of Modified Laplacian using various window sizes

In Fig. 2a, and Fig. 2b, the depth map is computed using Grey Level Variance or GLV and Sum of Modified Laplacian, or SML, respectively. These figures clearly show that, the regions with bad illumination give bad results. When the window size is increased the shape smoothens but the values of the reconstruction starts to mismatch with the original values and thus is inaccurate.

In Table 1 and Fig. 3, we have compared the average values of pixels in well and poorly illuminated regions with various window sizes using both SML and GLV methods. We can see from the figure that the pixel values for both SML and GLV are small and are almost constant for good illuminated region, and for bad illuminated region the pixel values changes significantly for both the methods. Hence it proves our view point that the pixel values in darker region changes quite appreciably with the change in window size, where as in well illuminated region these values do not change that much. The change in values of pixels in poorly illuminated region is due to the fact that large window size incorporates some pixels of quite dissimilar values too that lie far from the pixel in consideration. In case of SML, the difference in the average normalized value of pixel in poorly illuminated region is more than double when the window size is increased from 3x3 to 15x15. And, in case of GLV the difference is about 15 to 20%, thus introducing a big error in actual measurement of depth. While in case of well illuminated region the average normalized value of pixel remains almost consistent.

Table 1. Comparison of average normalized value of pixels in well and poorly illuminated region

Window Size	SML		GLV	
	Well Illuminated Region	Poorly Illuminated Region	Well Illuminated Region	Poorly Illuminated Region
3x3	0.895522388	0.373134328	0.895522388	1
5x5	0.880597015	0.507462687	0.895522388	0.955223881
7x7	0.850746269	0.597014925	0.865671642	0.925373134
9x9	0.850746269	0.656716418	0.850746269	0.910447761
11x11	0.850746269	0.746268657	0.850746269	0.865671642
13x13	0.865671642	0.746268657	0.865671642	0.865671642
15x15	0.835820896	0.76119403	0.865671642	0.820895522

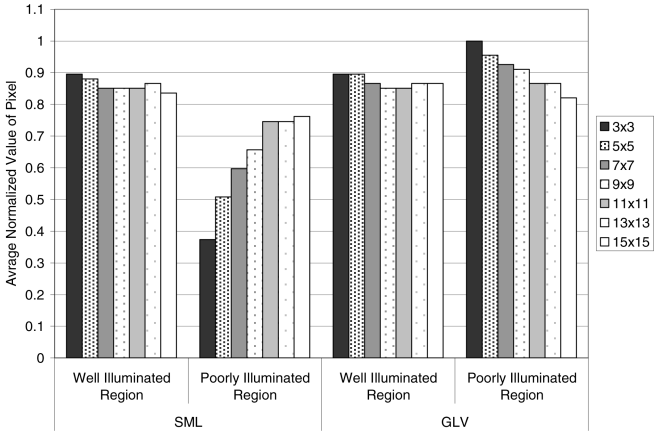


Fig. 3. Comparison of average normalized value of pixels; in well and poorly illuminated regions

3 What Are Shadows?

Shadows are defined as "An area that is not or partially irradiated or illuminated because of the interception of radiation by an opaque object between the area and the source of radiation". A cast shadow consists of two discernible parts: the umbra and the penumbra. The penumbra is a fringe region of half shadow resulting from the partial obstruction of light rays by an object, while the umbra represents the shadow of the complete obstruction. An ideal light source generates only umbrae in shadows. However, penumbrae are not suppressible in the real world condition.

Shadow in an image is a function of shape, reflectance and illumination. Several factors are required to conclude the presence of shadows in an image: the knowledge of geometric information, the existence of obstructions, and the characteristics of material(s) and light source. Since the knowledge of these factors cannot be readily obtained under real world conditions, it is still a difficult task to identify or eliminate shadows from the image.

Despite all these difficulties, a number of approaches have been studied to overcome the problem of detecting shadow regions. Most of these shadow-related algorithms only provide the location of shadows, and many don't provide a complete solution for applications, that must suppress the shadows invisibly.

Offset-Reduction technique and Shadow Flash technique by J. J. Yoon, C. Koch, R. J. Ellis [4], addresses the problem of removing shadows in an actively illuminated environment by a light source with infinite dimensions.

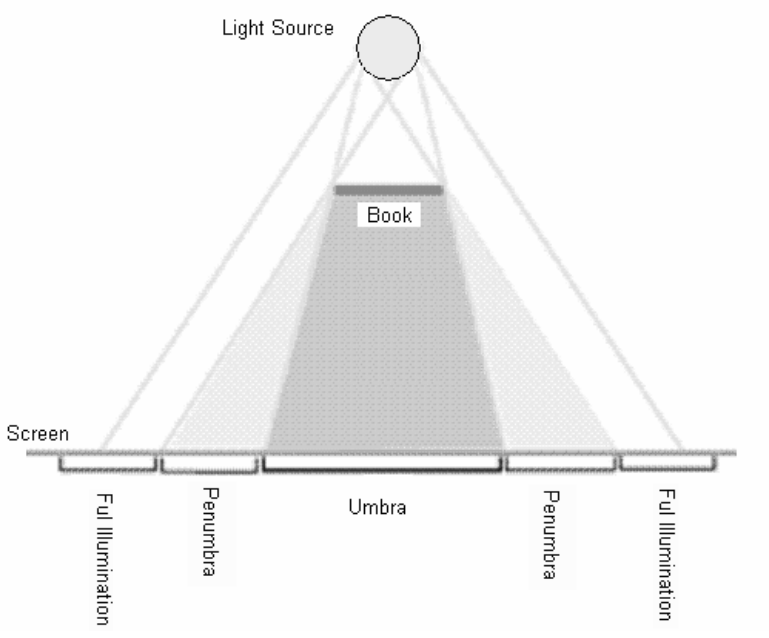


Fig. 4. Shadow types

The Shadow Density technique is defined as a measure of brightness [1]. Using the shadow density, the image is segmented into several regions that have the same density, and then the brightness modification is done to remove the shadow. Other contribution includes the works by [1], [3], [4], [14], [15].

4 Rectification of Illumination for SFF

The necessity of removal of shadows from the sequence of images used for the SFF algorithms has been established in the last section. In this section the proposed algorithm for the removal of shadows in the sequence of images for SFF is described.

The proposed Shadow Removal algorithm (based on shadow density technique [1]) is:

1 Convert RGB images into Grey scale images.
2 Read images of Grey scale values and define intensity ≡ I.

3 Normalize I → 0~1.

$$I_{norm} = \frac{I}{I_{\max}} \tag{3}$$

4 Calculate Shadow Density (S).

$$S = 1 - I_{norm} \tag{4}$$

S → 1: for bad illuminated pixels (Umbra and Penumbra Regions),
S → 0: for well illuminated pixels.

5 Quantize into regions.
6 Extract regions.
7 Region enhancement.
8 Save new enhanced-image.
9 Repeat for all images.
10 Save new sequence of enhanced images.

The quantization of images, that is, to limit the values into a discrete set of values, helps us to separate and extract umbra and penumbra regions. Fig. 5 shows the original and quantized images.

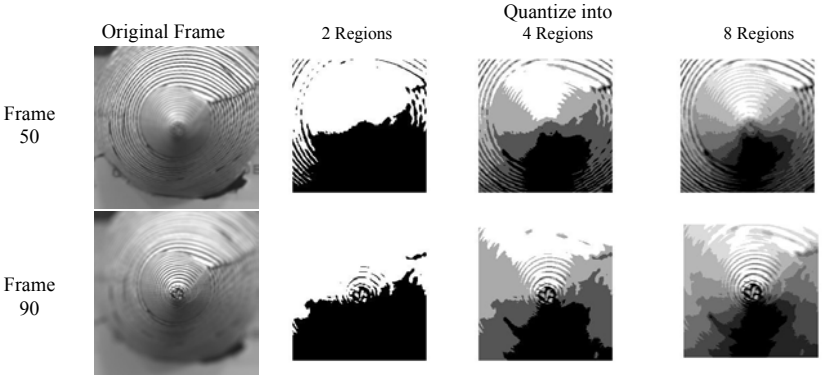


Fig. 5. Images quantized into 2, 4 and 8 regions

After the quantization the regions are extracted. We quantized images into 4 regions. The regions are defined as follows:

Region 1: $R_1 = \{i(x,y): 0 \le S < h_1\}$; (Illuminated Region (sunshine region)).
Region 2: $R_2 = \{i(x,y): h_1 \le S < h_2\}$; (Penumbra 2 shadow).
Region 3: $R_3 = \{i(x,y): h_2 \le S < h_3\}$; (Penumbra 1 shadow).
Region 4: $R_4 = \{i(x,y): h_3 \le S < h_4\}$; (Umbra shadow)

Where, h_k is defined by the following formula

$$h_k = k \cdot \frac{S_{\max}}{N} \tag{5}$$

Where, N is the total number of segmented areas, (in our case N=4).

After the Shadows are detected and separated into regions, the pixel values of the shadow regions (Umbra and Penumbra) are enhanced, by enhancing the contrast of images by transforming the values in the intensity image I. Region R1 is left as it is, R2 and R3 are enhanced less than R4, and R4 is enhanced, most of all regions.

After the enhancement of regions the enhanced images are combined, using bilinear interpolation in order to eliminate artificially induced boundaries; and then saved. The Fig. 6 shows the images before and after the shadows are removed from the images. It clearly shows that the proposed algorithm for shadow removal (especially for the image sequence used for SFF), improves the quality of the images and removes the shadows efficiently.

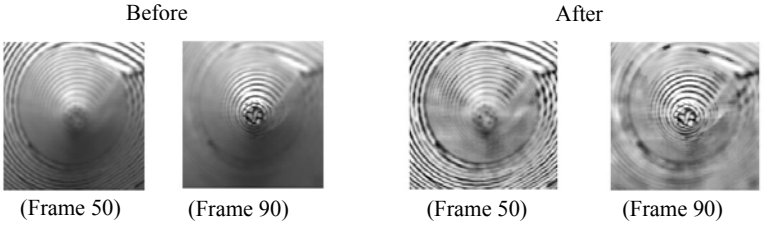


Fig. 6. Images; before and after the removal of shadows and image enhancement

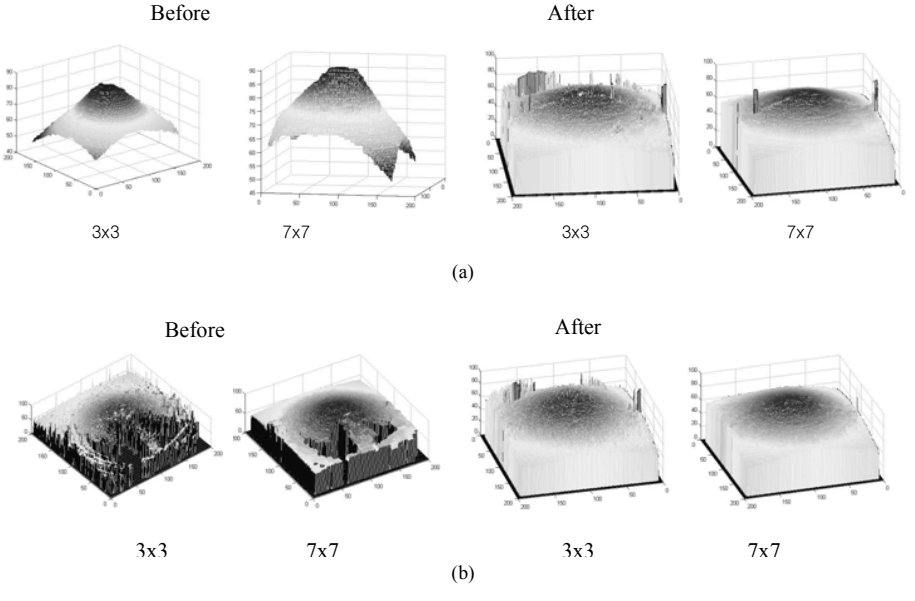


Fig. 7. Depth Map Comparison using (a) GLV, (b) SML; (with window sizes 3x3 and 7x7, before and after shadow removed)

5 Depth Map Computation with New Sequence of Images

Using the shadow removed images for computing depth map and eventually for recovery of 3D shape following results are obtained which are shown in Fig. 6.

The depth maps are computed using Grey Level Variance and Sum of Modified Laplacian, with 3x3, 5x5 and 7x7 window sizes. The result shows that the enhancement of images improves the reconstruction of 3-D shape using SFF algorithms.

In Fig. 7a, the results are shown using GLV, with window sizes 3x3, 5x5 and 7x7, before and after the removal of shadows. The comparison of the SML before and after the shadow removal is shown in Fig. 7b. It is clear that the results are greatly improved when the shadows are removed; even with the small window size the 3D

Table 2. Comparison of average normalized value of pixels; using SML and GLV methods

	SML			GLV		
Window Size	Well Illuminated Region	Poorly Illuminated Region		Well Illuminated Region	Poorly Illuminated Region	
		Shadow Removal			Shadow Removal	
		Before	After		Before	After
3x3	0.895522388	0.373134328	0.823762687	0.895522388	1	0.823877612
5x5	0.880597015	0.507462687	0.823146269	0.895522388	0.955223881	0.824528358
7x7	0.850746269	0.597014925	0.823086567	0.865671642	0.925373134	0.824195522

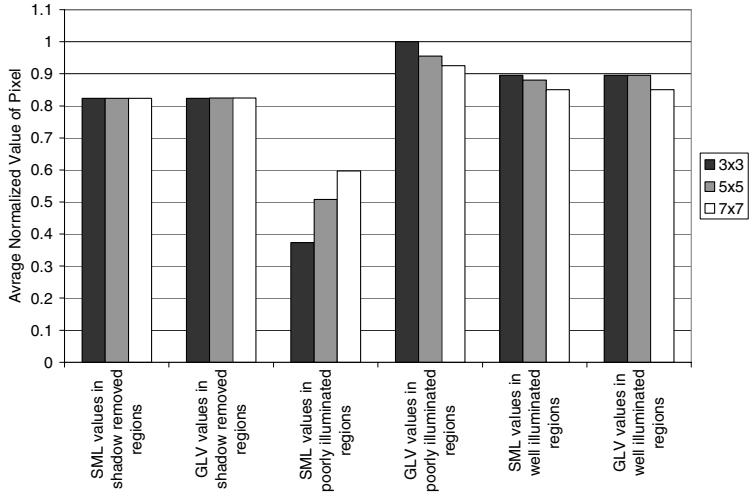


Fig. 8. Comparison of average normalized value of pixels; in well and poorly illuminated regions; using SML and GLV methods, before and after the shadow removal

shape recovery is better. The graphical representations of the results are shown in Table 2 and Fig. 8.

It is clear from the Fig. 8 that the enhancement of the poorly Illuminated regions, not only enhanced the image but also improves the results of 3-D reconstruction.

6 Conclusion

In this paper the necessity of explicit detection and efficient removal of the shadows from the image sequence used for SFF methods have been discussed.

We proposed new shadow removal algorithm specifically for SFF. The proposed algorithm removes shadows from the sequence of images. These shadow-removed images are then provided for SFF algorithms. The results of these shadow removed images are compared with the results of original images. We made the comparisons and showed that the improvement of images give good depth maps, even when small window sizes are used.

Acknowledgement

This work was supported by Korea Research Foundation Grant under Grant No. KRF-2004-041-D00640.

References

1. Baba M., Mukunoki M., Asada N, “Shadow removal from a real Image based on shadow density.”, Proceedings of SIGGRAPH 2004 conference on Sketches and Applications.
2. Franz Stephan Helmli, Stefan Scherer, “Adaptive Shape from Focus with and Error Estimation in Light Microscopy”, Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, ISAP 2001, pp. 188-193, June 2001.
3. Graham D. Finlayson, Steven D. Hordley, Cheng Lu, Mark S. Drew, “On the Removal of Shadows from Images”, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 28, No. 1, January 2006.
4. J. J. Yoon, C. Koch, R. J. Ellis, “Shadow Flash: an approach for shadow removal in an active environment”, In Proceedings of the 13th British Machine Vision Conference, pp. 636-645, September 2002.
5. M. Subbarao, T. Choi, A. Nikzad, “Focusing Techniques”, Optical Eng. Vol. 32, no. 11, pp. 2824-2836, Nov. 1993.
6. Minori Noguchi and Shree K. Nayar, “Microscopic Shape from Focus Using Active Illumination,” IEEE International Conference on Pattern Recognition, pp. 147-152, October 1994.
7. Muhammad Asif, Aamir Saeed Malik, Tae Sun Choi, “3-D Shape Recovery from Image Defocus Using Wavelet Analysis”, ICIP 2005, IEEE International Conference on Image Processing, Vol. 1, September 2005.
8. Muhammad Asif, Tae Sun Choi, “Shape from Focus using Multilayer Feed-forward Neural Networks”, IEEE Transactions on Image Processing, Vol.10, No.11, November 2001.

9. Muhammad Bilal Ahmed, Tae Sun Choi, "A Heuristic Approach for Finding Best Focused Shape", IEEE Transactions on Circuit and System for Video Technology, Vol. 15, No. 4, April 2005.
10. Muhammad Bilal Ahmed, Tae Sun Choi, "Fast and Accurate 3-D Shape from Focus Using Dynamic Programming Optimization Technique", ICASSP 05, IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 2, March 2005.
11. Murali Subbarao, Tae Choi, "Accurate Recovery of Three-Dimensional Shape from Image Focus", IEEE Trans .on Pattern Analysis and Machine Intelliegence, Vol 17. No. 3, pp 266-274, March 1995
12. S. K Nayar, K. Ikeluchi, T Kanade, "Surface Reflection: physical and Geometrical Perspective", IEEE Trans. PAMI, Vol.13, No. 7, pp. 611-634, 1991.
13. S. Nayar, "Shape from Focus System for Rough Surface", Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition, Champaign, III., pp. 302-308, June 1992.
14. Sato I., Sato Y., Ikeuchi K., "Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions", ICCV (2), 875-882.
15. Sato, Y. Sato, K. Ikeuchi. "Illumination distribution from brightness in shadows: Adaptive Estimation of Illumination distribution with unknown reflectance properties in shadow regions." In Proceedings of ICCV' 99, pp. 822-875. 1999.
16. Shree K. Nayar, Nakagawa, Y "Shape from Focus System", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, Issue 8, pp.824-831, August 1994.
17. Shree K. Nayar, Yasuo Nakagawa, "Shape from Focus: An Effective Approach for Rough Surfaces". Proc. IEEE conf. on Computer Vision and Pattern Recognition, pp. 302-308, June 1992.
18. Tae Sun Choi, Muhammad Asif, Joungil Yun, "Three-Dimensional Shape Recovery from Focused Image Surface", ICASSP 99, IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 6, March 1999.

Bilateral Edge Detection on a Virtual Hexagonal Structure

Xiangjian He[1,2], Wenjing Jia[1], Namho Hur[2], Qiang Wu[1], Jinwoong Kim[2], and Tom Hintz[1]

[1] Computer Vision and Visualization Research Group
University of Technology, Sydney
Australia
{sean, wejia, wuq, hintz}@it.uts.edu.au
[2] Broadcasting System Research Group
Digital Broadcasting Research Division
Electronics Telecommunications Research Institute
Korea
{namho, jwkim}@etri.re.kr

Abstract. Edge detection plays an important role in image processing area. This paper presents an edge detection method based on bilateral filtering which achieves better performance than single Gaussian filtering. In this form of filtering, both spatial closeness and intensity similarity of pixels are considered in order to preserve important visual cues provided by edges and reduce the sharpness of transitions in intensity values as well. In addition, the edge detection method proposed in this paper is achieved on sampled images represented on a newly developed virtual hexagonal structure. Due to the compact and circular nature of the hexagonal lattice, a better quality edge map is obtained on the hexagonal structure than common edge detection on square structure. Experimental results using proposed methods exhibit encouraging performance.

Keywords: Edge detection, image processing, image analysis, Gaussian filtering, hexagonal image structure.

1 Introduction

Computer vision involves compositions of picture elements (pixels) into edges, edges into object contours and object contours into scenes. The determination of edges depends on detection of edge points (pixels) of a 3-D physical object in a 2-D image. This first step in the process is critical to the functioning of machine vision. As the success of subsequent steps are sensitive to the quality of results at this step, the performance of higher level processes such as extraction of object contours and object recognition relies heavily on the complete and correct determination of edges. Edges contain major image information and need only a small amount of memory storage space compared to the original image. Hence, edge detection simplifies images and thus facilitates image analysis and interpretation.

Edge detection is based on the relationship a pixel has with its neighbours. It extracts and localizes points (pixels) around which a large change in image brightness has

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 176 – 185, 2006.
© Springer-Verlag Berlin Heidelberg 2006

occurred. A pixel is unsuitable to be recorded as an edge if the brightness around a pixel is similar (or close). Otherwise, the pixel may represent an edge.

During the last three decades, many algorithms have been developed for edge detection e.g. Roberts edge detector, Sobel edge detector, Marr-Hildreth edge detector, etc. All these edge detectors resolve some problems but still have their disadvantages. In 1986, Canny [1] developed an optimal edge detection scheme using linear filtering with a Gaussian kernel to suppress noise and reduce the sharpness of transition in intensity values. In order to recover missing weak edge points and eliminate false edge points, two edge strength thresholds are set to examine all the candidate edge points. Those below the lower threshold are marked as non-edge. Those which are above the lower threshold and can be connected to points whose edge strengths are above the higher threshold through a chain of edge points are marked as edge points [2].

However, the performance of Canny edge detection relies on Gaussian filtering. Gaussian filtering not only removes image noise and suppresses image details but also weakens the edge information [2]. In this paper, an additional filter called range filter [3] is combined with the conventional Gaussian filter to get a bilateral filter in order to reduce the blur effect using the Gaussian filter only.

Moreover, the success of Canny edge detection is often limited when it comes to curved features. This is primarily due to the algorithm's effectiveness being affected by the quality of input image based on square structure. Although using high resolution square grids is able to represent the data with sufficient fidelity, it also increases the computational complexity [2]. In this paper, a new edge detection algorithm is implemented on a virtual hexagonal structure, which was only introduced recently in [4]. Hexagonal lattice promises better efficiency and less aliasing [5]. In addition, it also makes humans less sensitive to edges in the diagonal rather than horizontal or vertical direction [6].

The organization of this paper is as follows. A virtual hexagonal structure is introduced in Section 2 followed by the edge detection using a bilateral filter in Section 3. Section 4 presents the experimental results. Conclusions are given in Section 5.

2 A Virtual Hexagonal Structure

The possibility of using a hexagonal grid to represent digital images has been studied for more than thirty years. Hexagonal grids have higher degrees of symmetry than the square grids. This symmetry results in a considerable saving of both storage and computation time [5, 7]. In order to properly address and store hexagonal images data, Sheridan [8] proposed a one-dimensional addressing scheme for a hexagonal structure, called Spiral Architecture, as shown in Fig. 1.

For the whole image, following the spiral-like curve, as shown in Fig. 1, one can find out the location of any hexagonal pixel with a given spiral address starting from the central pixel of address 0. From Fig. 1, as shown in [4], it is easy to see that the location (denoted by L) of the pixel with a given spiral address

$$a_n a_{n-1} \cdots a_1, \ (a_i = 0,1,2,\cdots,6 \text{ for } i = 1,2,\cdots,n.)$$

can be found from the locations of

$$a_i \times 10^{i-1} \text{ for } i = 1,2,\cdots,n.$$

using

$$L(a_n a_{n-1} \cdots a_1) = \sum_{i=1}^{n} L(a_i \times 10^{i-1}). \tag{1}$$

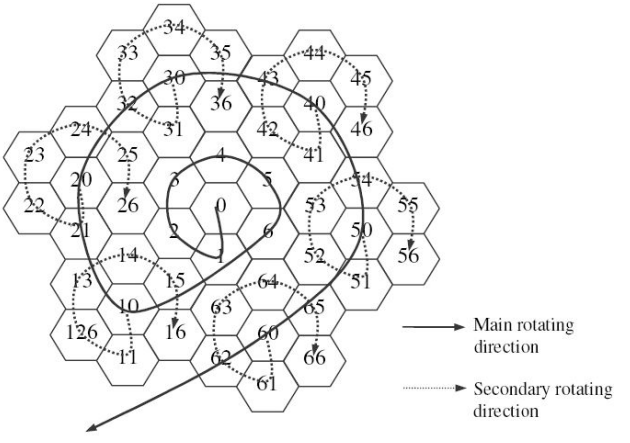


Fig. 1. Spiral Architecture with spiral addressing

Because there has been no hardware available for image display and capture on hexagonal structure, in this section, we introduce a software approach to the construction of a hexagonal structure [4]. To construct hexagonal pixels, each square pixel is first separated into 7×7 small pixels, called *sub-pixels*. To be simple, the light intensity for each of these sub-pixels is set to be the same as that of the pixel from which the sub-pixels are separated. Each virtual hexagonal pixel is formed by 56 sub-pixels as shown in Fig. 2. To be simple, the light intensity of each constructed hexagonal pixel is computed as the average of the intensities of the 56 sub-pixels forming the hexagonal pixel. Fig. 2 shows a collection of seven hexagonal pixels constructed with spiral addresses from 0 to 6.

Let us assume that original images are represented on a square structure arranged as 2*M* rows and 2*N* columns, where *M* and *N* are two positive integers. Let the centre of the virtual hexagonal structure be located at the middle of rows *M* and *M+1*, and at column *N*. Note that there are 14*M* rows and 14*N* columns in the (virtual square) structure consisting of virtual sub-pixels obtained from the original square pixels. Let us construct the first hexagonal pixel using the 56 sub-pixels with centre located in the middle of rows 7*M* and 7*M*+1 and the column 7*N* of the virtual square structure. After the 56 sub-pixels for the first hexagonal pixel are allocated, all sub-pixels for all hexagonal pixels can be assigned from (1). It is worth to note that unlike the work done in [2] on bilateral edge detection, the assignment of sub-pixels to corresponding hexagonal pixels is not required. We do not need to compute the intensities for the virtual hexagonal pixels either. Hence, the computation complexity for edge detection is greatly reduced.

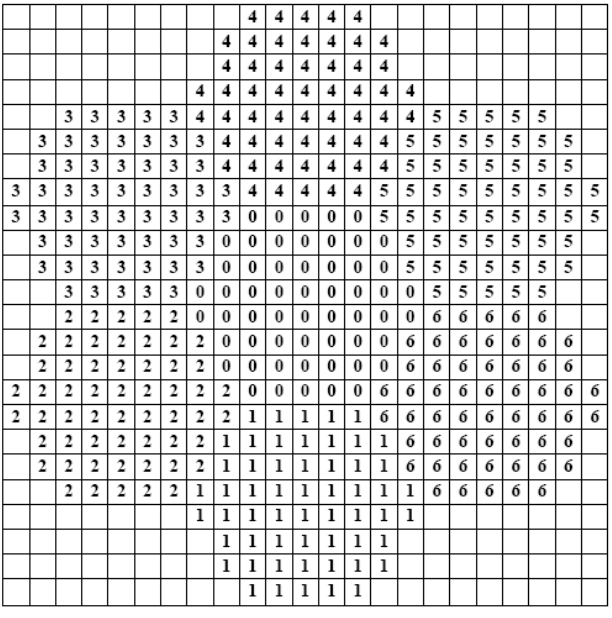


Fig. 2. A cluster of seven hexagonal pixels

3 Edge Detection

In this section, the performance of edge detection will go through three steps: noise filtering using a bilateral filter, edge detection using Sobel operator and edge refining using thresholds.

3.1 Noise Filtering

Before the edge map of an image is found, it is common that image noise is removed (or suppressed) by applying a filter that blurs or smoothes the image.

One commonly used filter is implemented by convolution of the original image function with a Gaussian kernel as defined in Equation (3) below. In order to achieve a more desirable level of smoothing in applications, a bilateral filter has recently been introduced as shown in [2]. Bilateral filtering replaces the intensity value of a pixel with a weighted average intensity value of those pixels that either have similar intensity values as that of the given pixel or are close to the given pixel. In this form of filtering, a *range filter* is combined with a *domain filter*. A domain filter enforces spatial closeness by weighing pixel intensity values with coefficients that fall off as distance of the neighbouring pixel increases [2]. A range filter, on the other hand, assigns greater coefficients to those neighbouring pixel values that are more similar to the given reference pixel value. Hence, the original intensity value at a given pixel would be better preserved after the value replacement, thanks to range filtering. Range filtering by itself is of little use because values of the pixels that are far away from a given pixel should not contribute to the new value. In one word, the kernel

coefficients of a bilateral filter are determined by the combined closeness and similarity function. We explain how a bilateral filter works using mathematical terms as follows [2].

Let $f:\Re^2 \to \Re$ be the original brightness function of an image which maps the coordinates of a pixel, (x, y) to a value in light intensity. Let a_0 be the reference pixel. Then, for any given pixel a at location (x, y), the coefficient assigned to intensity value $f(a)$ at a for the range filter is $r(a)$ computed by the similarity function s as:

$$r(a) = s\left(f(a), f(a_0)\right) = e^{-\frac{\left(f(a)-f(a_0)\right)^2}{2\sigma_1^2}} \tag{2}$$

Similarly, the coefficient assigned for the domain filter is $g(a)$ computed by the closeness function c as:

$$g(a) = c\left(a, a_0\right) = e^{-\frac{(a-a_0)^2}{2\sigma_2^2}} \tag{3}$$

Therefore, for the reference pixel a_0, its new intensity value, denoted by $h(a_0)$, is

$$h(a_0) = k^{-1}\sum_{i=0}^{n-1} f(a_i)\times g(a_i)\times r(a_i) \tag{4}$$

where k is the normalization constant and is defined as

$$k = \sum_{i=0}^{n-1} g(a_i)\times r(a_i) \tag{5}$$

Equation (4) above is called a convolution of the image brightness function f with domain filter g and range filter r. It will take a long time to carry on the convolution processing as shown in (4) if n is large. Considering about 99.5% of energy is found in the central area of "Mexico cap" (the curve of Gaussian function with parameter σ) within the radius of 3σ, in order to increase the computation speed, Equation (4) in this paper is computed only over a small area (called *convolution window*) surrounding each reference pixel and covering the disk with centre at the reference pixel and radius of $3\sigma_2$. In this paper, σ_2 is set to be 1. Therefore, the convolution window is set to be a 7 × 7 pixel block (assuming the distance between two adjacent square pixels is 1) on square structure. Similarly, when an image represented on the virtual hexagonal structure is filtered using either a Gaussian filter or bilateral filter, its window size is set to be 49 too for the weighted average computation using Equations (4) and (5), and the convolution window consists of the 49 hexagonal pixels centred at the reference pixel. Hence, n = 49 in this paper. Furthermore, σ_1, the parameter for the range filtering, is computed as the standard deviation of grey values in input image.

The computation to include the 49 hexagonal pixels for convolution using (4) and (5) on the hexagonal structure is achieved sub-pixel by sub-pixel without the need of conversion from square pixels to hexagonal pixels. The computation is as follows.

For each sub-pixel located in a reference hexagonal pixel (e.g., the central pixel with address 0 in Fig. 2), find the 48 corresponding sub-pixels that are in the 48 hexagonal pixels sounding and centred at the reference pixel. These 48 sub-pixels and the given sub-pixel have the same location in their hexagonal pixels. They are on the

same row and column in the sets of 56 sub-pixels each constituting the corresponding hexagonal pixel. For example, if the given (reference) sub-pixel is at 1st row and 2nd column at hexagonal pixel 0 as shown in Fig. 2, then the sub-pixel at 1st row and 2nd column at hexagonal pixel 1 is the corresponding sub-pixel at hexagonal pixel 1. The computation to find all these 48 corresponding sub-pixels is easy. For example, the sub-pixel at hexagonal 1 is at the location by counting 8 sub-pixels downwards from the reference sub-pixel, and the sub-pixel at hexagonal 2 is by counting 4 sub-pixels downwards and 7 sub-pixels leftwards. After all 6 corresponding sub-pixels in the 6 neighbouring hexagonal pixels are obtained, the 6 corresponding sub-pixels in the hexagonal pixels at spiral addresses 10, 20, …, 60 can be computed in a similar way. For example, the sub-pixel at address 10 is at the location by counting 8 sub-pixels upwards and 14 sub-pixels leftwards from the sub-pixel at address 6. After the sub-pixels at 1, 2, …, 6, 10, 20, …, 60 are all obtained, the remaining sub-pixels can be computed from Equation (1) in Section 2.

After all the 49 sub-pixels are found (some sub-pixels may not be found because they are outside the image area), the new intensity for the reference sub-pixel can then be computed using Equations (2)-(4) above for bilateral filtering, and using Equations (3) and (4) only without including the range values $r(a)$.

3.2 Edge Detection

Conventional Sobel edge detector [9] is an edge detector using a 3×3 mask as an edge operator. It is used widely in image processing area. In order to implement edge detection on the virtual hexagonal structure, a modified Sobel operator, as presented in [2] and shown in Fig. 3, is applied in this paper. Using the modified Sobel operator, edge strength and direction at each sub-pixel can be calculated in the similar way as on the square structure. Again, these are computed sub-pixel by sub-pixel.

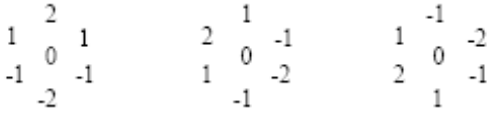


Fig. 3. Modified Sobel operator

3.3 Edge Refining

After the edge detection step shown in Subsection 3.2, all sub-pixels have been assigned new intensity values that show the edge sub-pixels and their strengths. An edge map on the original square structure can hence be obtained by simply computing the intensity value of every square pixel as the average of the intensities of the sub-pixels constituting the square pixel. This edge map shows the square edge pixels and their strengths. We can then follow the remaining steps of Canny's method to obtain the final edge map by using one lower threshold and one higher threshold as stated in Section 1.

4 Experimental Results

To study the effect of new edge detection method on the virtual hexagonal structure and compare with the results obtained on square structure, 8-bit grey level Lena image of size 256×256 is chosen as our sample image to be processed, because it is one of most widely accepted testing image to evaluate the performance of edge detection algorithm (see Fig. 4).



Fig. 4. Original Lena image

Four different edge maps are produced in order to demonstrate the performance improved by new edge detection method. The first edge map is produced by common Canny edge detection through Gaussian filtering on square structure. The second and the fourth edge maps are obtained based on the virtual hexagonal structure. The second edge map is created after the Gaussian filtering and the fourth edge map is produced after the bilateral filtering as shown in Section 3. The third edge map is obtained after the bilateral filtering but based on square structure.

All the above edge maps are generated based on the consistent environment parameters such as σ_1=65 and σ_2=1 for Gaussian filtering and bilateral filtering. Recall that the size of window for carrying out the Gaussian convolution and bilateral processes is 49. In addition, the same lower and higher thresholds are used to locate the exact edge points for all edge maps. The higher threshold is 0.125 and the lower threshold is 0.05.

Fig. 5 shows four images obtained from the original image shown in Fig. 4 using different filtering operations on different image structure. It is shown that changing image structure from square to hexagonal structure does not affect much on the results of Gaussian filtering. However, like the results obtained in [2], bilateral filtering not only smoothes the original image as Gaussian filtering but also enhances the major (or important) edge information.

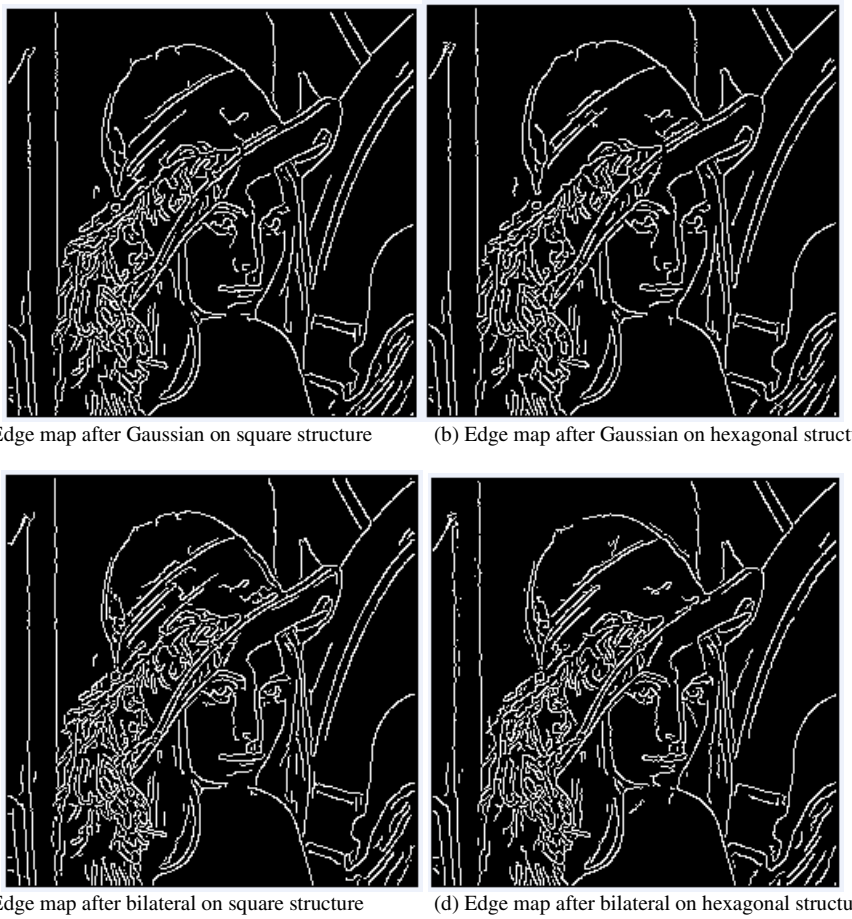


(a) Gaussian filtering on square structure (b) Gaussian filtering on hexagonal structure

(c) Bilateral filtering on square structure (d) Bilateral filtering on hexagonal structure

Fig. 5. Images processed by different filtering operations on different image structures

Fig. 6 demonstrates four edge maps which are obtained after different filtering processes. It is obvious that after Gaussian filtering edge map (Fig. 6(b)) obtained on the hexagonal structure has almost the same but slightly clearer background and edges than the one (Fig. 6(a)) obtained on square structure without losing major edge points. This is mainly due to the uniformly connected and close-packed structure of hexagonal lattice. One more mask defined for the Sobel operator (see Fig. 3) than the conventional operator having two masks on square structure also improves accuracy of edge detection results. Fig. 6(c) and Fig. 6(d) show that bilateral filtering further improves the quality of edge map because it enhances edge area information while suppressing image noise and trivial edge pixels. It can be clearly seen in Figs. 6(c) and 6(d) that the fine details of eyes have been well extracted. It is also found that Fig. 6(d) shows a better performance than Fig. 6(c) for detecting edges in diagonal directions. This can be seen from the lip edges in Fig. 6(d) that are closer to real lip boundaries.



(a) Edge map after Gaussian on square structure (b) Edge map after Gaussian on hexagonal structure

(c) Edge map after bilateral on square structure (d) Edge map after bilateral on hexagonal structure

Fig. 6. Edge maps of the filtered images shown in Fig. 5

5 Conclusions

In this paper, a new edge detection method is presented. The use of bilateral filtering combined with the advantages of hexagonal image architecture has achieved encouraging edge detection performance under the same experimental conditions.

Unlike the work done in [2], our edge detection algorithm is based on a new virtual hexagonal structure and avoids the complicated and time-consuming conversion between the square and hexagonal structure. We take the advantages of higher degree of symmetry and equality of distances to neighbouring pixels that are special to hexagonal structure for better performance of image filtering and more accurate computation of gradients including edges and their strength.

Note that in this paper, in order to simplify the computation, the value of parameter σ_1 used for bilateral edge detection on the virtual hexagonal structure has been set to be the same as the one computed on the square structure. This value, however, should

be different on the hexagonal structure and must be adjusted in the future to improve the performance.

Acknowledgement

This work was partly sponsored by the Australian Research Council under Large ARC-Discovery Grant DP0451666.

References

1. J. F. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679-698, Nov. 1986.
2. Qiang Wu, Xiangjian He and Tom Hintz, "Bilateral Filtering Based Edge Detection on Hexagonal Architecture", *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA, Volume II, 2005, pp.713-716.
3. D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 844-847, 2002.
4. Xiangjian He, Tom Hintz, Qiang Wu, Huaqing Wang and Wenjing Jia "Modified Virtual Spiral Architecture", *International Conference on Image Processing, Computer Vision and Pattern Recognition*, June, 2006, to appear.
5. R. M. Mersereau, "The Processing of Hexagonally Sampled Two-Dimensional Signals" *Proc. the IEEE*, Vol. 67, pp. 930-949, 1979.
6. R. M. Gray, P. C. Cosman, and K. L. Oehler, *Digital Images and Human Vision*. Cambridge, MA: MIT Press, 1993.
7. I. Her, "Geometric Transformations on the Hexagonal Grid," *IEEE Trans. on Image Processing*, vol. 4, 1995.
8. P. Sheridan, T. Hintz, and D. Alexander, "Pseudo-invariant Image Transformations on a Hexagonal Lattice," *Image and Vision Computing*, vol. 18, pp. 907-917, 2000.
9. L. S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics and Image processing*, vol. 4, pp. 248-270, 1975.

Issues and Implementation of C^1 and C^2 Natural Neighbor Interpolation

T. Bobach[1], M. Bertram[2], and G. Umlauf[1]

[1] IRTG, Geometric Algorithms Group, TU Kaiserslautern, Germany
[2] Computer Graphics Group , TU Kaiserslautern, Germany
{bobach, bertram, umlauf}@informatik.uni-kl.de

Abstract. Smooth local coordinates have been proposed by Hiyoshi and Sugihara 2000 to improve the classical Sibson's and Laplace coordinates. These smooth local coordinates are computed by integrating geometric quantities over weights in the power diagram. In this paper we describe how to efficiently implement the Voronoi based C^2 local coordinates. The globally C^2 interpolant that Hiyoshi and Sugihara presented in 2004 is then compared to Sibson's and Farin's C^1 interpolants when applied to scattered data interpolation.

1 Introduction

Scattered data interpolation is the most prominent application for a family of local coordinates in scattered point sets, which were introduced in 1980 by Sibson [1] based on the natural neighbors in Voronoi diagrams. Another, simpler and less smooth set of local coordinates based on natural neighbors has been proposed as non-Sibsonian or Laplace coordinates [2,3,4]. Hiyoshi et al. [5] generalized Laplace and Sibson's coordinates to C^k continuity over the domain except at the data sites. We will call these coordinates for short Hiyoshi's C^k coordinates. Recently, Hiyoshi [6] proposed a stable computation method for his coordinates, taking a different approach than the one we present in this paper.

For scattered data interpolation the discontinuities of the derivative at the data sites can be solved by using a polynomial of the local coordinates. Sibson [7] first proposed a globally C^1 interpolant that interpolates least squares fitted gradients at the data sites. A more general solution was given by Farin [8] who constructed a C^1 interpolant with quadratic precision using multivariate Bézier simplexes. This was extended by Hiyoshi et al. [9] who used higher degree simplexes to build a globally C^2 interpolant based on Hiyoshi's C^k coordinates.

Generalizations of natural neighbor interpolation exist for transfinite interpolation of circles and line segments [10,11,12] and points scattered on manifolds [13]. Efforts have been made in accelerating the computation of Sibson's interpolant using graphics hardware [14,15].

The main contribution of this paper consists of two parts. *First*, we give guidelines for an efficient and robust implementation of Hiyoshi's C^2 coordinates. A comparison to Hiyoshi's approach [6] is not given for we have only recently

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 186–195, 2006.
© Springer-Verlag Berlin Heidelberg 2006

learned of it. *Second*, we compare the resulting interpolant to other smooth scattered data interpolants based on local coordinates, and discuss the importance of derivative estimation in this context. Specifically, we compare Sibson's, Farin's and Hiyoshi's interpolants applied to a downsampled crater lake data set and to data sampled from an analytic function.

We start in Section 2 with some facts about power diagrams as a foundation to briefly restate Hiyoshi's C^k coordinates in Section 3. We then focus on the actual implementation of Hiyoshi's C^2 coordinates in Section 4. The estimation of derivatives up to order two for comparison is discussed in Section 5. Finally, we present a comparison of Sibson's and Farin's C^1 interpolants and Hiyoshi's C^2 interpolant in two different settings in Section 6.

2 Voronoi and Power Diagrams

The scattered data interpolation problem for scalars valued functions is stated as follows. Given a set of data sites $X = \{x_i\}_{i=1,\dots,m} \subset \mathbb{R}^2$ and corresponding data $Z = \{z_i\}_{i=1,\dots,m} \subset \mathbb{R}$, find a function $f : \mathbb{R}^2 \to \mathbb{R}$ satisfying $f(x_i) = z_i$. Further constraints can be stated, e.g. interpolation of given derivatives. Texts on scattered data interpolation in general can be found in [16,17].

Local scattered data schemes compute the interpolated value $f(x)$ by taking a linear combination of the data values in a neighborhood $N(x)$ and weights $\lambda_j(x)$ such that $f(x) = \sum_{j \in N(x)} \lambda_j(x)\, z_j$. The following explains the definition of $N(x)$ in case of natural neighbor interpolation.

Denote by $d(\cdot,\cdot)$ the Euclidean distance in $\mathbb{R}^2$. From the data sites X we construct the Voronoi diagram $\mathcal{V}(X) := \{\mathcal{T}_i\}_{i=1,\dots,m}$, which is the partition of space into convex tiles $\mathcal{T}_i$ such that

$$\mathcal{T}_i = \{x \in \mathbb{R}^2 | d(x, x_i) \leq d(x, x_j), i \neq j\}.$$

See [18,19] for a thorough treatment of Voronoi diagrams. Two generators x_i and x_j are called *natural neighbors* if the intersection $\mathcal{T}_i \cap \mathcal{T}_j$ of their associated tiles is not empty. Because generators on the convex hull of X have unbounded tiles, we restrict our considerations to the interior $\mathscr{D}$ of the convex hull of X. We denote the set of indices of the natural neighbors for a generator x_j by N_j. If at least 3 tiles share a common point, the unique circum-circle through their generators is called *Delaunay circle*, which contains no other generator in its interior.

A power diagram is a Voronoi diagram with weighted generators (x_i, w_i), $w_i \in \mathbb{R}$ and the distance measure $d_p(\cdot,\cdot)$,

$$\big(d_p(x, x_i)\big)^2 = \big(d(x, x_i)\big)^2 - w_i.$$

Then, all notions from the Voronoi diagram carry over: tiles are convex polygons and for uniformly chosen weights, the power diagram coincides with $\mathcal{V}(X)$. For power diagrams tiles can be empty depending on the weights w_i, which does not happen for the special choice of weights w_i in this paper.

Assume we are interested in the natural neighbors N_0 of an arbitrary point $x_0 \in \mathscr{D}$. We treat x_0 as a variable position that moves around in $\mathscr{D}$. Virtually inserting x_0 into X results in a new Voronoi diagram $\mathcal{V}'$ with tiles $\mathcal{T}_i'$. Furthermore, we associate with each generator $x_1, \ldots, x_m$ a zero power weight $w_i := 0$, except for x_0 with power weight $w_0 := -w$. This basically means x_0 loses influence with growing w, and its tile becomes smaller until it vanishes. Note that changing w only affects the shape of $\mathcal{T}_0'(w)$ and $\mathcal{T}_i'(w)$ of its natural neighbors, $i \in N_0$. Although strictly speaking, N_0 also depends on w, we always take it to be as in the standard Voronoi diagram, i.e. $w = 0$. N_0 now only depends on the position of x_0. Expressing x_0 as a linear combination of its neighbors, $x_0 = \sum_{i \in N_0} \lambda_i x_i$, allows the definition of the scattered data interpolant

$$f(x_0) = \sum_{i \in N_0} \lambda_i z_i. \tag{1}$$

Where N_0 is unique, $f(x_0)$ is a C^∞-continuous function. Continuity issues arise whenever N_0 changes, i.e.

1. x_0 crosses a Delaunay circle, or
2. x_0 passes another generator x_i.

The first problem is solved by construction of λ_i, while the second is solved by construction of the interpolant. In the next section we focus on the first problem.

3 Natural Neighbor Coordinates

Natural neighbor coordinates express a point x_0 as a convex combination of its natural neighbors $\{x_i\}_{i \in N_0}$, i.e. $x_0 = \sum_{i \in N_0} \lambda_i x_i$. The weights λ_i are usually derived from the geometry of the Voronoi tiles of x_0 and its neighbors. Laplace coordinates λ_i^0 are defined with respect to the line segments $l_i(w)$ perpendicular to the line connecting x_0 and x_i and bounding $\mathcal{T}_0'(w)$ and the distances $r_i = d(x_0, x_i)$, $i \in N_0$,

$$\tilde{\lambda}_i^0(w) := |l_i(w)|/r_i \quad \text{and} \quad \lambda_i^0(w) := \tilde{\lambda}_i^0(w) / \sum\nolimits_{j \in N_0} \tilde{\lambda}_j^0(w).$$

Laplace coordinates are continuous in $\mathscr{D}$.

Another set of natural neighbor coordinates are Sibson's coordinates λ_i^1 which are C^1 in $\mathscr{D} \setminus X$. Based on the area $v_i(w)$ of $\mathcal{T}_i \cap \mathcal{T}_0'(w)$ they are defined as

$$\tilde{\lambda}_i^1(w) := |v_i(w)| \quad \text{and} \quad \lambda_i^1(w) := \tilde{\lambda}_i^1(w) / \sum\nolimits_{j \in N_0} \tilde{\lambda}_j^1(w).$$

Laplace and Sibson's coordinates are members of a class of natural neighbor coordinates generalized by Hiyoshi et al. [5] to C^k continuity in $\mathscr{D} \setminus X$. The area covered by $\mathcal{T}_0'(w)$ shrinks with growing w until it vanishes for $w_{max} = \inf_w(\mathcal{T}_0(w) = \emptyset)$, see Figure 1(left). Using $v = w_{max} - w$, $k \geq 1$ the recursion

$$\tilde{\lambda}_i^k(u) = \int_0^u \tilde{\lambda}_i^{k-1}(v)\, dv \quad \text{and} \quad \tilde{\lambda}_i^k = \tilde{\lambda}_i^k(w_{max}) \tag{2}$$

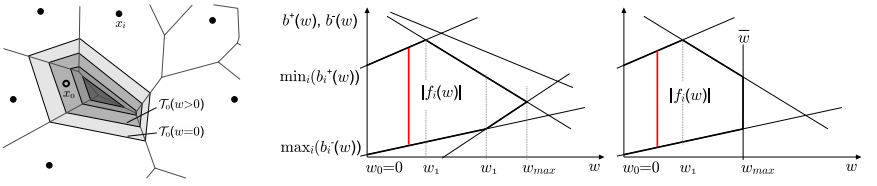


Fig. 1. (left) The Voronoi diagram $\mathcal{V}$ of X (black dots) and the tile $\mathcal{T}_0(w)$ of x_0 for different values of $w \geq 0$. (right) Two examples for $|l_i(w)|$ and its bounding constraints $N_{i,j}y \leq b_{i,j}(w)$.

yields by normalization a relation between the different natural neighbor coordinates,

$$\lambda_i^k := \tilde{\lambda}_i^k / \sum\nolimits_{j \in N_0} \tilde{\lambda}_j^k. \tag{3}$$

These coordinates are called Hiyoshi's coordinates. In $\mathscr{D} \setminus X$ they are C^k in x_0, see [5].

4 Computing Hiyoshi's C^2 Coordinates

In [6] Hiyoshi proposed to compute his coordinates by accumulating the contributions of Delaunay triangles and edges to the value of λ^k. Our approach is different and based on the direct evaluation of equation (2) which requires to express $\lambda^0(w)$ as an analytic expression in x_0 and $\{x_i\}_{i \in N_0}$. Therefore, we first examine how the geometry of $\mathcal{T}_0(w)$ depends on w.

For $w \geq 0$ the distance between $l_i(0)$ and $l_i(w)$ is given by $w/2r_i$. Since the vertices of $\mathcal{T}_0(w)$ are located on the bisectors of $\mathcal{V}$, the length of the line segment $l_i(w)$ is a piecewise linear function on intervals $[w_j, w_{j+1})$, $j \geq 1$,

$$|l_i(w)|_{[w_j, w_{j+1})} = \alpha_{ij} + w\beta_{ij},$$

see for examples Figure 1(right). The tile $\mathcal{T}_0(w) \subset \mathbb{R}^2$ is a convex polygon confined in the intersection of half spaces $n_i x \leq b_i(w), i = 0 \ldots m-1$, $m = |N_0|$ with

$$n_i = (x_i - x_0)/r_i \qquad \text{and} \qquad b_i(w) = c_i + wd_i,$$

with $c_i = (x_i + x_0)n_i/2$ and $d_i = -1/2r_i$. Thus, $\mathcal{T}_0(w)$ can be described by the matrix inequality

$$\mathcal{T}_0(w) = \big\{x \in \mathbb{R}^2 \mid N\,x \leq b(w)\big\}, \quad b(w) \in \mathbb{R}^m, N \in \mathbb{R}^{m \times 2}. \tag{4}$$

Now, l_i is represented by $n_i x = b_i(w)$ which can be used to eliminate $x_j, j = 1, 2$, from (4). This is equivalent to a projection of $l_i(w)$ along the j-th coordinate direction. With a new matrix $N_{i,j}$ and right-hand side $b_{i,j}(w)$ the projected line segment is given by

$$l_{i,j}(w) = \big\{y \in \mathbb{R} \mid \; N_{i,j}y \leq b_{i,j}(w)\big\} \tag{5}$$

which is an intersection of intervals in $\mathbb{R}$. Thus, the length of l_i can be computed as

$$|l_i(w)| = |l_{i,j}(w)|/|n_{i,j}|, \tag{6}$$

where $n_{i,j}$ is the j-th component of n_i.

In case there is a line segment $l_k, k \neq i$, parallel to l_i there is an upper bound $\overline{w} > 0$ for w such that $l_{i,j}(w)$ is not empty. This upper bound $\overline{w}$ is given by the row of $N_{i,j}y \leq b_{i,j}(w)$ representing the constraint imposed by l_k. Otherwise, we set $\overline{w}$ to infinity.

Classifying the inequalities in (5) as upper and lower bounds for the interval $l_{i,j}(w)$, distinguished by superscript + and -,

$$y \leq c_k^+ + wd_k^+ =: b_k^+(w) \quad \text{and} \quad y \geq c_l^- + wd_l^- =: b_l^-(w)$$

yield for the length of $l_{i,j}(w)$

$$|l_{i,j}(w)| = \max_{w \leq \overline{w}} \big\{0, \min_k\{b_k^+(w)\} - \max_l\{b_l^-(w)\}\big\}.$$

This can be efficiently computed by appropriate sorting, intersecting and merging. According to (6) the length of $l_i(w)$ is now represented by a list $(w_j, \alpha_{ij}, \beta_{ij})$.

The integral yielding Hiyoshi's C^2 coordinates can be retrieved by applying recursion (2),

$$\tilde{\lambda}_i^2 = \int_0^{w_{max}} \int_0^u \tilde{\lambda}_i^0(w_{max} - v)\, dv\, du. \tag{7}$$

Because $\tilde{\lambda}_i^0$ is piecewise linear this integration can be done per interval $[w_j, w_{j+1})$. Denote by $S_j(u)$ the inner integral of (7) for $u \in [w_j, w_{j+1})$, that is of the form

$$\begin{aligned} S_{-1}(u) &:= 0, \\ S_j(u) &= S_{j-1}(w_j) + \big((u^2 - w_j^2)d_j/2 + (u - w_j)\, c_j\big)/r_i. \end{aligned}$$

This gives for $\tilde{\lambda}_i^2$ the expression

$$\begin{aligned} \tilde{\lambda}_i^2 = \sum\nolimits_j \Big(&(w_{j+1}^3 - w_j^3)\ d_j/6r_i + (w_{j+1}^2 - w_j^2)\ c_j/2r_i + \\ &(w_{j+1} - w_j)\ \big(S_{j-1}(w_j) - (w_j^2 d_j - 2w_j c_j)/2r_i\big)\Big). \end{aligned}$$

Remark 1. Because $|l_i(w)|$ is piecewise linear the recursive integral reduction in [5] is not applicable.

5 Derivative Estimation

The interpolant derived in (1) is as smooth as the coordinate functions. For natural neighbor coordinates, $\lambda_i^k(x_0)$ is only C^0 continuous at the generators.

Sibson proposed an involved construction of a polynomial in $\lambda_i^1(x_0)$ to also interpolate gradients at the generators, thus constructing a globally C^1 continuous interpolant in [7]. Later, Farin used multivariate Bézier simplexes over local coordinates to interpolate arbitrary higher order derivatives at the generators, and presented a C^1 implementation with quadratic precision in [8]. This approach can be generalized to arbitrary degrees of smoothness and applies to all regular, convex barycentric coordinate functions. The C^2 implementation of this idea was used by Hiyoshi to construct a global C^2 interpolant based on $\lambda_i^2(x_0)$ in [9].

All these interpolants depend on derivative information being available, like in the truncated Taylor expansion of f at x_i,

$$Z_i(x) = z_i + \mathbf{g}_i x + \frac{1}{2} x^T H_i x. \tag{8}$$

In order to estimate first order derivatives, we adopt Sibson's weighted least squares (WLS) fit of gradients $\mathbf{g}_i$ [7]:

$$\sum\nolimits_{j \in N_i} s_{i,j} |z_i + \mathbf{g}_i x - z_j|^2 \to \min . \tag{9}$$

The weights for the neighbors are chosen to be $s_{i,j} := \lambda_j^1 / d(x_i, x_j)$ and $s_{i,j} := 1/d(x_i, x_j)$ where λ_j^1 is not defined. Since the interpolation property fixes $f(x_i) = z_i$, fitting $\mathbf{g}_i$ has two degrees of freedom and $|N_i|$ constraints. With an average of six constraints, $\mathbf{g}_i$ is overdetermined and the least squares solution well defined in general.

To estimate derivative information up to order two in a first approach we fit (8) to $\{z_j\}_{j \in N_i}$, finding $\mathbf{g}_i$ and H_i such that

$$\sum\nolimits_{j \in N_i} s_{i,j} |Z_i(x_j) - z_j|^2 \to \min . \tag{10}$$

The five unknowns are only well-defined for $|N_i| \geq 5$ and $\{x_j\}_{j \in N_i}$ in non-degenerate positions. Even for $|N_i| = 5$, the averaging nature of the WLS method is lost and the interpolant becomes very sensitive to the local data configuration, as can be seen in Figure 3(e).

To stabilize the process a larger neighborhood could be used. As Sibson pointed out, the choice of weights for the WLS method is important. But $s_{i,j}$ has no obvious extension for generators other than the natural neighbors.

In a second approach, we employed a different method that naturally incorporates a larger neighborhood and only depends on $s_{i,j}$. First, we determine $\tilde{\mathbf{g}}_i$ as in (10). Then, we find $\mathbf{g}_i$ and H_i such that

$$\sum\nolimits_{j \in N_i} s_{i,j} (|Z_i(x_j) - z_j|^2 + \|\nabla Z_i(x_j) - \tilde{\mathbf{g}}_i\|^2) \to \min . \tag{11}$$

6 Natural Neighbor Interpolation

The scattered data interpolation methods we are going to compare interpolate given scalar values $Z = \{z_i\}_i$, gradients $G = \{\mathbf{g}_i\}_i$, and Hessians $H = \{H_i\}_i$ at the data sites:

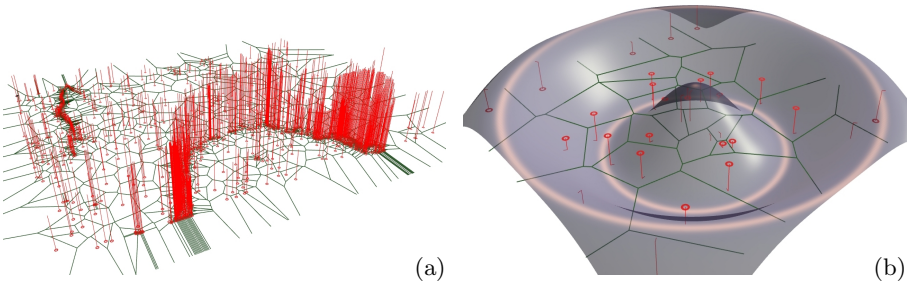


Fig. 2. Input data sets. (a) The original crater lake data set is due to US Geological Survey with $344 \cdot 463$ points. The input to the method is a reduced version with 1493 scattered points. (b) $\cos \|x\|$ sampled at 31 positions.

Sibson's C^0 interpolant for Z: This interpolant is C^0 at X and C^1 in $\mathscr{D} \setminus X$ [7].

Sibson's C^1 interpolant for Z and G: This interpolant is C^1 in $\mathscr{D}$ and reproduces spherical quadratics [7].

Farin's C^1 interpolant for Z and $\tilde{G}$: This interpolant is C^1 in $\mathscr{D}$, and has quadratic precision [8].

Hiyoshi's C^2 interpolant for Z, G, and H: This interpolant is C^2 in $\mathscr{D}$, and has cubic precision [9].

We apply the interpolants and derivative estimation to two data sets. The first is a reduced elevation map of the Crater Lake, see Figure 2(a). The second data set is randomly sampled from the function $\cos \|x\|$ with 0 centered to the shown domain, see Figure 2(b). For both data sets, G in Sibson's and Farin's interpolants has been estimated using equation (9). G and H for Hiyoshi's interpolant have been estimated using both equation (10) and equation (11) to illustrate stability issues. For the analytic data set we furthermore considered the exact values for G and H.

Figure 3(a) shows the Crater Lake data set with piecewise linear interpolation on a Delaunay triangulation for comparison to an established method. Sibson's C^0 interpolant in Figure 3(b) is well suited to capture rough features while Sibson's and Farin's C^1 interpolants in Figure 3(c,d) better deal with smooth regions, giving the best overall impression. There is no noticeable difference between these two methods. If the derivative information for Hiyoshi's C^2 interpolant is aquired from equation (10), strong oscillations occur as in Figure 3(e). Using equation (11) to estimate G and H stabilizes the interpolant in Figure 3(f) but still tends to produce more wriggles than necessary.

The computational complexity of the interpolants is directly related to the order of smoothness they achieve. Sibson's and Farin's C^1 interpolants perform very similar both in quality of the interpolant and in running time, because both interpolate the same gradients and differ only in the polynomial in λ_i^1. Serious performance issues in Hiyoshi's method were observed for the construction of the quintic Bézier control net for large neighborhoods, i.e. $|N_i| > 20$.

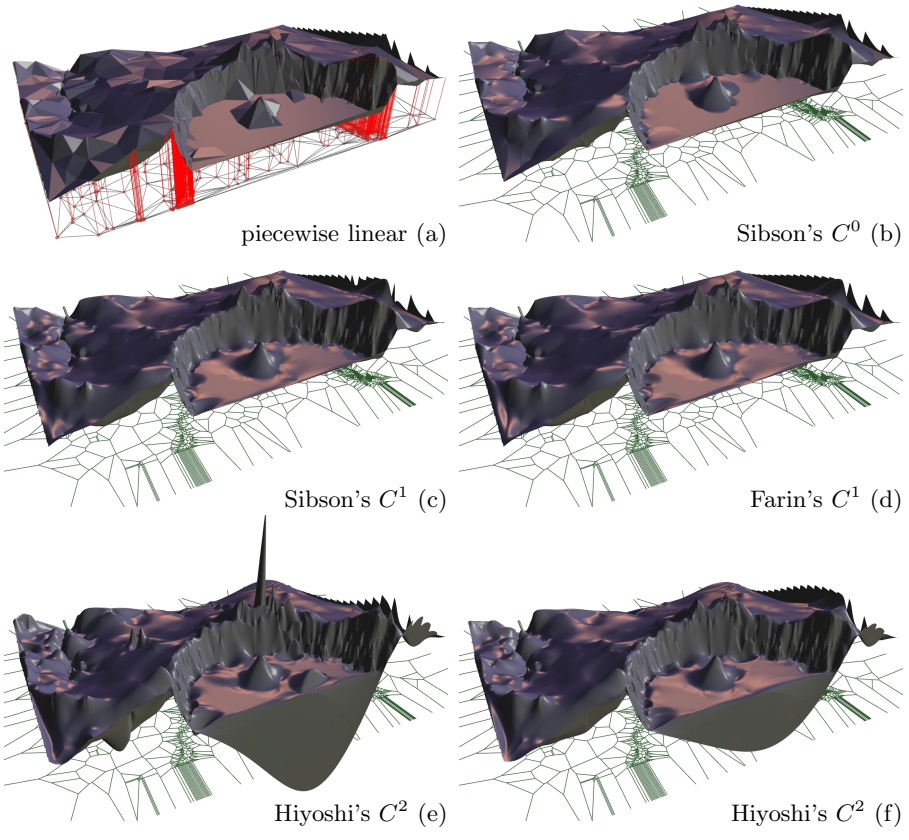


Fig. 3. The crater lake data set with 1493 points is approx. 1% of the original data set. (a) Delaunay tessellation, (b) Sibson's C^0, (c) Sibson's C^1, (d) Farin's C^1, (e),(f) Hiyoshi's C^2 with G,H estimated based on equation (10) and (11).

Since in our experiments, Sibson's and Farin's interpolants show no noticeable difference, the results for the Cosine data set only show Farin's interpolant.

The sample data distribution for the Cosine data set is shown in Figure 2(b). The difference between Farin's interpolant with exact derivatives and with G estimated as in equation (9) is clearly visible in Figure 4(a,b). The sample density in the lower left corner is so low that the gradient could not be reliably estimated, yielding a badly behaved interpolant. In Figure 4(c) derivatives from equation (10) again appear very sensitive and result in Hiyoshi's interpolant being wriggly. Figure 4(d), in contrast, shows Hiyoshi's interpolant with the stabilized method from equation (11). Note how well Hiyoshi's interpolant performs on exact derivative information in Figure 4(e). Sibson's C^0 interpolant in Figure 4(f) is unable to reproduce the smooth underlying function, but it does not oscillate at all.

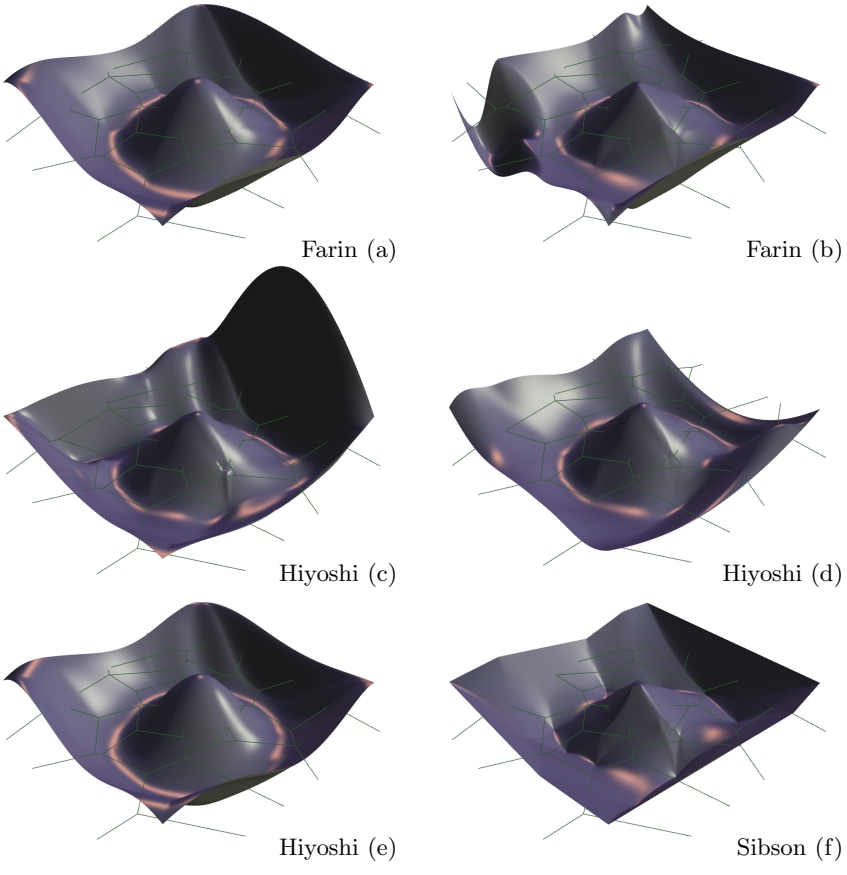


Fig. 4. (a), (b) Farin's interpolant with exact and estimated derivatives. (c), (d) Hiyoshi's interpolant with G, H estimated using (10) resp. (11), (e) Hiyoshi's interpolant with exact derivatives, (f) Sibson's C^0 interpolant.

It must be stressed that although for Sibson's, Farin's and Hiyoshi's interpolants, C^1 and C^2 continuity are guaranteed by construction, the quality of the resulting elevation map very much depends on the given derivative information. With exact derivative information as in the Cosine data set, all interpolants produce results close to the input data, with Hiyoshi's method performing best. Without derivative information available, the outcome differs depending on the nature of the represented data and the method used for derivative estimation.

Especially for data stemming from heterogeneous sources like terrains, there seems to be no best approach, but rather the demand for interpolants with adaptive smoothness. Furthermore, since most scattered data comes without derivative information, more sophisticated estimation of the latter could improve the applicability of the interpolants in practice.

Acknowledgements

This work was supported by the German Science Foundation (DFG grant number GRK 1131) as part of the International Graduate School (IRTG) on "Visualization of Large and Unstructured Data Sets."

References

1. Sibson, R.: A vector identity for the Dirichlet tessellation. Mathematical Proceedings of Cambridge Philosophical Society **87** (1980) 151–155
2. Christ, N.H., Friedberg, R., Lee, T.D.: Weights of links and plaquettes in a random lattice. Nuclear Physics B **210(3)** (1982) 337–346
3. Belikov, V., Ivanov, V., Kontorovich, V., Korytnik, S., Semenov, A.: The non-Sibsonian interpolation: A new method of interpolation of the values of a function on an arbitrary set of points. Comp. Math. and Math. Physics **37** (1997) 9–15
4. Sugihara, K.: Surface interpolation based on new local coordinates. Computer Aided Design **13** (1999) 51–58
5. Hiyoshi, H., Sugihara, K.: Voronoi-based interpolation with higher continuity. In: Symposium on Computational Geometry. (2000) 242–250
6. Hiyoshi, H.: Stable computation of natural neighbor interpolation. In: Proc. of the 2^{nd} Int. Sym. on Voronoi Diagrams in Science and Engineering. (2005) 325–333
7. Sibson, R.: A brief description of natural neighbor interpolation. Interpreting Multivariate Data (1981) 21–36
8. Farin, G.: Surfaces over Dirichlet tessellations. Computer Aided Geometric Design **7** (1990) 281–292
9. Hiyoshi, H., Sugihara, K.: Improving the global continuity of the natural neighbor interpolation. In: ICCSA (3). (2004) 71–80
10. Gross, L., Farin, G.E.: A transfinite form of Sibson's interpolant. Discrete Applied Mathematics **93** (1999) 33–50
11. Hiyoshi, H., Sugihara, K.: An interpolant based on line segment Voronoi diagrams. In: JCDCG. (1998) 119–128
12. Anton, F., Mioc, D., Gold, C.: Line Voronoi diagram based interpolation and application to digital terrain modelling. In: ISPRS - XXth Congress, Vol.2, International Society for Photogrammetry and Remote Sensing (2004)
13. Boissonat, J.D., Flötotto, J.: A coordinate system associated with points scattered on a surface. Computer-Aided Design **36** (2004) 161–174
14. Fan, Q., Efrat, A., Koltun, V., Krishnan, S., Venkatasubramanian, S.: Hardware-assisted natural neighbor interpolation. In: Proc. 7th Workshop on Algorithm Engineering and Experiments (ALENEX). (2005)
15. Park, S., Linsen, L., Kreylos, O., Owens, J., Hamann, B.: Discrete sibson interpolation. In: IEEE Trans. on Vis. and Comp. Graphics 12. Volume 2. (2006) 243–253
16. Lodha, S.K., Franke, R.: Scattered data techniques for surfaces. In: Proc. of Dagstuhl Conf. on Sci. Vis., IEEE Comp. Soc. Press (1999) 182–222
17. Wendland, H.: Scattered Data Approximation. Volume 17 of Cambridge Monographs on Appl. and Comp. Math. Cambridge University Press (2004)
18. Aurenhammer, F.: Voronoi diagrams - a survey of a fundamental geometric data structure. ACM Computing surveys **23** (1991) 345–405
19. Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial Tessellations: Concepts and applications of Voronoi diagrams. Wiley series in probability and statistics. John Wiley & Sons Ltd (2000)

Iris Recognition Using a Low Level of Details

Jaemin Kim[1], Seongwon Cho[1], Daewhan Kim[1], and Sun-Tae Chung[2]

[1] School of Electronics and Electrical Engineering, Hongik University,
72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea
swcho@hongik.ac.kr
[2] School of Electronic Engineering, Soongsil University
cst@csu.ac.kr

Abstract. This paper describes a new iris recognition algorithm, which uses a low level of details. Combining statistical classification and elastic boundary fitting, the iris is first localized. Then, the localized iris image is down-sampled by a factor of m, and filtered by a modified Laplacian kernel. Since the output of the Laplacian operator is sensitive to a small shift of the full-resolution iris image, the outputs of the Laplacian operator are computed for all space-shifts. The quantized output with maximum entropy is selected as the final feature representation. Experimentally we showed that the proposed method produces superb performance in iris segmentation and recognition.

Index Terms: iris segmentation, iris recognition, shift-invariant, multiscale Laplacian kernel.

1 Introduction

Human iris is visible through the clean cornea as the colored disc inside the eye. It lies between the cornea and the crystalline lens, and is attached to the eye's ciliary body and open to the pupil. The iris is divided into two basic regions: the central pupillary zone and the surrounding ciliary zone. The pupillary zone contains the sphincter muscle, and the ciliary zone contains the dilator muscle. The collarette, which forms the border between two areas and contains the minor arterial circle, appears as a zigzag circumferential ridge.

As the amount of light entering the eye diminishes in the dark, the iris dilator muscle runs radially through the iris and pulls away from the center. When too much light enters the eye, the iris sphincter muscle, which encircles the pupil, pulls toward the center. These complex muscle movements cause the pupillary and ciliary zones to shrink or to expand and cause the zigzag pattern of the collarette to vary.

Automated iris recognition is very attractive because human iris patterns are highly distinctive to an individual and the image of an eye can be taken at a distance [1]. Various iris recognition methods have been proposed for automatic personal identification and verification. Daugman [2] first presented a prototype system for iris recognition. For the feature representation, it makes use of a decomposition derived from application of a two-dimensional Gabor filter to the iris image pattern. Quantized local phase angles yield the final representation. The similarity measure for feature classification is the Hamming distance between the acquired and data base representations. It reported excellent performance on a diverse database.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 196 – 204, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Wildes et al. [3] presented another iris recognition system. It constructs a Laplacian pyramid by iteratively applying a Gaussian lowpass filter and decimation operator to the iris image. Quantized differences between a level and its next lower resolution level yield the final representation. The similarity measure is the normalized correlation between the acquired and data base representations. It reported as good performance as the system of Daugman.

Both systems of Daugman and Wildes employ carefully designed image acquisition devices to get equal high quality iris images [3,4,5]. The devices minimize the deformation of the iris pattern due to the illumination variation, and acquire the sharp and glare-free iris images under fixed illumination. However, these demands are not easily satisfied in many field applications.

Cho et al. [6] presented a method using a wavelet sub-image at a low-resolution pyramid level. The neural network is used to classify the extracted feature. Lim et al. [7] presented the same method later. However, the subband is sensitive to a small shift of a full-resolution iris image in the space domain [8,9,10]. Zhu et al. [4] presented a shift-invariant method, which decomposes the iris pattern into multiple bands using a two-dimensional Gabor transform or a wavelet transform. To make the feature representation shift-invariant, the means and variances of all bands are used for the feature representation. However, the means and variances do not contain the information on the shape of the iris pattern. This restriction limits the performance.

Mallat built a translation-invariant representation from zero-crossings of the full-density wavelet subbands [9]. Boles et al. [11] decompose one-dimensional intensity signals computed on circles in the iris and use zero-crossings of the decomposed signals for the feature representation. This method is reported to be robust to the additive white Gaussian noise. However, the number of zero-crossings can differ among iris image samples of an identical user due to the noise. This method was improved by Roche et al. [5], who assume that if two samples belong to the identical iris, most zero-crossings of one sample can find their corresponding zero-crossings of the other sample in their neighborhood. Based on this assumption, the similarity measure is a binary Hamming distance between two sets of zero-crossing points.

Iris images are corrupted by various noises. Since a high level of detail of the iris pattern is sensitive to the noise, we make use of a low level of detail for the feature representation. For the detection of a low level of detail, the localized iris image is down-sampled by a factor of m, and filtered by a modified Laplacian kernel. Since the low-resolution output of the Laplacian operator is sensitive to the space-shift of the full-resolution iris image, we compute the outputs of the Laplacian operator for all space-shifts. The quantized output with maximum entropy is selected as the final feature representation. An appropriate formulation of similarity measure is defined for classification. This paper also addresses how to accurately detect iris boundaries for improved iris recognition.

2 Iris Localization

Iris recognition methods require accurate iris localization for successful processing because the iris is a small part of an acquired image. The visual surface of the iris lies outside the pupil and inside the limbus, the border between the sclera and the iris.

Sometimes, the eyelids cover parts of the iris. The iris boundaries have been parametrically modeled and subsequently detected by circles or parabolic arcs [2,3,4,12]. This parametric shape modeling is reported to produce better performance than the Canny edge detector and binarization because it is less sensitive to thick eyelashes and the shadow of the upper eyelid. If the same boundary location is always detected, it does not matter whether the parametric model does not fit to the boundary or not. However, as the pupil dilates or constricts, the shape of a boundary, the pupil contour, becomes different due to the complex movement of the dilator and sphincter muscles. This is illustrated in Fig. 1. To always detect the same boundary location, we describe a new robust iris localization method, which combines statistical classification and elastic boundary fitting.

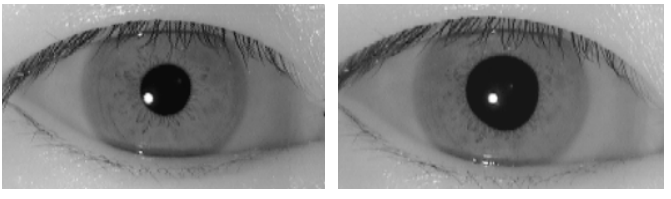


Fig. 1. Pupil size and shape vs. illumination $[(a) \quad (b)]$: (a) in dark illumination, (b) in bright illumination

2.1 Eye Segmentation

A Gaussian mixture model is highly effective for object segmentation when objects have different intensity distributions [13,14]. In the image acquisition system using infrared LEDs (light emitting diode), the captured images have an intensity distribution typical of that shown in Fig. 1. A reasonable model distribution is a mixture of three Gaussian distribution components: dark, intermediate and light. The dark component contains the portion of the image corresponding to the pupil. The intermediate component contains the portion corresponding to the iris.

One common method for solving Gaussian mixture models is an expectation-maximization (EM) algorithm [13,14]. EM is an effective procedure for estimating model parameters when data is incomplete. When the histogram of an image deviates significantly from typical due to glasses and non-uniform illumination, the segmentation based on the estimated parameter can result in poor performance. To alleviate this problem, we project the estimated parameters onto a predefined solution space.

The segmentation based on the Gaussian mixture model can be marred by speckle noise. Such noise can be removed by segmentation rules. For our purposes, if more than half of the pixels in the neighborhood of the pixel to be filtered belong to one of three components, the pixel is classified as the component.

2.2 Detection of Pupillary Boundary with Elastic Contour Model

Once the image is segmented into three different components (dark light, and intermediate), we seek definition of an oval shaped dark region in the central area of the image. From this region, the pupillary contour and its centroid are approximately

estimated. Since this estimated contour can contain local error due to effects such as the reflection of the light source and eyelashes, the initial contour is refined using an elastic contour model. The iris boundaries detected using an elastic contour model remains sensitive to noise such as the eyelashes and the reflection of the illumination source. To alleviate this problem, the weighting factor is decreased if a node point is near image artifacts such as the eyelashes and light source reflection. The portion corresponding to the light source reflection is covered by the bright distribution component. The reflection on the pupil area or on the iris area is easily located. The portion corresponding to the eyelashes is covered by the dark distribution component in the iris area and by the intermediate distribution in the pupil area. Once the pupil area and the iris area are detected, we can easily detect these artifacts. The detection results are shown in Fig. 4(c).

2.3 Detection of the Limbus

When infrared LEDs provide light over the eye, the limbus does not have sharp intensity transition, compared with the eyelid and eyelashes. Some eye has a vague outline at the limbus. This makes it difficult to detect the limbus using the active snake model.

The limbus is modeled by a circle, and the model parameters, the radius and the center, are detected by minimizing the objective cost, which takes the gradient of pixel intensity in the radial direction at the circle and integrate it along the arc as follow:

$$\min_{r,c} \int_{\theta_{l1}}^{\theta_{l2}} I(r,\theta) - I(r+\Delta r,\theta) d\theta + \int_{\theta_{r1}}^{\theta_{r2}} I(r,\theta) - I(r+\Delta r,\theta) d\theta. \quad (1)$$

where r, c are the radius and the center. Since the eyelids can cover parts of the limbus, $\theta_{l1}, \theta_{l2}, \theta_{r1}, \theta_{r2}$ are determined in the eye segmentation.

3 Feature Extraction

3.1 Preprocessing

Since the acquired iris image has different contrast and non-uniform illumination due to the position and angle of the light source, the image is enhanced using local histogram equalization. The variation of the illumination causes the iris to shrink or to expand. For the iris texture analysis, it is necessary to compensate the variation of the iris size. A common method is to map the disk-shaped iris to a rectangle block of a fixed size [3]. This is shown in Fig. 2.



Fig. 2. A normalized iris texture mapped from a disk-shaped iris

3.2 Shift-Invariant Multiscale Laplacian Operator

The feature representation should have information enough to classify various iris textures and be insensitive to various noises: sensor noise, iris localization error, different image contrast, the iris texture deformation due to a small amount of movement of the dilator and sphincter muscles etc. Since a high level of detail of the iris texture is sensitive to the noise, a low level of detail is used for the feature representation. In this section, we describe a multiscale Laplacian operator, which can detect a low level of signal variation in all directions. For this, the iris image is smoothed by a convolution with a Gaussian function, down-sampled by a factor of m_1-by-m_2, and filtered with a 3-by-3 Laplacian kernel. However, the output of the multiscale Laplacian operator is sensitive to a small shift of the full-resolution iris image. The shift-sensitive problem is reported in many papers on wavelet transform, and various solutions are proposed [8,9,10]. As a similar approach to the M-band method of Liang et al. [8], we compute the outputs of the Laplacian operator for all space-shifts of the full-resolution image. Following the notation in Liang et al., the outputs of Laplacian operator for all space shifts are expressed as

$$l_{0,0}(n_1,n_2)=\sum_{k_1}\sum_{k_2} g(k_1)g(k_2)f(k_1+m_1\cdot n_1,k_2+m_2\cdot n_2) \tag{2a}$$

$$l_{i,j}(n_1,n_2)=\sum_{k_1}\sum_{k_2} g(k_1)g(k_2)f(k_1+m_1\cdot n_1+i,k_2+m_2\cdot n_2+j) \tag{2b}$$

$$d_{0,0}(n_1,n_2)=\sum_{k_1}\sum_{k_2} c(k_1,k_2)l_{0,0}(k_1+n_1,k_2+n_2) \tag{2c}$$

$$d_{i,j}(n_1,n_2)=\sum_{k_1}\sum_{k_2} c(k_1,k_2)l_{i,j}(k_1+n_1,k_2+n_2) \tag{2d}$$

where $g(\cdot)$ is the Gaussian smoothing function, m_1, m_2 are the down-sampling factors, $(i=1,\cdots,m_1-1,\ j=1,\cdots,m_2-1)$ are the space shifts, $l_{i,j}(n_1,n_2)$ is a down-sampled iris image for the space shift (i,j), $c(k_1,k_2)$ is a 3-by-3 Laplacian kernel, $d_{i,i}(n_1,n_2)$ is the output of the Laplacian operator for the space shift (i,j). The operator of down-sampling by (m_1,m_2) is a linear operator of period (m_1,m_2), which means that if the input is shifted by $(a\cdot m_1,b\cdot m_2)$, the output of the Laplacian operator will be shifted by (a,b). Therefore, instead of calculating the output for $(N_1\times N_2)$ different shifts, we only need to the compute the output for $(m_1\times m_2)$ shifts. Once the outputs of the Laplacian operator for all shifts are computed, the output with maximum entropy is selected as the final feature representation.

When the size of a normalized iris image is 48-by-384 and the down-sampling factor is 6-by-6, the number of features is 512. The 512 features are too large for some application (i.e. a magnetic stripe card). To reduce the feature size, we quantized each feature element using Lloyd-Max method.

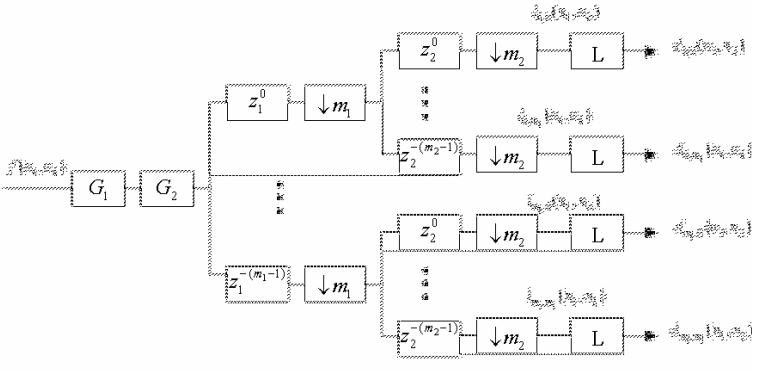


Fig. 3. Shift-invariant multiscale Laplacian Operator

5 Experimental Results

The database used for the experiments consists of 200 eye photos, 5 photos per person. The photos were taken in different hour during four weeks. The photos were taken in different illumination condition: little bright, normal, and little dark. The light source is composed of direct infrared LED (light emitting diode) and indirect normal electric lamps. We control illumination using indirect electric lamps. The photos have different contrast: sharp and smooth (not blurred). In eye photos, the iris is not covered or slightly covered by the eyelids and the eyelashes.

A typical intensity distribution is modeled as a mixture of three Gaussian distribution components and is shown in Fig. 4(a). The segmentation and the eyelash detection are shown in Fig. 4(b). The segmented pupil boundary has some error due to the eyelashes. This error is corrected by using an elastic contour model. This is illustrated with a white dotted line in Fig. 4(c). The pupillary contour of Fig. 4(d) shows that the pupil shape differs from the circle. The detected limbic boundary is shown with a black line in Fig. 4(c) and (d).

Generally, EM is known to be sensitive to initial conditions. In our experiments, with a single condition, successfully segmentation is achieved among various irises captured with our image acquisition system: eyes with thick eyelashes covering the pupil, eyes with a very dark shadow of the upper eyelid, eyes with light reflection on the iris region, an eye damaged by accident. The proposed segmentation method yields little iris localization error, which improves the iris classification performance.

Finally, we compared the presented iris recognition method with a zero-crossing method proposed by Martin-Roche et al. [5] and a Laplacian pyramid decomposition method proposed by Wildes et al. [3]. In the implementation of the zero-crossing method, we experimentally found the best threshold value for computing the

Hamming distance. In the implementation of the Laplacian pyramid method, we experimentally found the best combination of levels that results in the best performance because the high-resolution levels are sensitive to noises.

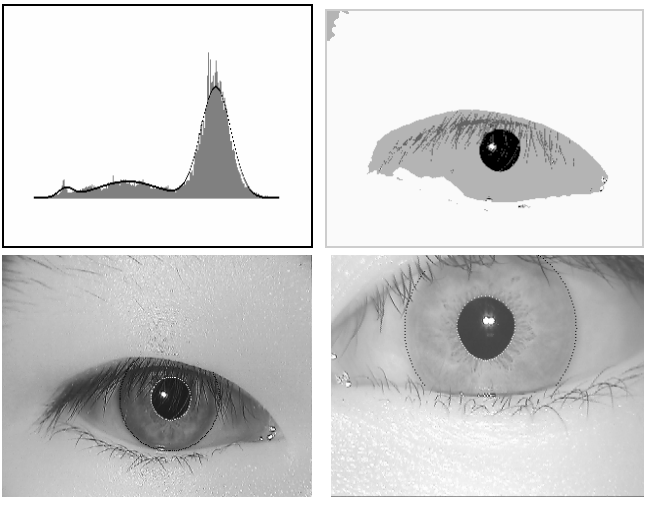


Fig. 4. Segmentation results $\begin{bmatrix} (a) & (b) \\ (c) & (d) \end{bmatrix}$.(a) A typical intensity distribution and its estimation based on a Gaussian mixture model with EM, (b) The pupil, part of the eye covering the iris, the eyelashes, (c),(d) The pupillary contour is indicated by a white dotted line and the limbic boundary is indicated by a black line.

Fig. 5 shows the distributions of similarity degrees between feature vectors. The distributions of similarity degrees between sample pairs of the same class are illustrated with a black bar. The distributions of similarity degrees between sample pairs of different classes are illustrated with a gray bar. We computed 40000 similarity degrees between different pairs. For the illustration, similarity degrees of four methods are normalized to [0,1]. For the successful verification, the similarity degrees between sample pairs of the same class should be overlapped as less as possible with those between sample pairs of different classes. Fig. 5(a) shows the results of the zero-crossing method. Fig. 5(b) shows the results of the zero-crossing method. Fig. 5(c) and (d) are the results of the presented methods using shift-variant/shift-invariant multiscale Laplacian operator, respectively. The method using the shift-invariant multiscale Laplacian operator yields the least overlap.

Table 1 shows FARs (false accept rate) and FRRs (false rejection rate) of four different methods for iris verification. The proposed shift-invariant method yields far less FAR and FRR. Table 2 shows the classification-fail rates of four different methods. All methods use the 1-NN classification. The proposed shift-invariant method yields the smallest classification-fail rate. The experimental results show that the presented method outperformed conventional methods.

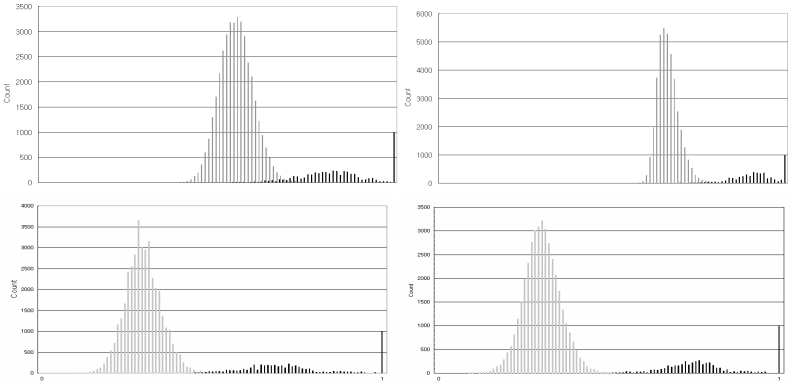


Fig. 5. Similarity degrees between samples of the same class and similarity degrees between samples of different classes $\begin{bmatrix}(a) & (b)\\ (c) & (d)\end{bmatrix}$: (a) zero-crossing method, (b) Laplacian pyramid method, (c) multiscale Laplacian method, (d) shift-invariant multiscale Laplacian method

Table 1. FARs and FRRs of four different methods $\begin{bmatrix}(a) & (b)\\ (c) & (d)\end{bmatrix}$: (a) zero-crossing method, (b) Laplacian pyramid method, (c) multiscale Laplacian method, (d) shift-invariant multiscale Laplacian method

Threshold	FAR(%)	FRR(%)
0.64	5.7	2.1
0.65	3.8	2.7
0.66	2.4	3.6
0.67	1.5	4.1
0.68	0.9	5.2

Threshold	FAR(%)	FRR(%)
0.74	2.0	0.8
0.75	1.22	1.1
0.76	0.68	1.7
0.77	0.37	2.2
0.78	0.16	3.1

Threshold	FAR(%)	FRR(%)
0.44	1.27	0.0
0.45	0.89	0.0
0.46	0.55	0.0
0.47	0.37	0.2
0.48	0.28	0.4

Threshold	FAR(%)	FRR(%)
0.49	0.203	0.0
0.50	0.144	0.0
0.51	0.082	0.2
0.52	0.046	0.1
0.53	0.020	0.2

Table 2. Classification results of four methods

Method	Classification fail
Zero-crossing	3.00%
Laplacian Pyramid	0.58%
multiscale Laplacian	0.28%
shift-invariant multiscal Laplacian	0.04%

6 Conclusion

We have presented a new method for iris recognition, which uses a low level of details. Combining statistical classification and elastic boundary fitting, we first localized the iris. Then, we used a shift-invariant Laplacian filter for extracting features for recognition. Finally, we quantized the extracted feature without much loss of recognition performance. Experimentally we showed that the proposed method produces superb performance in iris segmentation and recognition.

Acknowledgement

This work was supported by 2006 Hongik University Research Fund, the Soongsil University Research Fund, and BK21.

References

1. F. H. Adler, Physiology of the Eye. St. Louis, MO: Mosby, 1965.
2. J. Daugman, "High Confidence Visual Recognition of Person by a Test of Statistical Independence," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, pp. 1148-1161, 1993.
3. R. P. Wildes, "Iris Recognition: An Emerging Biometric Technology," Proceedings of the IEEE, vol. 85, pp.1348-1363, 1997.
4. Y. Zhu, T. Tan and Y. Wang, "Biometric Person Identification based on Iris Pattern," ICPR2000, pp.805-808, 2000.
5. D. de Martin-Roche, C. Sanchez-Avila, and R. Sanchez-Reillo, "Iris Recognition for Biometric Identification using Dyadic Wavelet Transform Zero-Crossing," Security Technology, 2001 IEEE 35th International Carnahan Conference, pp. 272-277, 2001.
6. S. Cho and H. I. Sung, "Wavelet transform and Competitive Learning Neural Network with Multi-dimensional Wiener Decision Strategy," Proceeding of KFIS Conference, vol. 8, No. 2, pp. 341-345, 1998.
7. S. Lim, K. Lee, O. Byeon, and T. Kim, " Efficient iris recognition through improved of feature vector and classifier," ETRI Journal, vol. 23, No. 2, pp 61-70, June. 2001.
8. J. Liang and T. W. Parks, "Image coding using translation invariant wavelet transforms with symmetric extensions," IEEE Trans. Image Processing, vol. 7, No.2 1998.
9. S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 14, pp 710-732, July 1992.
10. E. P. Simoncelli, W. T. Freeman, E.H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," IEEE Trans. Inform. Theory, vol. 38, pp 587-607, Mar. 1992.
11. W. W. Boles, and B. Boashah, "A Human Identification Technique Using Images of the Iris and Wavelet Transform," IEEE Trans. on Signal Processing, vol. 46, pp. 1185-1188, 1998.
12. S. Masahiko and K. Yuji, **"**Iris Region Extraction and Individual Identifying Device," Japan Patent, JP2000189403A2, July 11, 2000.
13. J. Li and A. Barron, Mixture Density Estimation, in Solla, Leen, and Mueller (eds.), Advances in Neural Information Processing Systems 12, The MIT Press, 2000.
14. A. Lundervold and P. Storvik, **"**Segmentation of brain parenchyma and cerebrospinal fluid in multispectral magnetic resonance images," IEEE Trans. Medical Imaging, vol.14, pp. 339-349, Jun. 1995.

Dynamic Reconstruction of Complex Planar Objects on Irregular Isothetic Grids

Antoine Vacavant[1], David Coeurjolly[2] and Laure Tougne[1]

[1] LIRIS - UMR 5205, Université Lumière Lyon 2
5, avenue Pierre Mendès-France
69676 Bron cedex, France
[2] LIRIS - UMR 5205, Université Claude Bernard Lyon 1
43, boulevard du 11 novembre 1918
69622 Villeurbanne cedex, France
{antoine.vacavant, david.coeurjolly, laure.tougne}@liris.cnrs.fr

Abstract. The vectorization of discrete regular images has been widely developed in many image processing and synthesis applications, where images are considered as a regular static data. Regardless of final application, we have proposed in [14] a reconstruction algorithm of planar graphical elements on irregular isothetic grids. In this paper, we present a dynamic version of this algorithm to control the reconstruction. Indeed, we handle local refinements to update efficiently our complete shape representation. We also illustrate an application of our contribution for interactive approximation of implicit curves by lines, controlling the topology of the reconstruction.

1 Introduction

The representation by lines of graphical elements is an important task in many image processing and synthesis applications [4, 5, 8, 12, 13]. In general, images are digitized on regular grids: all the pixels have the same size, and their position can be easily indexed. However, it is now common to successively divide an image into subimages, as in *quadtree* decomposition [11], to represent a part of an image in a more compact and adapted manner. An other example: to approximate an implicit curve, the algorithms designed in interval arithmetic build an irregular tiling of the plane [6, 9, 13]. We have presented in [14] a system to represent the elements contained in an *irregular isothetic grid* ($\mathbb{I}$-grid for short) [1]: the pixels are defined by variable sizes and positions, and may be determined by subdivision rules. We describe the topology of the elements in a two-dimensional (2-D) image by their associated Reeb graph [10], then we geometrically represent them by a simple polygonal structure. By using an irregular image representation, we naturally produce less line segments than in the regular discrete applications.

In this article, we propose to efficiently modify this representation to handle the dynamic construction of an $\mathbb{I}-$grid. Moreover, we control the topology of the recognized irregular objects. Indeed, we may build an irregular grid by successively applying *local refinements by inclusion* that respects a grid model. If we consider an irregular grid $\mathbb{I}$ and a cell $P \in \mathbb{I}$, such a refinement performed on P

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 205–214, 2006.
© Springer-Verlag Berlin Heidelberg 2006

is a set of subcells $S(P) \subseteq P$ (inclusion rule), obtained by the process S. For example, in the quadtree decomposition, a cell $P \in \mathbb{I}$ is divided into four subcells if the region of the 2-D image bounded by P is not homogeneous. In this case, S represents the operation of division in four subcells, and we consider the 2-D image data to check the homogeneity criterion. To completely define such a grid model, the choice of the refined cell P can be guided by two main approaches: (1) an irregular grid can be successively refined by inclusion to respect a subdivision model, as in the quadtree decomposition where all the heterogeneous cells are subdivided; (2) we can manually choose the cells to refine. This interactive way of construction is interesting for implicit curve approximation by interval arithmetic, where an user may locally refine the approximation. In this kind of implicit curve approximation, the digitization algorithms consist in tiling the plane with isothetic rectangles so that the real function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ belongs to each rectangle. For now, we focus on the interactive refinement of those *data driven grids* [1]. From a coarse description of f, we choose one or more rectangles in $\mathbb{I}$, where a finer description of f is performed. We proove that we can handle local refinements by inclusion in our reconstruction of an irregular object. By an efficient and simple algorithm, we update its topological representation by the Reeb graph, then we deduce a new polygonal approximation of its shape.

We first introduce the concepts of $k-$arcs by recalling some definitions, then we present the extended supercover model on an $\mathbb{I}-$grid. We also recall the invertible reconstruction of $k-$arcs described in [2]. In the third part, we give details about our contribution: we shortly describe the topological and geometrical reconstruction of a complex object based on the Reeb graph [10] that we have proposed in [14]. Then, we present our algorithm designed to update this structure considering one or several local refinements. We also present an application of our system to the interactive approximation of an implicit curve by lines. We finally discuss about possible extensions of our method.

2 Preliminaries

We first define an irregular isothetic grid, denoted $\mathbb{I}$, as a tiling of the plane with isothetic rectangles. Each rectangle P (also called *cell*) of $\mathbb{I}$ is defined by its center $(x_P, y_P) \in \mathbb{R}^2$ and a size $(l_P^x, l_P^y) \in \mathbb{R}^2$. In our framework, adjacency relation is an important feature that we depict through the following definitions.

Definition 1 ($ve-$adjacency and $e-$adjacency). *Let P and Q be two cells. P and Q are $ve-$adjacent (vertex and edge adjacent) if :*

$$or \begin{cases} |x_P - x_Q| = \frac{l_P^x + l_Q^x}{2} \text{ and } |y_P - y_Q| \leq \frac{l_P^y + l_Q^y}{2} \\ |y_P - y_Q| = \frac{l_P^y + l_Q^y}{2} \text{ and } |x_P - x_Q| \leq \frac{l_P^x + l_Q^x}{2} \end{cases}$$

P and Q are $e-$adjacent (edge adjacent) if we consider an exclusive "or" and strict inequalities in the above $ve-$adjacency definition. k will now represent either e or a ve adjacency in the following definitions.

Definition 2 ($k-$arc). *Let $\mathcal{E}$ be a set of cells, $\mathcal{E}$ is a $k-$arc if and only if for each element of $\mathcal{E} = \{P_i, i \in \{1, ..., n\}\})$, P_i has exactly two $k-$adjacent cells, except P_1 and P_n which are called extremities of the $k-$arc.*

We now consider the extension of the supercover model from [3] on irregular isothetic grids [1] to digitize Euclidean objects on $\mathbb{I}$.

Definition 3 (Supercover on irregular isothetic grids). *Let F be an Euclidean object in $\mathbb{R}^2$. The supercover $\mathbb{S}(F)$ is defined on an irregular isothetic grid $\mathbb{I}$ by :*

$$\begin{aligned}\mathbb{S}(F) &= \left\{P \in \mathbb{I} \mid \mathbb{B}^\infty(P) \cap F \neq \emptyset\right\} \\ &= \left\{P \in \mathbb{I} \mid \exists (x, y) \in F, |x_P - x| \leq \frac{l_P^x}{2} \text{ and } |y_P - y| \leq \frac{l_P^y}{2}\right\}\end{aligned}$$

where $\mathbb{B}^\infty(P)$ is the rectangle centered in (x_P, y_P) of size (l_P^x, l_P^y) (if $l_P^x = l_P^y$, $\mathbb{B}^\infty(P)$ is the ball centered in (x_P, y_P) of size l_P^x for the L_∞ norm).

We now present the $k-$arc reconstruction algorithm we use in our complex object geometrical representation phase (Section 3.1). Moreover, this approach respects the supercover model we have just presented. The algorithm proposed in [2] to decompose a curve into segments is first based on the following definition of an irregular digital line.

Definition 4 (Irregular isothetic digital straight line). *Let S be a set of cells in $\mathbb{I}$, S is called a piece of irregular digital straight line (IDSL for short) iff there exists an Euclidean straight line l such that :*

$$S \subseteq \mathbb{S}(l)$$

In other words, S is a piece of IDSL iff there exists l such that for all $P \in S$, $\mathbb{B}^\infty(P) \cap l \neq \emptyset$.

The Figure 1 illustrates the progressive construction of cones in a $k-$arc, and the resulting segmentation into lines.

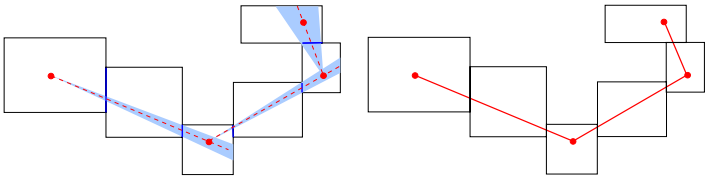


Fig. 1. An example of the progressive construction of cones in a $k-$arc ($left$), and the reconstruction into segments we obtain *(right)*

3 Dynamic Representation of Complex 2-D Objects on Irregular Isothetic Grids

3.1 Topological and Geometrical Reconstruction of a Complex 2-D Object

To represent the shape of a $k-$object $\mathcal{E}$, we have chosen in [14] an incremental directional approach to build its associated Reeb graph G, as in continuous space. It is an interesting structure introduced by G. Reeb [10] based on the Morse theory [7]. The Reeb graph $G = (V, E)$ is associated to a *height function* f defined on $\mathcal{E}$, and nodes of G represent the critical points of f. These critical nodes represent the important elements of the topology of $\mathcal{E}$. More exactly, we denote them *begin* (b), *merge* (m), *split* (s) and *end* (e) nodes (see Algorithm 1). To handle the orientation of f, we choose a direction to treat the cells of $\mathcal{E}$. Without loss of generality, we can suppose that we choose the left-to-right orientation along X axis, *i.e.* the height function f is defined along X axis. Moreover, we want to represent an edge between two nodes by a $k-$arc, to have a minimal information in the topological representation of $\mathcal{E}$. To respect this feature, we use a simple procedure to update and recode a $k-$arc with one or more adjacent cells (see [14] for more details). In Algorithm 1, we present the main stages of our reconstruction algorithm of complex irregular objects. As a $k-$arc A is associated to an edge $n_1 - n_2 \in E, n_1, n_2 \in \{b, e, m, s\}$, we note in this algorithm $inf(A) = n_1$ and $sup(A) = n_2$. We also abusively say that a $k-$arc A and a cell P are $k-$adjacent if there exists a cell Q in A such that P and Q are $k-$adjacent. We finally note $\mathcal{E} = \{P_i\}_{i=0,\ldots,n}$ a given 2-D set of cells.

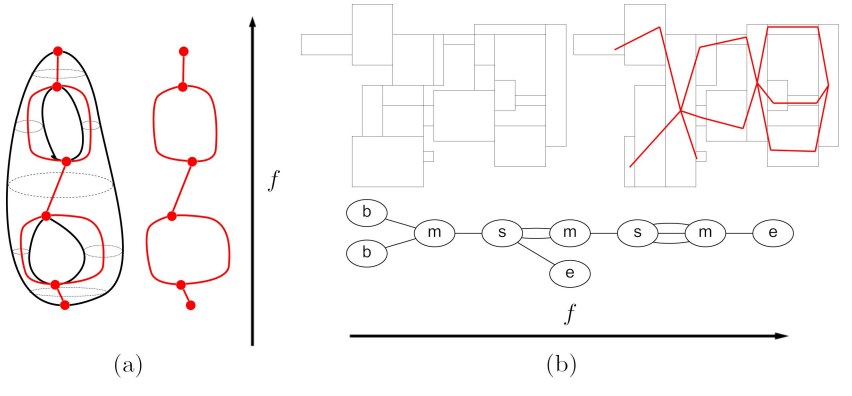


Fig. 2. *(a)* : an example of the Reeb graph G of a continuous object $\mathcal{E}$. The node of G represent the critical points of f (maxima, minima, inflection points), and an edge is a connected component of $\mathcal{E}$ between two critical points. *(b)* : an example of an irregular object $\mathcal{E}$ *(left)*, the final recoded structure with $k-$arcs, the obtained polygonization *(right)* and the Reeb graph associated to the height function f defined on $\mathcal{E}$ *(bottom)*.

Indeed, our method first builds a complete topological representation of $\mathcal{E}$ with the Reeb graph G by recognizing and linking critical nodes. At the end of our

approach, we use the $k-$arc reconstruction algorithm (see section 2) to build a polygonal description of $\mathcal{E}$. We process the geometrical reconstruction of E from merge and split nodes (internal nodes of our structure) to the others. Thus, all the merge or split nodes in G are represented by a single point. Moreover, since the recognition algorithm is greedy, the possible error induced by the visibility cone approach is propagated to the extremities of $\mathcal{E}$, instead of those intersections that represent the shape of the object. The complexity in the worst case of this algorithm is $\mathcal{O}(m \times n)$: for all P_i, we test all the m $k-$arcs A_j to find which cell P_i is adjacent with A_j. In Figure 2, we present the Reeb graph and the polygonization obtained by our approach on a simple example.

Algorithm 1. Reconstruction of an object $\mathcal{E} \in \mathbb{I}$

Input : $\mathcal{E} = \{P_i\}_{i=1,...,n}$, a set of n 2-D cells in $\mathbb{I}$.
Output : $G = (V, E)$, the Reeb graph associated to $\mathcal{E}$
L, the reconstruction by lines of $\mathcal{E}$
$A = \{A_i\}_{i=1,...,m}$, the set of m $k-$arcs recoding $\mathcal{E}$

Add an edge $b - e$ in G
Create the arc A_1 with $inf(A_1) = b$ and $sup(A_1) = e$ {m = 1}
for all cells P_i, $2 \leq i \leq n$ **do**
  for all arcs A_j, $1 \leq j \leq m$ **do**
    if one cell P_i is adjacent with one arc A_j **then**
      Update A_j with P_i
    end if
    if p cells $P_i, P_{i+1}, ..., P_{i+p}$ are adjacent with A_j **then**
      Update A_j with $P_i, P_{i+1}, ..., P_{i+p}$
      Add a split node s in G $\{deg(s) = p + 1\}$
      $sup(A_j) = s$
      Link p new arcs $A_{m+1} = P_i, ..., A_{m+p} = P_{i+p}$ with A_j
      $inf(A_{m+1}) = ... = inf(A_{m+p}) = s$
    end if
    if one cell P_i is adjacent with p $k-$arcs $A_j, A_{j+1}, ..., A_{j+p}$ **then**
      Update $A_j, A_{j+1}, ..., A_{j+p}$ with P_i
      Add a merge node m in G $\{deg(m) = p + 1\}$
      $sup(A_j) = ... = sup(A_{j+p}) = m$
      Link a new arc $A_{m+1} = P_i$ with $A_j, A_{j+1}, ..., A_{j+p}$
      $inf(A_{m+1}) = m$
    end if
    if a cell P_i is adjacent with any arc A_j **then**
      Add an edge $b - e$ in G
      Create an arc A_{m+1} with $inf(A_{m+1}) = b$ and $sup(A_{m+1}) = e$
    end if
  end for
end for
Perform the polygonization over the set of arcs $\{A_j\}_{j=1,...,m}$ considering G

3.2 Dynamic Reconstruction of Complex 2-D Objects

We now want to update our reconstruction of complex objects, considering one or more local refinements by inclusion. More precisely, we have to insert the reconstruction of the set of cells defined in the local refinement into the global reconstruction. Here, we propose an efficient algorithm to locally change the Reeb graph and the polygonal approximation we compute in Algorithm 1.

Let $S_t(\Omega)$ be a set of cells obtained by a process S_t at time t on the domain $\Omega \subset \mathbb{R}^2$, *e.g.* a quadtree decomposition of level t in the image plane. Let $P \in S_t(\Omega)$ be a cell where we want to have a more precise decomposition, *e.g.* a local quadtree refinement of level $t+1$. We denote $S_{t+1}(P)$ the set of cells in the domain bounded by P obtained by the process S_{t+1}. G_t, L_t and A_t are the elements computed by Algorithm 1 (see Input of Algorithm 2). For each $k-$arc $A_i \in A_t$, its associated edge in G_t is denoted in a lower case a_i. In this section, we propose a simple algorithm to update G_t, L_t and A_t considering $S_{t+1}(P)$. First, we consider all the coarse $k-$arcs in $S_t(\Omega)$ that contain P. Then, we replace P with the smaller pixels $S_{t+1}(P)$, and perform Algorithm 1 over those pixels to have a local topological and geometrical representation of them. Then, we branch the local recognized arcs represented in $G_{t+1}(P)$, $L_{t+1}(P)$ and $A_{t+1}(P)$ (associated to $S_{t+1}(P)$) in G_t, L_t and A_t respectively.

In Figure 3, we illustrate the behaviour of our algorithm for the two cases $p = 1$, *i.e.* the cell P belongs to one $k-$arc (Algorithm 2, line 4) and $p > 1$, where P belongs to several $k-$arcs (Algorithm 2, line 2).

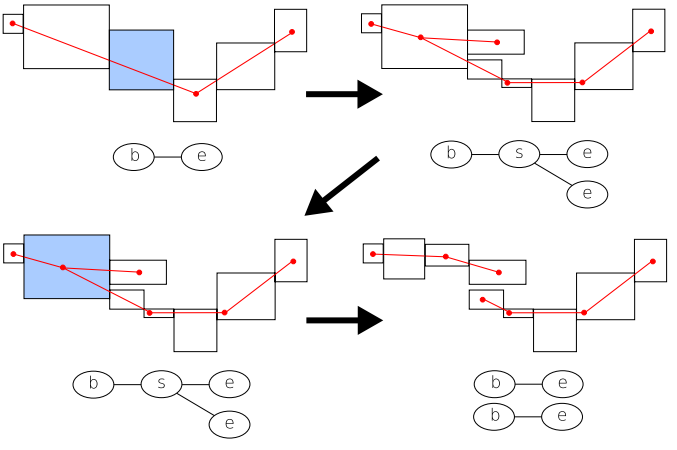


Fig. 3. Examples of results for Algorithm 2. The first reconstruction *(top-left)* contains one arc. Then, we choose the cell in dark and decompose it into two $k-$arcs *(top-right)*. This cell belongs to one arc, thus $p = 1$ and the edge $b - e$ has to be divided into two before adding the node s. The next update *(bottom)* considers three $k-$arcs, since we choose a split cell ($p > 1$). This cell and its associated node (s) are removed before inserting the new edges.

Algorithm 2. Update reconstruction of an object $\mathcal{E} \in \mathbb{I}$

Input : $S_t(\Omega)$, decomposition of $\Omega \subset \mathbb{R}^2$ by the process S_t
$S_{t+1}(P)$, the local decomposition of $P \in S_t(\Omega)$ by S_{t+1}
G_t, the Reeb graph associated to $S_t(\Omega)$
L_t, the reconstruction by lines of $S_t(\Omega)$
A_t, the set of $k-$arcs recoding $S_t(\Omega)$
Output : G_t, L_t and A_t are updated

1: Find the p $k-$arcs $A_k^t, ..., A_{k+p}^t \in A_t$ so that $P \in A_k^t, ..., A_{k+p}^t$
2: **if** $p > 1$ **then** {P represents a split or merge node in G_t}
3: Remove the node $n \in G_t$ associated to P
4: **else** {P is in one arc}
5: Split a_k^t into two edges a_k^t, a_{k+1}^t
6: **end if**
7: Remove the polylines of L_t associated to the $k-$arcs $A_k^t, ..., A_{k+p}^t$
8: Remove P from $A_k^t, ..., A_{k+p}^t$
9: Use Algorithm 1 to compute $G_{t+1}(P)$, $L_{t+1}(P)$ and $A_{t+1}(P)$ associated to $S_{t+1}(P)$

10: **for all** edges $a_i^{t+1} \in G_{t+1}(P)$ **do**
11: **for all** edges a_j^t, $j = k, ..., k+p$ **do**
12: **if** A_j^t and A_i^{t+1} are $k-$adjacent **then**
13: Branch a_i^{t+1} to a_j^t
14: Merge the $k-$arcs A_j^t and A_i^{t+1}
15: **end if**
16: **end for**
17: **end for**
18: Add all polylines from $L_{t+1}(P)$ to L_t
19: **for all** $k-$arcs A_j^t, $j = k, ..., k+p$ **do**
20: Compute $k-$arc reconstruction on A_j^t
21: **end for**

To find the p $k-$arcs required by the first operation of Algorithm 2 (line 1), we use a pointer between the cell P and a $k-$arc A_k^t such that $P \in A_k^t$. If P is inside A_k^t, it is clear that $p = 1$, and so only one $k-$arc is considered. When P stands on one of the extremities of A_k^t, we have to consider all the $k-$arcs linked with A_k^t. Thanks to its associated edge a_k^t in G, and the direct access to the edges linked of G with a_k^t, this search operation is computed in constant time $\mathcal{O}(1)$. Then, the next instructions change the Reeb graph G_t in the set of p edges $\{a_j^t\}_{j=k,k+p}$ and impact the p $k-$arcs $\{A_j^t\}_{j=k,k+p}$ found in the first operation (lines 2-17). The topology of the irregular object $\mathcal{E}$ is locally updated. The rest of the graph does not change. In the last phase of the algorithm, the polygonal reconstruction is first computed in the $k-$arcs belonging to the local refinement $A_{t+1}(P)$. Then, this reconstruction may be propagated to the p global $k-$arcs A_t, as in Figure 3 *(top)*. In the worst case, the update of the polygonal reconstruction is locally performed on $A_{t+1}(P)$ and A_t, and does

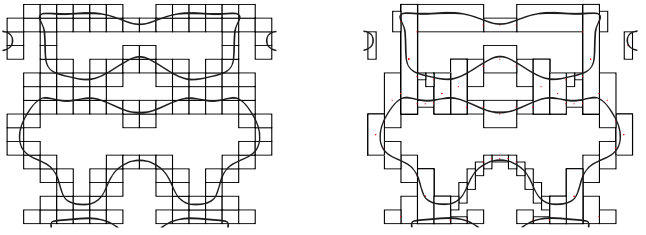
not modify the rest of the reconstruction. Finally, we can notice that our algorithm locally impact the topological and geometrical reconstruction we globally compute for $\mathcal{E}$.

To perform our algorithm on several refined cells, we first update the Reeb graph with the local graphs using the same approach (Algorithm 2, lines 1-17). Then, we perform only once the final polygonalization over the concerned $k-$arcs (see next section for a complete example).

4 Application to Interactive Approximation of Implicit Curves by Lines

In this part, we propose to apply our interactive system to approximate an implicit curve by interval arithmetic analysis. The interval arithmetic is a widely used range based model for numerical computation where each quantity x is represented by an interval $\bar{x}$ of floating point numbers. Arithmetic operations are defined so that each resulting interval $\bar{x}$ is guaranteed to contain the unknown value corresponding to the real quantity x. Briefly, an interval $\bar{x}$ is represented by $[\bar{x}.inf, \bar{x}.sup]$ and we have, for example, $\bar{x}+\bar{y} = [\bar{x}.inf+\bar{y}.inf, \bar{x}.sup+\bar{y}.sup]$ or $\bar{x}-\bar{y} = [\bar{x}.inf-\bar{y}.inf, \bar{x}.sup-\bar{y}.sup]$. Thanks to this interval definition of operations, we can build an approximation $f(\bar{x}, \bar{y})$ of a function $f : \mathbb{R}^2 \to \mathbb{R}$ by tiling the plane with an irregular isothetic grid, where each rectangle is given by $[\bar{x}.inf, \bar{x}.sup] \times [\bar{y}.inf, \bar{y}.sup]$ (Figure 4-*up* for example). Then, the polygonal representation of the computed irregular object permits to have a first geometrical analysis of the unknown underlying Euclidean curve. For example, we can approximate geometrical features (*e.g.* length) of f.

In this section, we first use an interval computing algorithm described in [13] to discretize $f = x^2+y^2+\cos(2\pi x)+\sin(2\pi y)+\sin(2\pi x^2)\cos(2\pi y^2) = 1$ through a set of cells $S_0(\Omega)$, where $\Omega = [-1.1; 1.1] \times [-1.1; 1.1]$ (Figure 4 *(top-left)*). The width δ of the final computed ranges is fixed: $\delta = 0.1$. There exists many rules to build various irregular tilings of the plane. We have chosen this simple criterion to generate a regular grid, but our algorithm handles any irregular isothetic grid. Then, we successively update the elements computed by Algorithm 1 with local refinements by inclusion. We select several cells at each time, and use Algorithm 2 as we have presented it in the previous section. The new cells contained in the refinement are deduced from the same interval computation algorithm, with a smaller range width ($\delta = 0.05$). In the figure 4, we show four times of update. The first update procedure, with the input presented in $t = 0$, does not modify the Reeb graph ($G_0 = G_1$) while the geometrical reconstruction (A_1 and L_1) is finer ($t = 1, left$). At time $t = 1$, we choose two cells that modify the Reeb graph G_1 by splitting the edges $m-m$ and $s-s$ ($right$). Finally, selecting the four cells at time $t = 2$ permits to create two new edges in G_3 ($t = 3$).



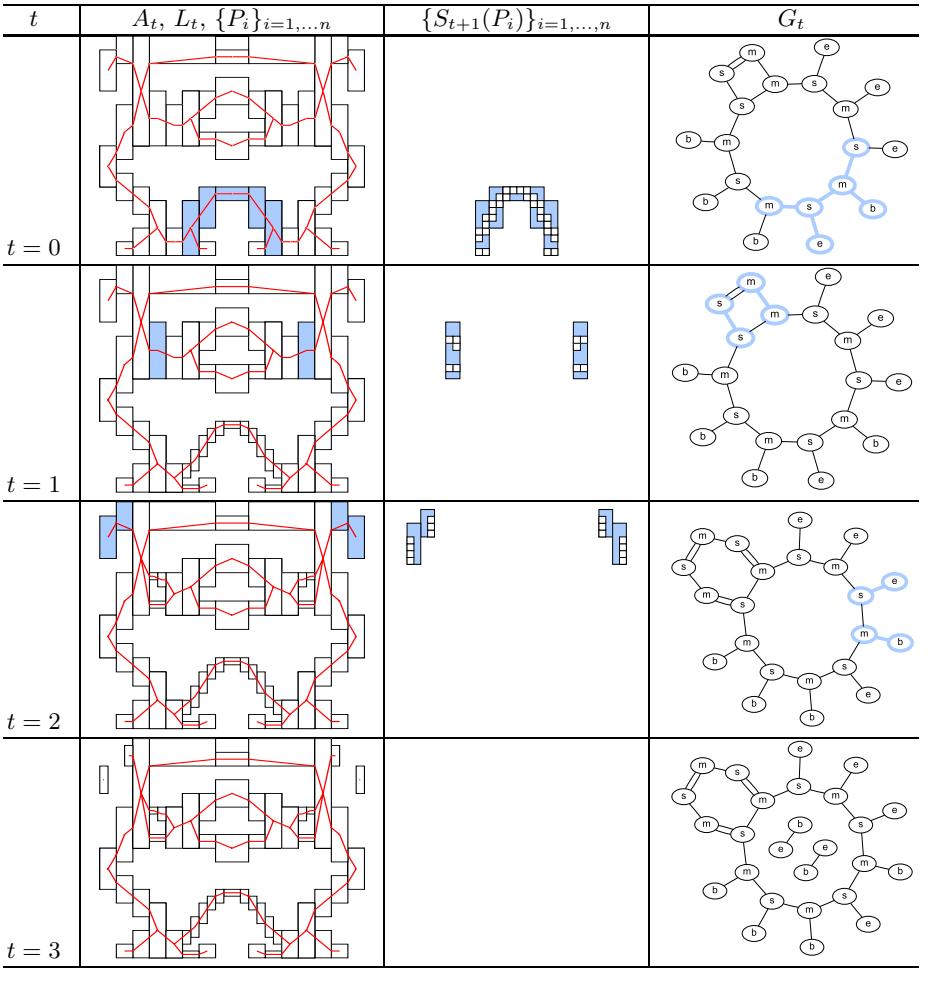


Fig. 4. The function $x^2+y^2+\cos(2\pi x)+\sin(2\pi y)+\sin(2\pi x^2)\cos(2\pi y^2) = 1$ on $[-1;1]\times[-1;1]$ *(top-left)* discretized in a set of cells by an algorithm described in [13]. We also present the result of our method on the right (A_3). For each update procedure, we show the recoded set of cells and the polygonalization *(left)*. The selected cells $P_1, P_2, ...P_n$ and their associated decomposition $S_{t+1}(P_1), S_{t+1}(P_2), ..., S_{t+1}(P_n)$ for the next iteration are presented *(center)*. The Reeb graph is presented in a circular format *(right)*. The considered arcs in the Reeb Graph are illustrated in a different color.

5 Conclusion and Future Work

In this article, we have proposed a global system for dynamic approximation of irregular graphical elements by lines. When we consider a local refinement by inclusion, we are able to update the topology of an irregular 2-D object by locally modifying its associated Reeb graph. Then, the polygonal description of this object is updated by changing the reconstruction by lines, in respect to its local geometry. We have also shown an application of our system for interactive approximation of implicit curves by lines. In future work, we would like to integrate our proposal into a complete software for 2-D interactive geometrical modelling.

References

[1] Coeurjolly, D.: Supercover model and digital straight line recognition on irregular Isothetic Grids. In *12th International Conference on Discrete Geometry for Computer Imagery*, LCNS 3429, pages 311–322, 2005.
[2] Coeurjolly, D., Zerarga, L.: Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids. In *Computer and Graphics*, 30(1):46–53, 2006.
[3] Cohen-Or, D., Kaufman, A.: Fundamentals of surface voxelization. In *Graphical models and image processing : GIMP*, 57(6):453–461, November 1995.
[4] Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques ? In *International Journal on Document Analysis and Recognition*, 3(2):73–88, published by Springer-Verlag, 2000.
[5] Debled, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition in linear time. In *12th International Conference on Discrete Geometry for Computer Imagery*, LCNS 3429, pages 311–322, 2005.
[6] de Figueiredo, L.H., Van Iwaarden, R., Stolfi, J.: Fast interval branch-and-bound methods for unconstrained global optimization with affine arithmetic. Technical Report IC-9708, *Institute of Computing, Univ. of Campinas*, June 1997.
[7] Gramain, A.: Topologie des surfaces. *Presses Universitaires Françaises*, 1971.
[8] Hilaire, X. and Tombre, K.: Robust and accurate vectorization of line frawings. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
[9] Kearfott, B.: Interval computations : introduction, uses, and resources. In *Euromath Bulletin* 2(1):95–112, 1996.
[10] Reeb, G.: Sur les points singuliers d'une forme de Pfaff complément intégrable ou d'une fonction numérique. In *Comptes Rendus de L'Académie ses Séances*, Paris 222, pages 847–849, 1946.
[11] Samet, H.: Hierarchical spatial data structures. In *Design and Implementation of Large Spatial Databases, First Symposium SSD'89*, 409:193–212, published by Springer, Santa Barbara, California, July 17–18 1989.
[12] Sivignon, I., Breton, R., Dupont, F., Andres, E.: Discrete analytical curve reconstruction without patches. In *Image and Vision Computing*, 23(2):191–202, 2005.
[13] Snyder, J.M.: Interval analysis for computer graphics. In *Computer Graphics*, 26(2):121–130, July 1992.
[14] Vacavant, A., Coeurjolly, D., Tougne, L.: Topological and geometrical reconstruction of complex objects on irregular isothetic grids. In *the 13th International Conference on Discrete Geometry for Computer Imagery (DGCI 2006)*. Szeged, Hungary, October 2006. *(to appear)*

Layout of Multiple Views for Volume Visualization: A User Study

Daniel Lewis, Steve Haroz, and Kwan-Liu Ma

University of California at Davis
{dglewis, sharoz, klma}@ucdavis.edu

Abstract. Volume visualizations can have drastically different appearances when viewed using a variety of transfer functions. A problem then occurs in trying to organize many different views on one screen. We conducted a user study of four layout techniques for these multiple views. We timed participants as they separated different aspects of volume data for both time-invariant and time-variant data using one of four different layout schemes. The layout technique had no impact on performance when used with time-invariant data. With time-variant data, however, the multiple view layouts all resulted in better times than did a single view interface. Surprisingly, different layout techniques for multiple views resulted in no noticeable difference in user performance. In this paper, we describe our study and present the results, which could be used in the design of future volume visualization software to improve the productivity of the scientists who use it.

1 Introduction

Volume visualization has become an increasingly important tool in many areas of scientific research and industry applications. It aids in the evaluation of engineering designs, the understanding of large-scale simulations, the viewing of multi-modal medical data, the mining of enormous web databases, and the analysis of homeland security information. Historically, the primary challenges in 3D visualization have been the limitations of hardware and algorithms. With the increasing power of graphics hardware and the maturity of rendering algorithms, the ability to generate useful visualizations is now more often limited by human interaction with the system. The development of appropriate user interfaces for performing volume visualization tasks, however, has not received sufficient attention. This need with respect to time-variant data is particularly great, as the amount of data can quickly overwhelm both the user and the computer. Innovative interface designs and layouts have been introduced, but most have not been adopted by commercial visualization tools. The effectiveness of these designs with respect to user performance needs to be evaluated against that of simpler conventional interfaces.

While the helpfulness of multiple views to time-varying data is obvious, we want to find the extent to which that is true. Furthermore, we want to determine if the arrangement of views influences user performance. Expecting the increased effort required by the more flexible layout schemes to result in longer task completion times, we were surprised to find that hypothesis incorrect. We compared the

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 215 – 226, 2006.
© Springer-Verlag Berlin Heidelberg 2006

conventional single-view interface with variations of multiple-view layouts by measuring how well users performed a set of tasks requiring fine tuning visualization parameters and combining visualization data. We conducted the study on both time-variant and time-invariant volumes to determine the utility of multiple-views.

2 Visualization Interfaces

In data visualization, initially raw data is filtered, mapped, rendered, and finally viewed. The filtering and mapping steps select which data the user wants to see and convert that data into an intermediate representation. The rendering step then generates images based on view position, lighting conditions, and any other parameters that help better present the data visually. A typical user performs this process iteratively, converging on a good representation that shows information of interest in the data. A good visualization interface should abstract the internal steps and complex parameter selection of visualization algorithms leaving the user with simple, high-level decisions made through an intuitive visual interface [16]. Prior research efforts on optimizing this process have primarily focused on improving the individual steps such as accelerating the feature extraction or rendering calculations. The user interface aspects of the problem, as well as the workflow for the entire process, have often been neglected.

The typical volume visualization interface has a single rendering window and a control panel that displays all the visualization parameters. Finding optimal settings for the myriad of parameters by manipulating many individual controls can be overly complicated and inefficient, and it can unnecessarily muddle the simplicity of a user's mental model of how an application works [4]. Having too many individual controls can reduce context and is unintuitive to users without expertise in graphics or visualization. One solution to this problem, the data-flow interface, represents the data exploration process by a directed graph of connected components. These components act upon the datasets or can be chained together in a pipeline. A few commercial visualization systems employ such an interface [1][18][19]. Data-flow interfaces display a graph showing the flow of information, which can be shared amongst collaborators to communicate the process needed to generate a requisite result. One shortcoming of the data-flow approach is that it does not indicate the history of the visualization process [9][10]. As is true in most interfaces, changes to the settings of one of the flow nodes cause the previous image to be lost.

Other approaches treat data exploration as an exploration of dimensions [15] or a search in the parameter space [9][7][10][11] resulting in simpler and more intuitive interface designs. The user can see the consequence of changes rather than the means by which to do so. Multiple views generally arranged in an evenly spaced layout show the same data with different parameters. The user, however, is required to browse through a large number of images, which could be quite time-consuming, and the user may have difficulty finding where to focus attention.

While these approaches to visualization interface design show promise, incorporating them into existing commercial visualization software, if possible at all, could require major restructuring of the systems. Commercial software tools, however, have begun to add simple displaying and browsing options such as multiple

linked views and focus plus context. Much research has been done on the benefits of multiple linked views [2] and their results are promising. A common theme repeatedly surfacing in volume visualization interfaces is the use of multiple views [3][13]. Nevertheless, none has studied or measured the extent of the helpfulness of giving volume visualization users multiple views, nor has anyone examined the effectiveness of the layout techniques for those views. In turn, we performed this study to learn exactly that.

3 Layout

Layout is an important and fundamental aspect of graphical user interface design. Appropriate arrangement of controls in a form is vital to usability, as it simplifies the flow of attention and localizes the proximity of widgets for related or sequential tasks. Control layout has even been used as a metric for evaluating user interfaces [14].

Most modern graphical interfaces have a variety of panels with each having its own unique set of controls and data displays. Statistical examinations of the usage of particular panels can reveal the appropriateness of the size and location these panels. The layout of these panels has been found to vary widely based on the specific type of application used [8]. Nevertheless, many of the same layouts are used for a variety of different applications without sufficient tests of the effectiveness of the layout in that particular type of use.

A more general approach to the examination of layout appropriateness regards the conformity of panels to a "regular pattern" [2]. Panels or windows with irregular shapes or inconsistent sizes have been found to benefit user controlled layout rather than automatic arrangement, yet when a simple and automatic layout is feasible, the extra burden on the user of manual arrangement is unnecessarily time consuming. The appropriateness of an automated layout should therefore be dependant on the regularity of data between multiple views.

The data sizes of different time steps in volume visualizations are usually consistent, so time variant volume visualization should benefit most from an interface with automatic layout. Unique layouts have already been found to be effective for video editing, which involves the manipulation of 2D time-variant data [5][17]. In contrast, we examined the effect of layout on time-variant and time-invariant 3D data.

4 Layout Variations

To test the helpfulness of multiple-view layouts, we created a volume visualization application that can utilize any one of four layout techniques. The application has a common control panel that allows the user to manipulate the colors and opacity levels of the volume's transfer function. The visualization is in a separate panel. The camera for each view can be independently controlled by dragging the mouse. The arrangement of views is the only difference between the four layout interfaces (see figure 1).

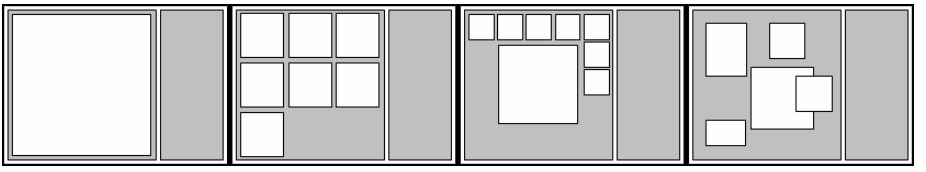


Fig. 1. Layout designs from left to right: simple (single view), split (evenly divided), radial (large primary view), and freeform (user controlled)

The simple layout has a single interactive view of the visualization. At any given time, the user can only see a single perspective of the data using the current transform and will not be able to see any previous actions or results. This basic interface is common in many applications and is therefore used as a control for our experiment. The split interface allows the user to evenly split the screen into different views of the data. All of the views are independent, interactive, and the same size. If the number of views is not a square number, blank space fills in the remaining area to make the boxes square. A user can create a new view with a 'New Window' button, which will copy the current view into a newly spawned view box. Each view also has a 'Load Data' button that loads a new set of data. All of the views have standard 'Close' and 'Maximize' buttons that remove or maximize the view respectively. If a square number of views remain, the views rearrange themselvess into a smaller square layout.

The radial layout consists of a large view in the center of the visualization panel with additional views surrounding the perimeter of the primary view. The outer views are smaller than the primary view, but they are all equally sized. They also have the same aspect ratio as the primary view. If not enough views can fill the perimeter, blank space is used. If the primary view is closed, one of the perimeter views will move to the center. Each view has a 'New Window' and 'Load Data' button that behaves similarly to that of the split interface. In this layout scheme, the maximize button moves the window into the central focus if the view is in the perimeter or maximizes the window to full screen if it is already in the center.

The freeform interface provides users with complete control over the layout of the views with no constraints. When a new view is opened, it appears in the left corner of the viewing pane with half of the width and height of the available area. The users can then freely move and resize the view within the viewing pane. Other than this distinction, it behaves similarly to the radial and split interfaces.

The common control panel to all four interface designs allows the user to modify the basics of the transfer function: color, curve shape, position, and opacity. The user can add predefined curve shapes to the transfer function or choose to hand draw it. The predefined shapes can be easily modified with the mouse to change the opacity, position, and size. Multiple transfer functions can be added or loaded from a previously saved transfer function to easily crop parts of the volume that are of interest (see figure 2).

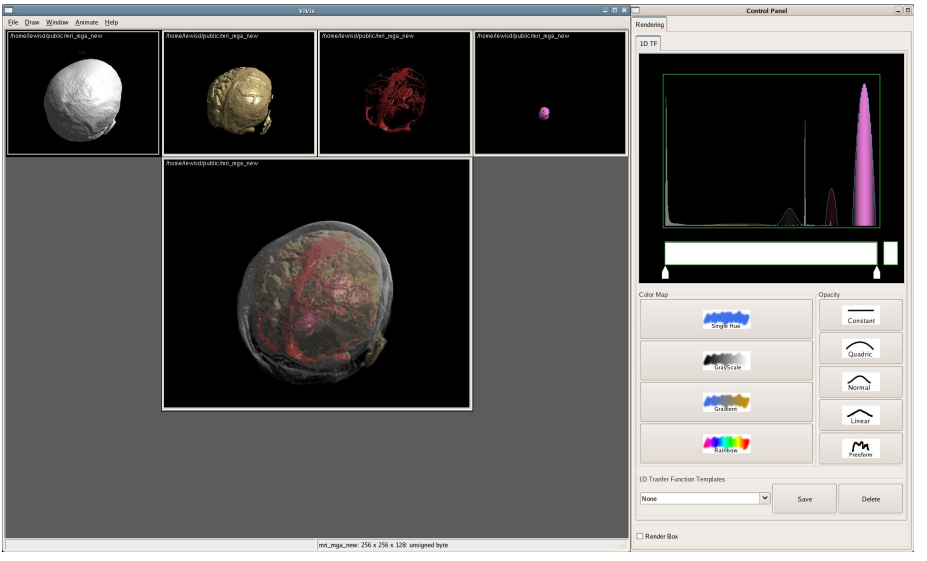


Fig. 2. This shows the brain dataset in the radial interface. Each of the brain's volume components is separated in a radial view, while the center shows the components with different levels of transparency. The control panel with the center view's transfer function is on the left.

5 Experiment Design

Twenty-four participants with a mixed range of technical backgrounds were involved in our study. They consisted of undergraduate and graduate students in Computer Science, Electrical Engineering, Civil Engineering, Mechanical Engineering, Mathematical, and Biological fields. We began each session by having our participants fill out a questionnaire to find their experience with computers and visualization. Their computer experience varied from having little experience (just email and typing documents) to being advanced developers. We randomly assigned users to interfaces and we manually reviewed the experience levels to assure even distribution of experience for each layout.

As another compensation for the range of experience, the users were given a tutorial to introduce them to the application. We explained the purpose of the transfer function and walked each person through the basics of configuring it to selectively crop parts of the volume. We also explained the benefits of differentiating parts of the volume by adjusting the color and opacity of ranges in the transfer function. We then had each user demonstrate competence in the program by creating a transfer function to show the bone structure, muscle, and skin layer of a dataset consisting of two scanned feet. We made sure that the users could achieve this task without any help from the experimenter to prevent any questions arising during the timed tasks. The tutorial given used the first layout that the user would encounter. After the first segment of the experiment was complete, a second tutorial was given to acquaint the user with the second layout interface. We trained each user for approximately ten minutes or until the user could easily complete simple tasks without our help.

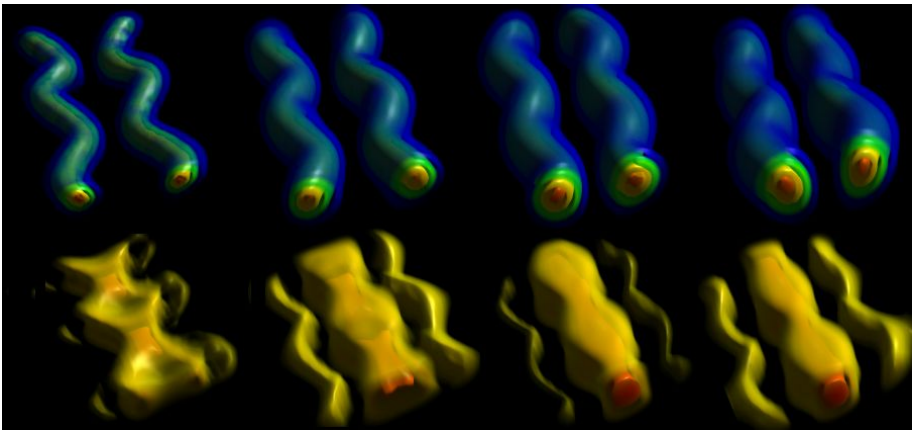


Fig. 3. These images show time steps 1 through 4 of the velocity data (top) and the vorticity data (bottom). Two spinning vortices produce high velocity winds in a tube-like shape (top), and they produce high vorticity, or local rotation, in the area in between the two tubes. Their progressive increase in strength and order can be seen in the successive time steps.

Each user performed two tasks, and each task was done using a different layout system. The tasks were individually timed to track the effect of the different interface designs on user performance. The first dataset is an MRI scan of a human head (see figure 2), and the second dataset is the velocity and vorticity produced by two spinning vortices over five time steps (see figure 3).

For the data containing the brain, the user was asked to adjust the transfer function to differentiate the skin, brain, blood vessels, and tumor inside the brain by varying their respective colors and opacities. We instructed the users to make all other features of the dataset transparent.

After the first test, the experimenter would explain the second user interface and allow the user to become comfortable with it. The other task was performed using two time-varying datasets with five time steps each. One dataset showed the magnitude of the velocity of two spinning vortices, while the other showed the vorticity, or local rotation, resulting from those vortices. The users were given the task of roughly sketching each of the five time steps for both the velocity and vorticity. After the users drew each time step, we asked them to explain the changes that occurred to velocity and vorticity throughout the time steps. The final time step for both velocity and vorticity was pre-drawn to ease the task of finding the best transfer function that best displayed the important features. After the test, the users were asked to explain what changed in the velocity and vorticity of the object in order to gauge their understanding of the volumes' transforms.

Each user performed the task on a different ordered pairing of the two datasets and two of the interfaces. Each permutation of the ${}_4P_2 = 12$ possible ordered couplings of the interfaces were given to two users. Each of those users was given a different ordering of datasets for a total of 24 combinations of two datasets and two out of four interfaces. The specific ordered pair was randomly assigned to each user to test all combinations and counterbalance any ordering or learning effects. We also had a

different set of users examine time-variant brain tumor datasets on all four interfaces (a within-user study) and found experience inconsistently affected performance.

The multiple views should allow the user to have multiple workspaces that allows for separating the three different volumes individually. Using multiple views should help modularize the tasks and combine all of the features in a final step. Using the single view, the users need to save the transfer function after each feature is found and then reload the previous transfer functions to add them together. The users however, may use a different strategy, such as trying to do everything from one view regardless of the interface used. We observed their use and asked them to describe their strategy when finished. We also asked which interface they preferred and why.

6 Results

Performance with time-invariant dataset (figure 4) was unaffected by the interfaceinterface used. All the interfaces had nearly identical average use times. The time-varying dataset showed a difference, but that difference only existed between the single-view layout and the multiple-view layout. The task completion times for the multiple-view layouts depended only on the actual dataset used.

6.1 Time-Invariant Data

Six people used each of the layouts for the fixed brain dataset. The users of the single-view layout, the experiment's control, for the time-invariant dataset required an average of 5:47 to complete the task (standard deviation of 1:50). The average times for the split, radial, and freeform layouts were 5:47, 5:38, and 5:30 respectively. The statistically insignificant differences show that both the availability and layout of multiple views was irrelevant to users' performance (see figure 4 and table 1).

6.2 Time-Varying Data

As with the time-invariant dataset, six participants used each of the layouts. With this dataset, however, a significant difference between the single-view times and the multiple-view times is apparent (see figure 5 and table 2). The average task-completion time for the single view is around five minutes longer than the average times of the multiple views. The single-view also had a significantly higher standard deviation. A clear difficulty arises from comparing images when an intermittent task (loading another volume) must be performed. This small pause appears sufficient to break users' attention to the point of being unable to identify small differences [12]. While the difference in quantity of views resulted in a strong difference, little performance difference resulted from changing the type of multiple-view layout.

6.3 User Preference

To evaluate user preference of the layouts each participant was asked to comparatively rate their preference of the two interfaces used on a scale from one to ten. The majority of those who had no preference between the first and second

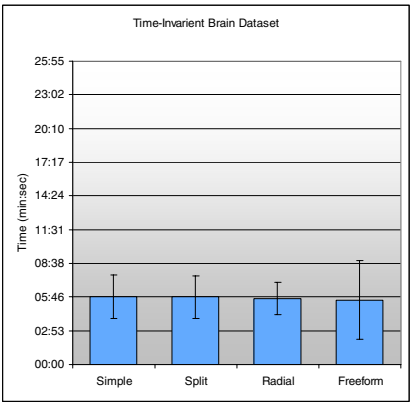


Fig. 4. Time-Invariant task execution times

Table 1. Time-invariant data results (note that average times are nearly identical)

Time-Invariant Brain Times

	Simple	Split	Radial	Free	All
Avg	05:47	05:47	05:38	05:30	05:41
Std Dev	01:50	01:49	01:24	03:21	02:05

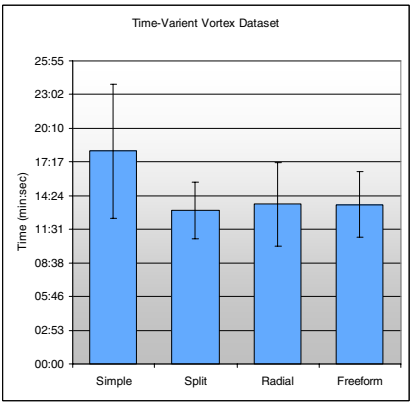


Fig. 5. Between-subject task execution times

Table 2. Time-varying data results for the vortex. The P-values were calculated between each layout's times and those of simple layout.

Time-Variant Vortex Times

	Simple	Split	Radial	Free	All
Avg	18:11	13:08	13:40	13:39	13:29
Std Dev	05:44	02:26	03:36	02:49	02:49
P-value	-	0.074	0.133	0.112	0.013

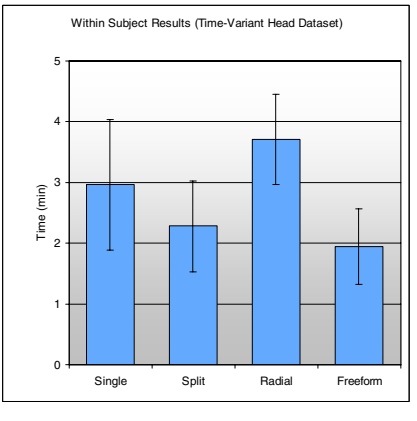


Fig. 6. Within-subject task execution times

Table 3. Time-varying data results. The P-values were calculated between each layout's times and those of simple layout.

Time-Variant Brain Times

	Simple	Split	Radial	Free	All
Avg	2:58	2:17	3:43	1:57	2:33
Std Dev	1:04	0:45	0:45	0:37	1:06
P-value	-	0.17	0.18	0.05	0.003

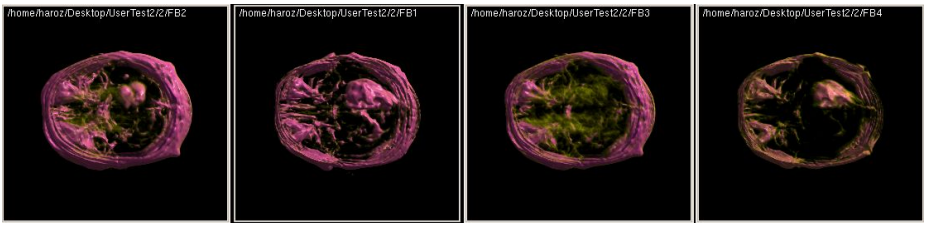


Fig. 7. These images show a time-variant dataset in which a tumor can be seen growing and shrinking inside a brain. This is one of the datasets used in the with-user study.

interfaces used the simple single view for the brain dataset and a multiple-view for the vortex dataset. The lack of necessity of the multiple views for the brain dataset resulted in few even realizing that a difference existed. The preferences were as follows: one participant preferred the simple interface; six preferred the split interface; five preferred the radial interface; and seven preferred the freeform interface. As with the task completion times, the user preferences between the different multiple-view layouts were equivalent.

6.4 Within-Subject Study

To examine within-subject time differences, we performed a separate study with twelve participants. Though different, these users were selected from a similar population to that of the first study. Each person tracked the size of a brain tumor as it grew and shrank between five time steps (see figure 7). After undergoing a training session using a sample dataset, the users were timed as they performed the task using each interface with a different dataset. Their only feedback was to state whether the tumor grew, shrank, or remained unchanged between each time step. The orders of the datasets and interfaces were selected randomly to give a nearly even distribution of each.

After examining the data, we realized that the times for all of the users' first trials were highly inconsistent. Due to the unusually high variation in that set of numbers, we chose to drop everyone's first trial. Therefore, only three of each user's four timed trials were included in the analysis. Even so, all dataset and interface combinations were presented and used in the analysis at least twice. Training the participants using the interface of their first trial was a likely culprit for the anomaly.

The results (see figure 6 and table 3) show long times and high standard deviations for the simple and radial layouts. In contrast, the split and free layouts have lower times and standard deviations. With the exception of the poor performance of the radial layout, these results are similar to those of the between-user study. The split and free layouts ultimately performed better than did the simple layout, but the performance difference between the split and free layouts is statistically insignificant.

7 Discussion and Future Work

While the between-user study was conducted on twenty-four people, each unique combination of two layouts and two datasets was given to only two people.

Conducting the study on more people could have helped reduce the high variance in the time measurements, which would lower the P-values to more acceptable levels. Had we had the opportunity, we would have also preferred to conduct an additional study on specialists in a particular field that makes use of time-variant volumes. If all of our users had the knowledge to describe complex datasets using precise and consistent language, we could have conducted a far more detailed study that required precise, more falsifiable responses. Though the vortex and tumor datasets had easily identifiable features, the volumes' minimal complexity made for somewhat simple responses from the participants such as "it got skinnier". In spite of this simplicity, the users of the single view still had a very difficult time identifying the changes in the volumes through time.

Although, we only looked at four view layouts, their simplicity and generality covered a significant number of the arrangements used in the few existing multiple-view volume visualization systems. The number of possible layouts is infinite, so we tested simple views that we believe are most practical. Nevertheless, this limited testing could have resulted in a missed opportunity to find a better, albeit more complex, layout.

An interesting observation was that some subjects using the simple view with the time-varying data expressed the desire to use multiple views. The numerous switching between views directly affected their overall time. This lack of direct comparison clearly limited their ability to perceive changes in time-varying data without knowing precisely where to attend.

Another notable find was the users' tendency to arrange the free layout views into a tiled layout similar to that of the split design. According to [2], this wasted expenditure of time with window management should have resulted in longer times for the free layout. A possible explanation for this discrepancy is the difficulty in making our application rearrange windows for the constrained layouts. The lack of a high affordance means of achieving this simple goal resulted in a couple of users closing the views and then reopening them in the desired order. The split view times and the radial view times may have been shorter if rearrangement were simpler.

8 Conclusions

This study identifies the differences in effectiveness of a variety of simple layouts for multiple-views of volume visualizations. We have presented four interfaces for volume visualizations with varying layout constraints and arrangements. The results support the obvious advantage of multiple views and their ability to aid in achieving faster and more accurate performance of feature identification for time-varying data. The experiment's control, the single view, resulted in slower task completion times as well as significantly decreased accuracy in describing changes between datasets. As was expected, for a time-invariant dataset, the number of views had no noticeable effect on the accuracy or speed of task accomplishment.

The more surprising observation is the irrelevance of the type of multiple-view layout as well as the level of constraint for that layout. We hypothesized that the constrained layout system would outperform the others, yet our results show that the layout of multiple volume visualization views has little, if any, effect on users'

performance. While we only tested three multiple-view layout designs, a wide range of designs exist. The lack of a significant and consistent performance difference between the free layout and the automated layouts may be implicative that time spent on window management is counteracted by the constraints of the automated layouts. The regularity in the size of the time variant dataset should have resulted in the automated layouts eliminating effort spent on window management [2]. Perhaps a similar user study carried out with a volume that significantly changes size, shape, and aspect ratio over time could yield different results, but obtaining such a dataset may be impractical.

Acknowledgements

We greatly appreciate all those who participated in our user study. This work is supported in part by NSF under grants IIS-0552334 and ACI-0222991.

References

1. Abram, G., Treinish, L. An extended data-flow architecture for data analysis and visualization. In *Proceedings of the IEEE Visualization '95 Conference*., pp. 263–270. October 1995.
2. Bly, S. A. and Rosenberg, J. K. A comparison of tiled and overlapping windows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. M. Mantei and P. Orbeton, Eds. CHI '86., New York, NY 1986 ACM Press.
3. Calhoun, P.S., Kuszyk, B.S., et al. Three-dimensional volume rendering of spiral CT data: theory and method. In *Radiographics*. 19(3). pages 745-64. 1999.
4. Cooper, A., Reimann, R. *About Face 2.0: The Essentials of Interaction Design*. Wiley, New York, 2003.
5. Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote, J., Golovchinsky, G., Uchihashi, S., and Wilcox, L. A semi-automatic approach to home video editing. In *UIST 00: Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology*. pages 81-89. New York, NY, 2000. ACM Press.
6. Gresh, D. L., Rogowitz, B. E., Winslow, R. L., Scollan, D. F., Yung, C. K. 2000. WEAVE: a system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings of the Conference on Visualization*. pages 489-492. IEEE Visualization. IEEE Computer Society Press, Los Alamitos, CA.
7. He, T., Hong, L., Kaufman, A., Pfister, H. Generation of transfer functions, with stochastic search techniques. In *Proceedings Of IEEE Visualization '96 Conference* (October 1996), pp. 227–234.
8. Henderson, D. A., Card, S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. In *ACM Trans. Graph.* 5(3) pages 211-243. 1986.
9. Jankun-Kelly, T. J., Ma, K.-L. A spreadsheet interface for volume visualization. In *Proceedings of IEEE Visualization 2000 Conference* (2000).
10. Ma, K.-L. Image graph: A novel approach to visual data exploration. In Proceedings of *IEEE Visualizaton '99 Conference* (1999), pp. 81–88.

11. Marks, J., Andalman, B., Beardsley, P., Freeman, W., Gibson, S., Hodings, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., And Shieber, S. Design Galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH '97 Conference* (August 1997), pp. 389–400.
12. O'Regan, J. K., Rensink, R. A., & Clark, J. J. Change-blindness as a result of 'mudsplashes'. In *Nature*, 398(6722), pages 34-34. 1999.
13. Roberts, J. C., On Encouraging Multiple Views for Visualisation. In *Proceedings of the international Conference on information Visualisation* IEEE Computer Society, Washington, DC, 8. 1998.
14. Sears, A., Layout appropriateness: a metric for evaluating user interface widget layout, In *IEEE Transactions on Software Engineering*. 19(7), July 1993, pp. 707-719.
15. Tory, M., Potts, S., And Möller, T. A Parallel Coordinates Style Interface for Exploratory Volume Visualization. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1. pages 71-80. Jan/Feb 2005.
16. Tzeng, F.-Y., Ma, K.-L., Lum, E. An intelligent system approach to higher-dimensional classification of volume data. In *IEEE Transactions on Visualization and Computer Graphics* 11, 3 (2005), 273–284.
17. Ueda, H., Miyatake, T., Sumino, S., and Nagasaka, A. Automatic structure visualization for video editing. In *CHI 93: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pages 137-141. New York, NY, USA, 1993. ACM Press.
18. Upson, C., Faulhaber, T., Kamins, D., Schlegel, D., Laidlaw, D., Vroom, J., Gurwitz, R., And Van Dam, A. The application visualization system: A computational environment for scientific visualization. In *IEEE Computer Graphics and Applications* 9, 4 (1989), 30–42.
19. Young, M., Argiro, D., And Kubica, S. Cantata: Visual programming environment for the Khoros system. In *ACM SIGGRAPH Computer Graphics* 29, 2 (1995), 22–24.

Video Indexing and Retrieval in Compressed Domain Using Fuzzy-Categorization

Hui Fang, Rami Qahwaji, and Jianmin Jiang

EIMC Dept., University of Bradford
{h.fang, r.s.r.qahwaji, j.jiang1}@bradford.ac.uk

Abstract. There has been an increased interest in video indexing and retrieval in recent years. In this work, indexing and retrieval system of the visual contents is based on feature extracted from the compressed domain. Direct possessing of the compressed domain spares the decoding time, which is extremely important when indexing large number of multimedia archives. A fuzzy-categorizing structure is designed in this paper to improve the retrieval performance. In our experiment, a database that consists of basketball videos has been constructed for our study. This database includes three categories: full-court match, penalty and close-up. First, spatial and temporal feature extraction is applied to train the fuzzy membership functions using the minimum entropy optimal algorithm. Then, the max composition operation is used to generate a new fuzzy feature to represent the content of the shots. Finally, the fuzzy-based representation becomes the indexing feature for the content-based video retrieval system. The experimental results show that the proposal algorithm is quite promising for semantic-based video retrieval.

1 Introduction

Video indexing and retrieval is becoming an important application because of the increasing demand for such systems that could provide effective organization, search and display of multimedia contents. This is caused by the tremendous growth of the multimedia archives, particularly in the digital video database [2] [4] [5] [6] [9]. In order to analyze the content of the videos, the meaningful information needs to be extracted. However, any single low-level feature is not adequate to provide full and efficient representation for videos. Therefore, a hyper feature extraction mechanism is important for video indexing and retrieval systems.

Systems based on fuzzy logic usually require fewer variables, use a linguistic rather than a numerical description, and can relate output to input by analyzing and understanding the data as a human being. Fuzzy logic has been used for video processing in recent years. In [14], fuzzy logic and fuzzy rules were defined to provide semantic interpretation for video contents. However, in [14] fuzzy logic was used for obtaining the rules but the membership functions were set manually. In this paper, an automated technique for determining the fuzzy membership functions is presented using the minimum entropy algorithm.

The traditional video indexing and retrieval process has four steps: temporal segmentation, key frames selection, feature extraction and classification. In this paper,

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 227 – 234, 2006.
© Springer-Verlag Berlin Heidelberg 2006

we assume that the shot boundaries have been detected and all the videos are organized in shots. The key-frame selection algorithm [1] [2] extracts some frames to represent the visual content in every shot. Following these two steps, two types of features, the spatial features and the temporal features, can be extracted separately from the shots and the key-frames. The spatial features are mainly visual features such as color, texture and shape which are extracted from key-frames by image analysis techniques. In [8], image database retrieval is carried out using the mean and variance for the local blocks that were extracted from DCT domain to create a histogram. In [11], the authors use the average color feature in the Luv space as a spatial feature to match the frames. The motion features represent the camera and objects movements and many motion descriptors are designed. Motion activity [3], which is provided by The MPEG-7 standard, shows the intensity of the activity. The magnitude of block-based motion vectors is directly obtained from MPEG compressed domain and its standard deviation is classified into five bins. The Pixel Change Ratio Map (PCRM) [4], which is motivated by the frequency of the pixel intensity changes, is another temporal feature characterizing motion content in a video shot. If the objects in a shot move faster, the pixel intensity changes more. A tensor histogram extracted directly from temporal slice [5] is another good motion descriptor to index and organize the motion content of video shots. In this method, the tensor histogram is compared with two more features, Gabor feature and Co-Occurrence Matrix to prove it is an efficient feature in the compressed domain. A descriptor using block-based motion vector and PCA is given in [6]. To compare the efficiency of these motion features, in [7] a framework to measure the intensity of motion activity of video shots is built to test the performance of some temporal descriptors and the results show that the MPEG-7 motion activity descriptor is one of the best performers. When obtaining all the features for retrieval, the final step is matching the features between the query shots and database shots. [9] [10] use a total distances which is obtained based on the weighted distances of various features while [5] use a hierarchical structure to match these two types of features sequentially. Using these steps, the retrieval results are ranked by the similarity distances between the query example and the descriptors in the video database.

Although these traditional technologies have got some reasonably good performance, they still have some drawbacks: (1) some features are high-dimensional features, which increase the computational complexity of the whole process and affect the retrieval speed; (2) some features are classified into different semantic classes by crisp threshold which makes the system inflexible and damages the performance; (3) a mechanism is required to integrate multi-features into a retrieval system. As illustrated before, fuzzy representation is used because it gives a more compact description for improving the retrieving performance [12]. Furthermore, it enables a human-based interpretation for the discriminative features. In addition, a new feature can be generated by combining different features in different fuzzy sets through the fuzzy composition relations. In this paper, a fuzzy-categorizing video indexing and retrieval paradigm is constructed and a new feature for comparing the similarity between the shots is exploited to improve the performance of the video retrieval system.

This paper is organized as follows: The proposed algorithm is proposed in section 2. Experimental results on a basketball video database are shown in section 3. Finally, the concluding remarks are provided in section 4.

2 Algorithm Design

2.1 Paradigm of the Video Indexing and Retrieval

Video Database and multimedia library indexing and retrieval system plays an important role in the multimedia application field. Despite the intensive research efforts, the development of low-level feature-based video retrieval algorithm is still in its early stage. Linking the low level features to high level semantics remains an open problem in computer vision [14]. In this paper, an inter-layer fuzzy feature is designed to bridge the gap between the huge amount of data representing low-level features and the meaningful content of videos. The low-level perceptual features are trajectory to a semantic fuzzy domain and the retrieval system retrieve the content of videos based on the integrated fuzzy features. In other words, the low-level features are grouped into some meaningful categories by the fuzzy logic and then the fuzzy values of those features are integrated to index the shots in the video database.

At present, video compression standards such as MPEG2, MPEG4, H263 and H264 have been designed to reduce the storage and transmission requirements. Therefore, the video indexing and retrieval in the compressed domain is a promising field. In this paper, temporal features and spatial features that are extracted from the compressed domain are employed in the fuzzy system for video shot-based retrieval. The details of these two features are given below:

2.1.1 Statistical Spatial Variance Feature

Given a shot with K I-frames (which are the intra-coded frames) and $M \times N$ blocks with DCT coefficients in each frame, the variance of each block, based on [8], is given in equation (1),

$$\text{var} = \frac{1}{64}\sum_{u=0}^{7}\sum_{v=0}^{7} C^2(u,v), \qquad (u,v) \neq (0,0) \tag{1}$$

where C(u, v) is the AC coefficient in this block. According to Equation (1), the spatial average variance for each frame is shown in equation (2),

$$AV = \frac{1}{M \times N}\sum_{m=0}^{M}\sum_{n=0}^{N} \text{var}_{mn} \tag{2}$$

where AV represents the average variance for each I-frame and *var* represents the same variance in equation (1). The spatial average variance for the whole shot is given as follow:

$$SF = \frac{1}{K}\sum_{k=0}^{K} AV_k \tag{3}$$

where K is the total number of I-frames in the shot. Variance is a key feature in video and image retrieval because it reveals the texture on the spatial image or frame and the variance is similar for the same semantic content. For example, the bird-view of the crowds has a large variance while the close-up of some players has a relative smaller variance value.

2.1.2 Statistical Temporal Deviation Feature

This feature is derived from the motion activity feature in MPEG-7 [3]. Instead of quantizing the motion magnitude deviation into 5-bins histogram, the average of the motion magnitude deviation of the shot is considered to be the motion feature. Motion features are important for video processing because the temporal continuity of visual cues is unique in a video stream. This feature does not exist for static images.

When the features are extracted, they are transformed into the fuzzy domain to relate them to some semantic concepts. In this paper, three concepts, which include full-court match, penalty and close-up, are used to train the fuzzy membership functions. The details are shown in the next subsection. Then the max composition operation is used to generate a 9-bins fuzzy feature to represent the semantic meaning for every shot. Then the correlation between the query shot and the shots in the database is calculated based on this semantic fuzzy feature.

We believe that there are three advantages for using this feature: it is (a) a novel feature linking the low-level features to the semantic categories; (b) a novel feature to represent the shot compactly and efficiently; (c) a novel feature to integrate the temporal feature and the spatial feature.

2.2 The Generation of Fuzzy Membership Functions

At the first step, fuzzy membership functions are required in order to fuzzify the features into the fuzzy sets. A common fuzzy membership function is shown in Figure-1,

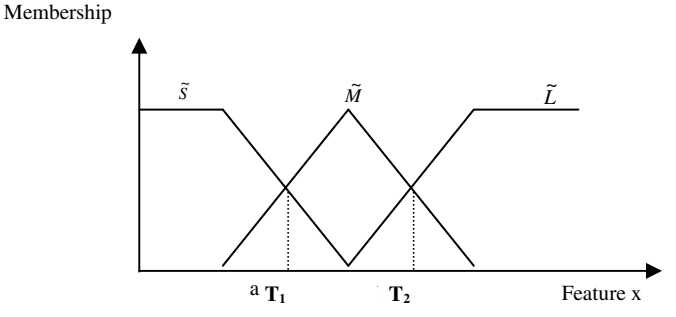


Fig. 1. Illustration of the fuzzy membership function

where three fuzzy sets S, M and L are shown to represent the fuzzy sets in the feature domain. The corresponding membership functions are defined as follows:

$$M(x)_L = \begin{cases} 1 & if \quad x > c \\ \dfrac{x-b}{c-b} & if \quad b < x < c \end{cases}$$

$$M(x)_M = \begin{cases} \dfrac{c-x}{c-b} & if \quad b < x < c \\ \dfrac{x-a}{b-a} & if \quad a < x < b \end{cases} \tag{4}$$

$$M(x)_S = \begin{cases} 1 & if \quad x < a \\ \dfrac{b-x}{b-a} & if \quad a < x < b \end{cases}$$

where x represents a feature, and the three parameters a, b, and c, represent the boundaries of the three fuzzy sets as shown in Figure 1.

In this paper, the minimum entropy optimization threshold algorithm [13] is used to train the membership function in order to fuzzify the motion features and the spatial features into the fuzzy sets. As shown in figure-1, two Thresholds T1 and T2 are trained by getting the minimum entropy value to separate the three classes. For T1 and T2, the membership values $M(x)$ of the classes are set to 0.5 [13]. The entropy of the threshold dividing two fuzzy sets can be calculated using the following equations [13]:

$$\begin{aligned} S(x) &= p(x)S_p(x) + q(x)S_q(x) \\ S_p(x) &= -[p_1(x)\ln p_1(x) + p_2(x)\ln p_2(x) \\ S_q(x) &= -[q_1(x)\ln q_1(x) + q_2(x)\ln q_2(x) \end{aligned} \tag{5}$$

where $p_1(x)$ is the probability that class 1 samples are located in region p, $p_2(x)$ is the probability that class 2 samples are located in region p, $q_1(x)$ is the probability that class 1 samples are located in region q , and finally $q_2(x)$ is the probability that class 2 samples are located in region q. The percentage of total number of samples belonging to region p in all samples is referred to as p(x), while the percentage of total number of samples belonging to region q in all samples is referred to as q(x). In this work, regions p and q are divided by thresholds T1 or T2 as shown in Figure 2.

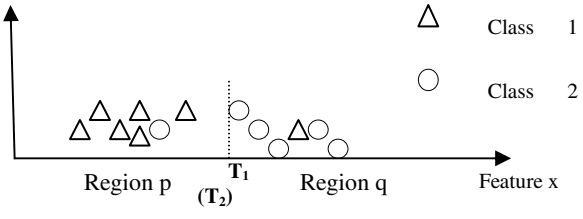


Fig. 2. Illustration of the entropy equation

When finding the minimum entropy thresholds by training the dataset and the parameters are obtained as follow:

For the motion feature:
T1 = 2.5, T2 = 9.2, a = -0.85, b = 5.85, c = 12.55;
For the variance feature:
T1 = 61, T2 = 87, a = 48, b = 74, c = 100;

Then, two membership function sets are obtained to fuzzify the two features into the fuzzy domain.

2.3 The Generation of Fuzzy Feature and Similarity Measure

After the previous step, each feature is fuzzified into three fuzzy sets. In order to keep the possibility of the principle elements in the fuzzy sets, the max composition operation [13] is used to calculate the possibility that the two features belong to the specific fuzzy sets. Then a nine-bin feature is generated to represent the shot in a semantic layer. At last, the Cosine Amplitude is employed for measuring the similarity between the shots, as shown in equation (6).

$$S_{ij} = \frac{\left| \sum_{k=1}^{9} f_{ik} f_{jk} \right|}{\sqrt{\left(\sum_{k=1}^{9} f_{ik}^{\ 2} \right) \left(\sum_{k=1}^{9} f_{jk}^{\ 2} \right)}} \tag{6}$$

where f_{ik} represents the fuzzy feature of the query shot and f_{jk} represents the fuzzy feature of each shot in the database. The larger S_{ij} means that the query shot and the retrieval shot are more similar.

3 Experimental Results

In this section, the fuzzy system described in the previous section is applied to a basketball database. The database consists of 440 video sequences taken from some NBA basketball matches. This database includes 210 full-court match shots, 70 penalty shots and 170 close-up shots. The duration span of the video shots varies is from 4s to 73s and the frames rate is 29.97frames/sec.

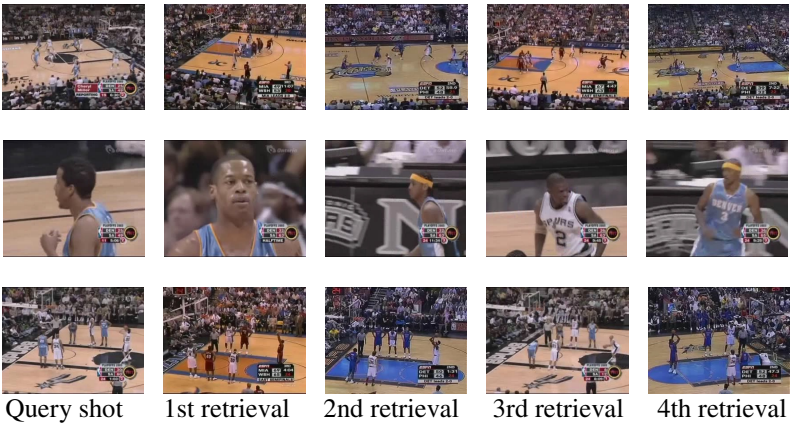


Fig. 3. Examples of video retrieval results for the top 4 best matching shots

The database is used to train the membership functions which have been illustrated in section 2 and the 9-bins fuzzy features are used to index the database. Based on the Jack-Knife technique, we use 90% of this dataset to train the membership function and the remaining 10% are used to test the retrieval performance. Figure 3 shows some examples for our video retrieval experiments. The first frame is exploited to represent the shot.

To further prove the efficiency of our algorithm, the precision of retrieval is considered. The average accuracy of all the retrieval results based on the return number is shown in Table-1, we use percentage as a unit because the numbers of the class samples are different. At the same time, the accuracy for different classes is drawn in figure-4. The shots in class 1 are the full-court match shots, the shots in class 2 are the penalty shots and the shots in class 3 are the close-up shots.

Table 1. Retrieval Accuracy according to the num percentage

Retrieval (%)	10	20	30	40	50
Accuracy	93.3	93.0	92.1	90.8	90.3
Retrieval(%)	60	70	80	90	100
Accuracy	90.2	89.8	89.5	88.8	87.4

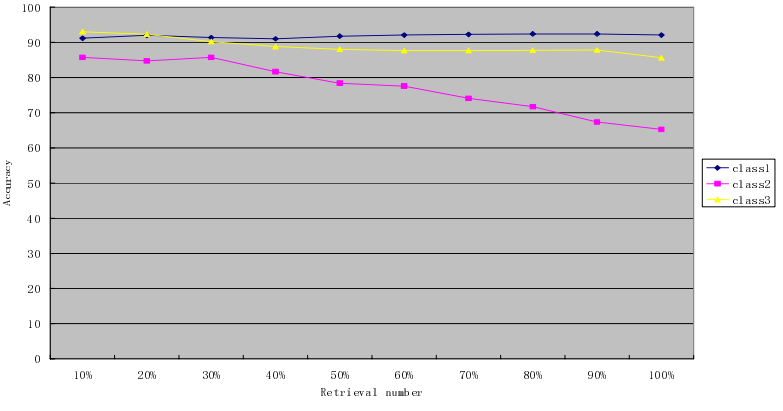


Fig. 4. The Accuracy of the retrieval results

4 Conclusion

A new fuzzy feature for retrieving the visual semantic content is presented in this paper. By analyzing the visual content in a basketball video database, three classes, which include full-court match, penalty and close-up, are used to train the fuzzy membership functions of a temporal feature and a spatial feature through the minimum entropy optimal algorithm. Then, the max composition operation is used to generate a 9-bins fuzzy feature. Finally, this representation is applied for the content-based video retrieval system. Experimental results show that this algorithm is a promising solution

for the content based video retrieval system and has a computational efficiency due to the compact feature and indexing process in the compressed domain.

In the near future, we would like to extend this application to provide fuzzy representation for other videos representing different real-life events based on the multi-dimensional feature extraction techniques. And neural network will be used to replace entropy in calculating fuzzy membership functions because of its suitability for higher dimensional features.

Acknowledgment

The authors wish to acknowledge the financial support from the EU IST Framework-6 programme under the IP project: Live staging of media events (IST-04-027312).

References

1. A. Hanjalic and H.J. Zhang: An Integrated Scheme for Automated Video Abstraction Based on Unsupervised Cluster-Validity Analysis. IEEE Transaction on Circuit and Systems for Video Technology, Vol.9, No.8, December 1999;
2. A.D. Doulamis and N.D. Doulamis: Optimal Content-based Video Decomposition for Interactive Video Navigation. IEEE Transactions on Circuits and Systems for Video Technology, Vol.14, No.6, June 2004;
3. B.S. Manjunath, P. Salembier, T. Sikora: Introduction to MPEG-7: Multimedia content description interface. John Wiley& Sons, LTD, ISBN 0471486787, 2002;
4. H. Yi, D. Rajan, L.T. Chia: A new motion histogram to index motion content in video segments. Pattern Recognition Letters 26 (2005) 1221-1231;
5. C.W. Ngo, T.C. Pong, H.J. Zhang: "On clustering and retrieval of video shots through temporal slices analysis. IEEE Tansactions on Multimedia, Vol.4, No.4, December 2002;
6. E. Sahouria and A. Zakhor: Content analysis of video using principal components. IEEE Transactions on Circuits and Systems for Video Technology, Vol 9, No.8, Dcemember 1999;
7. K. A. Peker and A. Divakaran: Framework for measurement of the intensity of motion activity of video segments. J. Vis. Commun. Image R. 15 (2004) 265-284;
8. G. Feng and J. Jiang: JPEG compressed image retrieval via statistical features. Pattern Recognition 36 (2003) 977-985;
9. J. Fan, W.G. Aref, A.K. Elmagarmid, M.S. Hacid, M.S. Marzouk, X. Zhu: MultiView: Multilevel video content representation and retrieval. Journal of Electronic Imaging 10(4), 895-908 (October 2001);
10. S.F Chang, W. Chen, H.J. Meng, H. Sundaram and D. Zhong: A Fully Automated Conent-Based Video Search Engine Supporting Spartiotemporal Queries. IEEE Transactions on Circuits and Systems for Video Technology, Vol.8, No.5, September 1998;
11. A. Hanjalic, L. Lagendijk, J. Biemond, and J. Biemond: Automated High-level Movie Segementation for Advanced Video Retrieval System. IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.4, June 1999, pp580-588;
12. A. Doulamis, N.D. Doulamis, S. D. Kollias: A Fuzzy Video Content Representation for Video Summarization and Content-based Retrieval. Signal Processing 80 (2000) 1049-1067;
13. Timothy J. Ross: Fuzzy Logic with Engineering Applications. John Wiley & Sons, Ltd, 2004;
14. A. Dorado, J. Calic, E. Izquierdo: A Rule-Based Video Annotation System. IEEE Transactions on Circuits and Systems for Video Technology, Vol.14, No.5, May 2004;

Computing Homology for Surfaces with Generalized Maps: Application to 3D Images

Guillaume Damiand[1], Samuel Peltier[2,⋆], and Laurent Fuchs[1]

[1] SIC - bât. SP2MI, Bvd M. et P. Curie
BP 30179, 86962 Futuroscope Chasseneuil Cedex - France
{damiand, fuchs}@sic.univ-poitiers.fr
[2] PRIP - Vienna University of Technology
Favoritenstr. 9/1832, A-1040 Vienna - Austria
sam@prip.tuwien.ac.at

Abstract. In this paper, we present an algorithm which allows to compute efficiently generators of the first homology group of a closed surface, orientable or not. Starting with an initial subdivision of a surface, we simplify it to its minimal form (minimal refers to the number of cells), while preserving its homology. Homology generators can thus be directly deduced from the minimal representation of the initial surface. Finally, we show how this algorithm can be used in a 3D labelled image in order to compute homology of each region described by its boundary.

Keywords: topological features, homology generators, generalized maps, minimal subdivision.

1 Introduction

In many domains of computer graphics (geometric modeling, computational geometry, image structuring,...), combinatorial structures are used to describe objects subdivided into cells of different dimensions (vertices, edges, faces, volumes,...). A common problem in each domain is to characterize structural (topological) properties of handled objects. Homology is a topological invariant, classically studied in algebraic topology, which characterizes an object by its "holes" in each dimension, which corresponds to its connected components in dimension 0, its tunnels in dimension 1, its cavities in dimension 2 ; this notion of hole can be generalized in any dimension. For each dimension d, the number of $d-$dimensional holes of a given object is called its d^{th} Betti number.

Generalized maps [1,2], are a combinatorial cellular structure which can be used to build and handle quasi-manifolds. In this work, generalized maps are used to compute a minimal cell decomposition (called *minimal map*) of a closed surface.

Kaczynski et al. proposed a general algorithm [3] for reducing the number of cells of a given object, without changing its homology. If the object is orientable, this general algorithm provides its Betti numbers.

⋆ Partially supported by the Austrian Science Fund under grants S9103-N04.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 235–244, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we adapt this general algorithm to the model of generalized maps, and we extend it (by adding an operation, called shifting edge operation) in order to compute a minimal map for any surface, orientable or not. Moreover, we show that a set of generators of the first homology group can directly be deduced from the minimal map.

In Section 2, basic notions related to generalized maps and homology groups are recalled. The removal operations, which are used to compute a minimal map are introduced. In Section 3, the algorithm for computing a minimal map is detailed, and its complexity is discussed. Finally, we prove that our algorithm provides a minimal object with the same homology that the initial one. In Section 4, we show how to use the minimal map algorithm for characterizing each region in a 3D labeled image.

2 Recalls

2.1 Generalized Maps

A subdivision of a 3D topological space is a partition of the space into 4 subsets whose elements are *cells* of dimension 0, 1, 2 and 3 (respectively called vertices, edges, faces and volumes). In the following, an i-cell will denote a cell of dimension i. Border relations are defined between these cells and the border of an i-cell is a set of *(j<i)*-cells . Two cells are *incident* if one belongs to the border of the other, and two i-cells are *adjacent* if they are both incident to a common *(j<i)*-cell.

For 3D subdivisions, incidence and adjacency relations can be represented using 3-dimensional generalized maps (3-*G-maps*) [2]. Intuitively, a 3D generalized map can be obtained by successive decompositions of a 3D object. We first decompose the volumes of this object, then the faces of these volumes, the edges of these faces, and then the vertices of these edges. The elements obtained from the last decomposition are called *darts* and are the basic elements of the generalized map definition.

To obtain the map, adjacency relations between i-cells (denoted α_i) are reported onto darts. *Involution*[1] α_0 connects two vertices incident to a same edge, face and volume, α_1 connects two edges incident to a same vertex, face and volume, and so on for all dimensions. These α_i have to verify some particular properties in order to ensure the validity of the represented subdivision (see Fig. 1 for an example and [2] for generalized map definition).

Within the generalized map framework, all cells are implicitly represented through the notion of *orbit*. Given, $\{p_1, \ldots, p_j\}$, a set of (darts) involutions, and a dart d, an orbit $< p_1, \ldots, p_j > (d)$ is the set of darts that can be reached with a breadth-first search algorithm, starting with d, and using all combinations of p_i $\forall k, 1 \leq k \leq j$. Based on the cells definition, we can retrieve the classical *cell degree* notion. The degree of an i-cell c is the number of distinct *(i+1)*-cells incident to c.

[1] An involution f on S is a one to one mapping from S onto S such that $f = f^{-1}$.

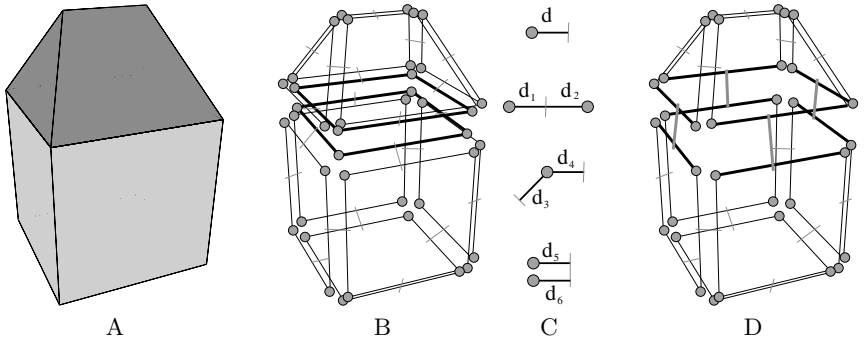


Fig. 1. Example of a 3D generalized map. (A) A 3D object. (B) Implicit representation of the corresponding generalized map. (C) Explanations of the graphical representation. d: representation of a dart. d_1 and d_2: two darts in relation by α_0. d_3 and d_4: two darts in relation by α_1. d_5 and d_6: two darts in relation by α_2. α_3 is not represented on figures but can be deduced from the shape of objects. (D) Map obtained from (B) by removing the black face. Adjacent faces of the initial removed face are joined by modifying α_2 relations.

2.2 Removal Operations

Removal operations are the basic operations used during the construction of a minimal map. The removal of an i-cell c (called i-removal of c) leads to the merging of the two *(i+1)*-cells incident to c (see an example of *2*-removal in Fig. 1). For 3D subdivisions, i-removal operations are defined for $i = 0, 1, 2$. A more complete description of the removal operation can be found in [4].

Any face of a 3-G-map can be removed, since the degree of a face in a 3D subdivision is always equal to one or two. The face removal operation consists mainly in locally modify the α_2 relation for each dart that belongs to the neighborhood of the removed face (all removal operations are based on a similar principle).

The 1-removal (edge removal) can only be applied for edges whose degree is one or two. Otherwise it is not possible to automatically decide how to connect the faces incident to the removed edge. This operation is achieved in a similar way than for face removal. In this case, α_1 relations are modified.

The 0-removal (vertex removal) is similar to 1-removal, but α_0 is modified.

2.3 Homology

In this part, basic homology notions are recalled; interested readers can find more details in [5]. The notion of homology can be defined in an algebraic way using the sets of $i-$cells used to describe the subdivision of the surface. Within this context, a *p-chain* (i.e. a chain of dimension p) is defined as a formal sum of p-cells. From this, the group C_p of p-chains is defined. The boundary of a p-chain is defined as the sum of the boundaries of its p-cells. Note that the boundary of a p-chain must be a $p-1$-chain. The set of p-chains which have a null boundary (i.e. *p-cycles*) is a subgroup of C_p (denoted Z_p). The set of p-chains which are

boundary of a $p+1$-chain (i.e. *p-boundaries*) form a subgroup of C_p (denoted B_p).

An essential property is that the boundary of any boundary is null. Hence, every boundary is a cycle and B_p is contained in Z_p. Two p-cycles z_1 and z_2 are *homologous* if their difference is a boundary, i.e. there is a $p+1$-chain f such that $z_1 = z_2 + \partial f$. From this, an equivalence relation can be defined and the homology class of z is the set $\{z+b \mid b \in B_p\}$. The *homology group of dimension* p, denoted H_p, is defined as the quotient group Z_p/B_p, and its elements are the homology classes. For a group G, a *set of generators* is a maximal subset S of elements of G, such that every element of G can uniquely be defined as a linear combination of elements of S.

3 Minimal Map Preserving Homology

3.1 Simplification Algorithm

The goal of Algorithm 1 is to simplify a generalized map M representing a closed surface into its minimal representation (minimal refers to the number of cells) while preserving its homology. Moreover, the set of edges obtained in the minimal map of M is a set of generators of the first homology group of initial surface described by M.

The principle of the algorithm is to remove as much cells as possible while preserving the homology of the initial object. For that, we use a similar approach than the algorithm described in [3]. As mentioned in the previous reference, this general algorithm may stop without a minimal object (i.e. the same object can be represented with less cells). We solve this problem by adding an operation (fictive edge shifting) that ensures that the obtained object is minimal.

This algorithm proceeds in two successive steps. The first one (line 1) removes useless edges. For that, each edge e is successively processed in any order. If e is a degree two edge, then it can be removed without modifying the homology (see the proof in Section 3.3). Indeed this operation consists in merging two distinct adjacent faces in a unique one.

If e is not a degree two edge, there are two distinct cases depending if e is a degree one edge dangling or not dangling. In the first case, e can be removed without modifying the homology. In the second case, e can not be removed. Indeed, its removal will disconnect its incident face in two connected components and this will involve a modification of the homology of the object.

When e is a dangling edge, e needs to be removed (like for degree two edges), but we eventually need to re-consider the edge adjacent to e. Indeed, if some edges form a path, when we consider an edge in the middle of the path, this edge is a not dangling degree one edge and thus it is not removed. But when the current edge is at the end of the path, it is a dangling edge and thus it is removed. But now, the previous edge in the path became a dangling edge and the process can be iteratively applied for each edge of the path. To process this case, when a dangling edge e is removed, we keep the unique edge e' adjacent to e, and e is removed. Now, if e' is a dangling edge, we reiterate the process.

Algorithm 1. Simplification of a surface in its minimal representation

Input: A generalized map M representing a closed surface
Output: The minimal map corresponding to M

```
1 foreach edge e of the map do
    if the degree of e is 2 then
      Remove e;
    else
      while e is a dangling edge do
        e' ← one edge adjacent to e;
        Remove e;    e ← e';
2 foreach vertex v of the map do
    if the degree of v is 2 then
      Remove v;
    else if it exists an edge e non-loop incident to v then
      Shift all edges (except e) incident to v along e;
      Remove v;
```

Otherwise, we are at the end of the path and we can go back to the first loop and consider another edge.

The second step of the simplification algorithm (line 2) removes useless vertices. For that, it runs through each vertex of the surface. If the current vertex is a degree two vertex, it is removed. Indeed, such removal involves to merge its two incident edges. If the degree of the vertex is different from two, a special processing must be applied before its removing. We first take an edge e incident to the current vertex which is not a loop (an edge is a non-loop edge iff it is incident to two distinct vertices). If such an edge does not exist, the subdivision is thus composed uniquely by loops and thus the current vertex is the unique vertex of the subdivision. This case occurs only for the last vertex of the map when the process is finished and the minimal map is obtained. Otherwise, all the edges incident to the current vertex are shifted along e. The edge shifting operation consists in pushing an edge along another one (see Fig. 4). After this operation, since all edges incident to the current vertex are pushed, the vertex degree is now equal to one and the vertex can be removed without modifying the homology of the object.

We can see in Fig. 2 and Fig. 3 two examples: the first one with an initial surface representing a double-torus and the second one with an initial surface representing a Klein bottle. For these examples, we have represented in (A) the initial subdivision, in (B) the map obtained after the first simplification step and in (C) the final map which is minimal.

3.2 Complexity

The complexity of Algorithm 1 is equal to $O((3-\chi)\times n)$ where n is the number of cells and χ is the Euler characteristic of the surface.

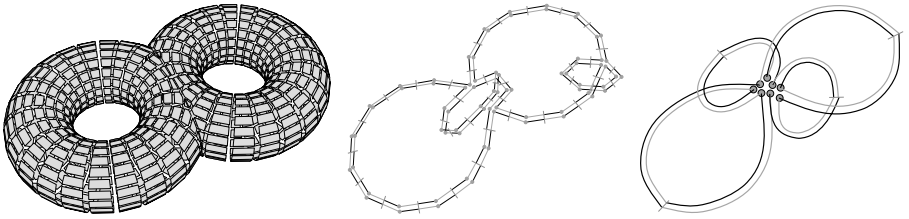


Fig. 2. An example of minimal map corresponding to a double torus. (A) The initial surface. (B) After edge removals. (C) After vertex removals.

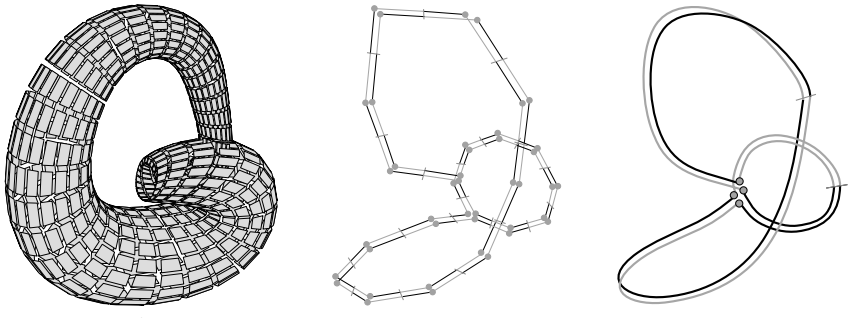


Fig. 3. An example of minimal map corresponding to a Klein bottle. (A) The initial surface. (B) After edge removals. (C) After vertex removals.

Each edge is processed exactly once during the first loop of the algorithm. The test on the edge degree can be achieved in two steps. First locally, in order to exclude all the edges with degree strictly greater than two (by testing if $\alpha_{23}(d) \neq \alpha_{32}(d)$). Then, there are two distinct cases to distinguish depending if the edge is a degree one or two edge. Indeed, the edge of the first case has to be removed while the edge of the second case does not. But these two cases can not be characterized locally since we need to know if both side of the edge belong to the same face.

In order to solve this problem, we use union-find trees [6] which allow to represent efficiently disjointed sets. This structure is handled by two operations: *find* which returns, given an element, the representative of the set, and *union* which allows to merge two sets. The amortized cost of a series of m union-find operations on n elements can be done in time $O(n.\alpha(m,n))$ with $\alpha(m,n)$ being the inverse Ackermann function which grows extremely very slowly, and which is less than 5 in practical cases (see [6] for the demonstration about the complexities).

We link each dart of a same face with an union-find tree representing the face. When we remove an edge, we merge both corresponding trees by using the *union*

operation. The test if d and $\alpha_2(d)$ belong to the same face is simply achieved by testing if *find(d)* is equal to *find*($\alpha_2(d)$).

To summarize the first step, the cost of the test on the edge degree can be bounded by 5, the cost of the test if the edge is a dangling edge is in constant time and the edge removal is also achieved in $O(1)$.

For the second loop of Algorithm 1 "*While e is a dangling edge*", in the worst case, we can run through all the edges of the map. This case case occurs when the map is a unique path, and if the first loop start with a dart in the middle of the path. In this case, each edge is considered only twice. The first time is during the first loop, without removing it since each edge in the middle of the path is a degree one edge and not a dangling one. The second time is during the second loop, after we have processed an edge at the extremities of the path. Since each edge of the second loop is removed, it will never be considered once again.

The second step of the simplification algorithm considers each vertex exactly once. The test on the degree of v is achieved by running through all the darts of the vertex. Each dart is only considered once. Moreover, the vertex removal operation is also achieved in linear time according to the number of darts of the removed cell.

The edge shifting operation is achieved linearly in number of edges incident to the removed vertex. The subdivision in which we simplify vertices is the result of the edge simplification. Thus, this subdivision is always composed of one face and n_l loops, where n_l is equal to $2 - \chi$ with χ being the Euler characteristic of the surface (indeed, the minimal map has 1 vertex, 1 face and n_l edges and we can compute n_l by using the Euler formula).

Thus, given a vertex v, there are at most $2 - \chi$ edges incident to v and thus the complexity of the second step of Algorithm 1 is in $O((2-\chi) \times n)$. By adding the cost of the first step $O(n)$, we obtain the final complexity $O((3 - \chi) \times n)$ (with $\chi \leq 2$).

3.3 Proof of the Algorithm

In order to ensure the validity of our algorithm, we prove that the obtained map is minimal and the homology is preserved during the simplification process.

It can be proven that the resulting map is minimal by contradiction. The initial map represents a closed surface. The map obtained after the first step of our algorithm is made of an unique face. Indeed, otherwise there are at least two faces and the edge between these two faces is a degree two edge. This contradicts the simplification algorithm which removes each degree two edge. Moreover, this map does not contain dangling edge since they are all removed during the simplification.

The map obtained after the second step of our algorithm is made of an unique vertex. Indeed, otherwise, there are at least two vertices, and since the map is connected, one edge non-loop between these two vertices. This contradicts the fact that in our algorithm, we remove each vertex after to have shifted all edges incident to it.

This proves that the resulting map can is minimal in number of faces and vertices. This map is thus obviously minimal in number of edges since, given a number of faces and vertices, the number of edges is fixed depending on the Euler characteristic of the surface.

In order to prove that the simplification process preserves the homology, we use the works of [3] and make a parallel between their elementary reductions and our simplification algorithm. In this work, the authors present their algorithm starting with a simplicial complex, and simplify it by using *interior face reduction*. Our removal operation is equivalent to this reduction which consists in taking a common i-face c of exactly two $(i+1)$-simplices a and b, and which deletes c and replaces b and c by a cell which represents their union. It is proven that interior face reduction preserves homology (see [3]) and thus we can conclude that removal operations too.

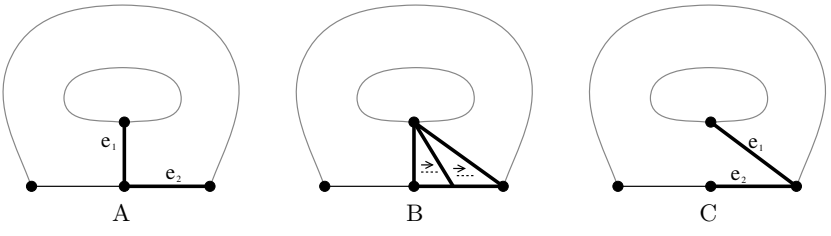


Fig. 4. An example of edge shifting. (A) The initial configuration. We are going to shift edge e_1 along edge e_2. (B) Edge shifting seen as a continuous mapping. (C) Final configuration.

In this work, we use an additional operation, edge shifting, in order to obtain a minimal representation. Proving that this operation preserves homology can easily be done by considering this operation as a continuous mapping between the initial configuration (e.g. Fig. 4A) and the configuration obtain after the edge shifting (e.g. Fig. 4C).

The edges of the resulting minimal map, are homology groups generators: they are cycles because they are self-loops and then their border is zero, and each edge can not be bordered because in this case, the map would have at least two faces. Moreover, these edges are not two by two homologous: if two edges e_1 and e_2 are homologous, then their exist a border ∂f of (at least) a face such that $e_1 = e_2 + \partial f$. This is not possible as we only consider minimal map of closed surfaces.

Moreover, in our case, the combinatorial structure used (i.e. generalized maps) allows us to distinguish between free generators (generators of free groups of the homology type) and torsion generators (generators of torsion groups of the homology type). This is due to the manner of the darts are proceeded when we run through the $< \alpha_0 \circ \alpha_1 >$ orbit (black darts on figures 2 and 3). Let d be a dart of a given edge, if $\alpha_2(d)$ is in the same orbit then the edge is a torsion generator, if not then the edge is a free generator. This way to recognize

torsion generators is valid because we are in the case of quasi-manifolds and then torsion coefficients cannot be more than 2. Fig. 3C shows example of free and torsion generators. It can also be noted that if we count free generators in each dimension, we obtain the Betti numbers of the surface.

4 Application for 3D Images

Given a 3D labeled image, it is possible to compute the minimal map which represents all the regions contained in the image and all the incidence and adjacency relations between these regions [7,8]. It has been shown in many works (for example [9,10,11,12,13]) that using structures based on combinatorial (or generalized) maps gives an efficient framework for image processing algorithms. In these algorithms, it is often necessary to compute some features on regions, either to characterize them for example in an recognition process, or to use these features during a segmentation process.

We can use our simplification algorithm in order to compute homology generators of each region contained in a 3D labeled image. This is particularly interesting in order to characterize these regions by their boundaries. For that, we start from the minimal map which represents all the regions of the image (called topological map, see [14,12] for a more detailed description). Since our algorithm works only for one surface (for the moment), we can not compute directly the generators on the topological map since it represents several surfaces.

In order to characterize a given region R, we first extract its surfaces. This is achieved easily by running through the topological map and extracting only surfaces belonging to R. This can be achieved with a breadth-first search algorithm, so linearly in number of darts of the surface. Region R can have one or several surfaces. Indeed, each region is always represented by one external boundary, but it can also be represented by some internal boundaries if a region has some included regions. But since these surfaces are disconnected, we can process each of them by our simplification algorithm and thus obtain generators for each surface.

5 Conclusion

In this paper we have presented an algorithm for computing the minimal map of a generalized map representing a subdivision of a closed surface. We have shown that a set of homology generators (free or torsion ones) can directly be deduced from the minimal map.

We have proven that the minimal map algorithm, based on the general algorithm proposed by Kaczynski et al. preserves the homology of an object represented by a G-map. Moreover, the complexity of the minimal map algorithm is $O((3 - \chi) \times n)$, where n the number of cells and χ is the Euler characteristic of the surface. An application of these results have been proposed for 3D images: characterizing a region of voxels by a set of generators of the first homology group of its boundary.

In future works, we plan to extend the minimal map algorithm for G-maps with boundaries and for computing a minimal map that preserves homology for higher dimension objects.

References

1. Lienhardt, P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. Commputer Aided Design **23** (1991)
2. Lienhardt, P.: N-dimensional generalized combinatorial maps and cellular quasi-manifolds. International Journal of Computational Geometry and Applications **4** (1994) 275–324
3. Kaczynski, T., Mrozek, M., Slusarek, M.: Homology computation by reduction of chain complexes. Computers & Math. Appl. **34** (1998) 59–70
4. Damiand, G., Lienhardt, P.: Removal and contraction for n-dimensional generalized maps. In: Discrete Geometry for Computer Imagery. Number 2886 in Lecture Notes in Computer Science, Naples, Italy (2003) 408–419
5. Hatcher, A.: Algebraic Topology. Cambridge University Press (2002) available on http://www.math.cornell.edu/∼hatcher/AT/ATpage.html.
6. Tarjan, R.: Efficiency of a good but not linear set union algorithm. Journal of the ACM **22** (1975) 215–225
7. Braquelaire, J.P., Desbarats, P., Domenger, J.P., Wüthrich, C.: A topological structuring for aggregates of 3d discrete objects. In: Workshop on Graph-Based Representations in Pattern Recognition, Austria, IAPR-TC15 (1999) 193–202
8. Bertrand, Y., Damiand, G., Fiorio, C.: Topological map: Minimal encoding of 3d segmented images. In: Workshop on Graph-Based Representations in Pattern Recognition, Ischia, Italy, IAPR-TC15 (2001) 64–73
9. Domenger, J.: Conception et implémentation du noyeau graphique d'un environnement 2D1/2 d'édition d'images discrètes. Thèse de doctorat, Université Bordeaux I (1992)
10. Fiorio, C.: A topologically consistent representation for image analysis: the frontiers topological graph. In: Discrete Geometry for Computer Imagery. Number 1176 in Lecture Notes in Computer Science, Lyon, France (1996) 151–162
11. Braquelaire, J.P., Brun, L.: Image segmentation with topological maps and inter-pixel representation. Journal of Visual Communication and Image Representation **9** (1998) 62–79
12. Damiand, G., Resch, P.: Topological map based algorithms for 3d image segmentation. In: Discrete Geometry for Computer Imagery. Number 2301 in Lecture Notes in Computer Science, Bordeaux, France (2002) 220–231
13. Bourdon, P., Alata, O., Damiand, G., Olivier, C., Bertrand, Y.: Geometrical and topological informations for image segmentation with monte carlo markov chain implementation. In: Vision Interface, Calgary, Canada (2002) 413–420
14. Bertrand, Y., Damiand, G., Fiorio, C.: Topological encoding of 3d segmented images. In: Discrete Geometry for Computer Imagery. Number 1953 in Lecture Notes in Computer Science, Uppsala, Sweden (2000) 311–324

Dynamic Texture Analysis and Synthesis Using Tensor Decomposition

Roberto Costantini, Luciano Sbaiz, and Sabine Süsstrunk

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

Abstract. Dynamic textures are sequences of images showing temporal regularity, such as smoke, flames, flowing water, or moving grass. Despite being a multidimensional signal, existing models reshape the dynamic texture into a 2D signal for analysis. In this article, we propose to directly decompose the multidimensional (tensor) signal, free from reshaping operations. We show that decomposition techniques originally applied to study psychometric or chemometric data can be used for this purpose. Since spatial, time, and color information are analyzed at the same time, such techniques permit to obtain more compact models. Only one third or less model coefficients are needed for the same quality and synthesis cost of 2D based models, as illustrated by experiments on real dynamic textures.

1 Introduction

Two approaches exist to synthesize static and dynamic textures starting from an image or video sample. The first approach is non-parametric and consists in collecting different parts of the texture. In the case of static textures, these parts are patches taken from the original texture sample, while for dynamic textures they are short sequences of consecutive frames of the video. In both cases, these parts are "fused" together during synthesis such that transitions between them are not noticeable. Historically, this approach was first applied to static textures [1,2,3] and then to dynamic textures [4,5], obtaining very realistic effects.

The second approach is parametric and consists in representing the texture according to a given model. The model parameters are estimated in the analysis step and used during the synthesis. Static textures are commonly analyzed using spatial filters, such as Laplacian, steerable pyramids, wavelets, etc. The aim is to extract salient features from the texture, thus mimicking the human visual system's ability to perceive spatial information. The synthesis is done by imposing that original and synthesized textures have similar histograms in the filter domain. This is commonly done using an iterative process [6,7,8]. An other approach consists in estimating autoregressive (AR) spatial models from the texture [9].

Dynamic textures are analyzed in a similar ways, but the temporal dimension. Spatio-temporal AR identification techniques [10] or spatio-temporal filtering

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 245–254, 2006.
© Springer-Verlag Berlin Heidelberg 2006

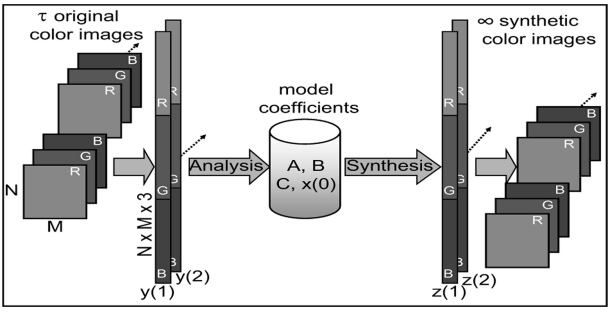


Fig. 1. Schematic view of the model based analysis/synthesis process

operations (3D wavelet [11] or Laplacian of Gaussian filters [12]) are used to build the model. The advantage of parametric approaches is that they are useful for other applications than synthesis, such as recognition and segmentation.

An interesting parametric approach is the *linear model* of Doretto et al. [13]. The dynamic texture is considered as a dynamical system, whose properties are modelled using methods borrowed from the system identification community. The analysis consists in representing each image as a point in a given subspace and in identifying its trajectory. After a validation step, where the model order is chosen, the synthesis is obtained by driving the system with white noise. The subspace is determined by the Singular Value Decomposition (SVD) of the video representing the dynamic texture, and the point trajectory is described using a multivariate auto-regressive (AR) process.

Fig. 1 shows the analysis-synthesis approach of the model. A sequence of τ color images of size $N \times M$ is reshaped into a sequence of column vectors of length $m = N \times M \times 3$. They are the input of the analysis algorithm, which computes the model parameters. Synthesis follows accordingly to the formula:

$$\begin{cases} x[k+1] = Ax[k] + B\nu[k] \\ z[k] = Cx[k] \end{cases} \tag{1}$$

where $z \in \mathbb{R}^m$ is the vector representing a synthesized color image, $x \in \mathbb{R}^n$ is the system state vector, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_v}$, and $C \in \mathbb{R}^{m \times n}$ are the model parameters, and $\nu \in \mathbb{R}^{n_\nu}$ is a Gaussian random vector $\mathcal{N}(0, I_{n_v \times n_v})$. The index k indicating the frame index is also referred to as the time index.

The n columns of matrix C contain the basis of the subspace onto which image vectors y are projected. The subspace is obtained considering the first $n < \tau$ singular values of the matrix $Y_1^\tau = [y(1), \ldots, y(\tau)] \in \mathbb{R}^{m \times \tau}$ that collects the original images; n is called the *order* of the system.

While a linear model is simple, powerful, and permits to describe, modify, and recognize dynamic textures in an efficient way [14,15,16,17], it requires a huge amount of coefficients. A solution was proposed in [18], where the SVD is computed on the FFT-transformed video frames, considering only the most significant coefficients. This permits to reduce the number of model coefficients,

but requires the computation of an inverse FFT transform for the synthesis of each frame. Even though this can be done efficiently on a PC, it might be too computationally expensive on a portable device. A different solution is given in [19], where it is proposed to use YC_bC_r instead of RGB color encoding. Chrominance channels can be downsampled both horizontally and vertically, leading to a model with the same synthesis quality but with half the number of coefficients and less computational complexity.

In this article, we propose to obtain a more compact model by directly decomposing the multidimensional matrix (tensor) that represents the dynamic texture without reshaping the video frames onto column vectors. Multidimensional data decomposition permits in fact to jointly consider spatial, temporal, and color pixel correlation, following the same direction of [18] and yet free from the burden of the inverse Fourier Transform for the synthesis of each frame. Moreover, our model can also exploit color information, where [18] considers grayscale images only.

Section 2 decribes the analysis using the SVD liner model, Section 3 introduces the tensor decomposition. Section 4 and 5 illustrate the way data are organized in a tensor and the corresponding model dimensions. Section 6 shows the cost of the synthesis. In Section 7, the linear model based on the tensor decomposition is compared to the SVD linear model. Finally, Section 8 concludes the paper.

2 Dynamic Texture Analysis: Two-Way SVD

In the linear model of Fig. 1, the analysis consists in decomposing the matrix $Y_1^\tau = [y(1), \ldots, y(\tau)] \in \mathbb{R}^{m \times \tau}$ using SVD[1], as shown in Fig. 2(a). We refer to this decomposition as *two-way SVD*.

We consider a n-order model defined by the matrices U_n, Σ_n, and V_n^T that collect the first n components of the SVD in the way indicated by the dashed lines in Fig. 2. The model parameters of Eq.(1) are the matrices C and A defined as:

$$\begin{cases} C = U_n \\ A = \Sigma_n [V_2^\tau (V_1^{\tau -1})^{-1}])^T \end{cases} \tag{2}$$

where $V_1^{\tau -1}$ and V_2^τ respectively indicate the first and the last $\tau - 1$ rows of V_n^T, $(\cdot)^{-1}$ the pseudo-matrix inversion, and T the transpose operator. Eq.(2) permits to identify the AR process of order one used to describe the texture dynamic (see [13] for details). The SVD is used to find the best rank-n approximation of Y_1^τ. Borrowing a terminology often used in the life-science community, $U_n \in \mathbb{R}^{m \times n}$ is the *score* matrix and V_n^T the *loading* matrix.

3 Dynamic Texture Analysis: Tensor Decomposition

In several cases it is more effective to decompose a multidimensional signal as it is collected, i.e., avoiding reshaping operations. Recently, the signal processing

[1] We suppose that the temporal mean of each pixel has been previously subtracted from Y_1^τ.

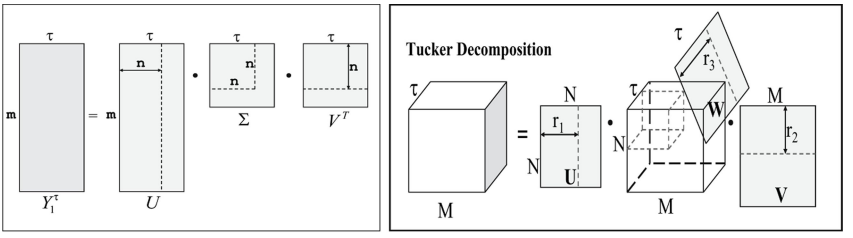


Fig. 2. (a) Singular Value Decomposition of the matrix Y_1^τ that contains the dynamic texture frames reshaped along column vectors. The dashed lines indicate the row/column truncation that leads to matrices U_n, Σ_n, and V_n^T; (b) The Tucker decomposition of a 3D tensor. Score matrix U and load matrices V and W are shown. The dashed lines indicate that an approximation of the tensor is obtained by retaining only some rows/columns of the score and load matrices.

community has started to study data dimension reduction techniques that work on multidimensional data [20,21]. Originally developed for the analysis of experimental data in psychology, chemometrics [22,23] and other fields, such methods are called multilinear, multi-way, higher-order, or tensor decompositions. In this article we consider the tensor decomposition depicted schematically in Fig. 2(b), and known as the Tucker decomposition.

The Tucker decomposition [22,23] (or *three-way* decomposition, since it originally was intended for the analysis of 3D data sets) corresponds to a generalization of the SVD to multiple dimensions. We follow the notation in [20], where this transform is defined as a generalization of the SVD. Given a $(I_1 \times I_2 \times \ldots \times I_n)$-tensor $\mathcal{H}$, the decomposition is defined as:

$$\mathcal{H} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_n U^{(n)} \tag{3}$$

where $\mathcal{S}$ is a *core* tensor (it plays the same role of the matrix Σ of the two-way SVD), $U^{(1)}, U^{(2)}, \ldots, U^{(n)}$ are unitary $(I_i \times I_i)$-matrices with $i = 1, 2, \ldots, n$, and the product $\times_i$ indicates the i-mode product between the tensor $\mathcal{S}$ and matrix $U^{(i)}$ (see [20] for comprehensive notations and definitions).

If we denote as $H_{(i)}$ the two-dimensional matrix obtained by unfolding tensor $\mathcal{H}$ along dimension i, the matrices $U^{(i)}$ of Eq.(3) derive from the two-way SVD of $H_{(i)}$: $H_{(i)} = U^{(i)} \Sigma^{(i)} V^{(i)^T}$. The core tensor $\mathcal{S}$ is computed by inverting Eq.(3): $\mathcal{S} = \mathcal{H} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \ldots \times_n U^{(n)^T}$. Incidentally, the two-way SVD is a particular case of Eq.(3): $Y_1^\tau = \Sigma \times_1 U \times_2 V$.

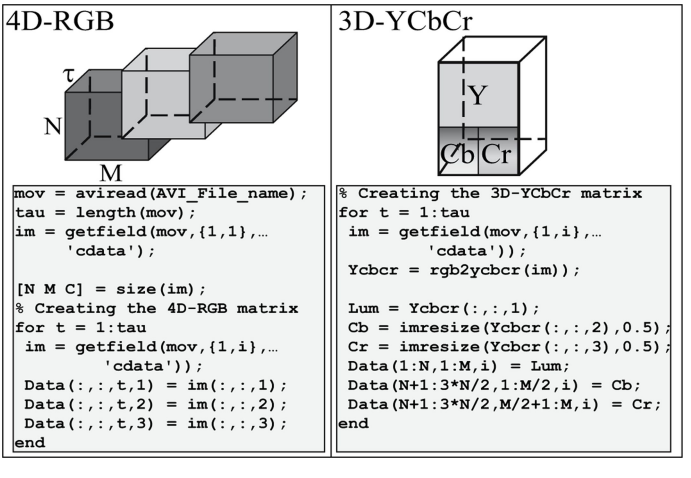
As opposed to the two-dimensional SVD, the core matrix is not diagonal, but has the property of being *all-orthogonal*, i.e. the scalar product of non-identical subtensors is null [20]. Another difference from the 2D case is that the the Tucker

decomposition obtained by truncating the first $r_1 < I_1, r_2 < I_2, \ldots, r_n < I_n$ columns of the matrices $U^{(i)}$ does not produce the best rank-$(r_1 < I_1, r_2 < I_2, \ldots, r_n < I_n)$ approximation of $\mathcal{H}$. An iterative process is necessary to find it [21].

Once the data dimension reduction is obtained from the Tucker decomposition, the computation of the matrix A of Eq.(1) is done similarly to the 2D case. Considering for instance the 3D case depicted in Fig. 2(b), A is computed from Eq.(2), exchanging V^T with W and applying Eq.(2) as done in the previous Section.

4 Data Organization

Several ways exist to create a tensor from the color video sequence that represents a dynamic texture. In Fig. 3 we illustrate the two representations that we have adopted. At left, data is organized in a 4D tensor according to horizontal, vertical, temporal, and chromatic dimensions, respectively. We depicted the three color components separately, since it is not possible to draw a 4D tensor in a picture. At right, YC_bC_r color encoding is used to compact a color image into a single plane that contains luminance and chrominance channels. Chrominance is downsampled by a factor of two both horizontally and vertically. Fig. 3 reports also the MATLAB code used to create the data structure. We organized 4D-RGB and 3D-YC_bC_r data such that the third component always corresponds to the temporal dimension.




```
mov = aviread(AVI_File_name);
tau = length(mov);
im = getfield(mov,{1,1},…
     'cdata');

[N M C] = size(im);
% Creating the 4D-RGB matrix
for t = 1:tau
 im = getfield(mov,{1,i},…
         'cdata'));
 Data(:,:,t,1) = im(:,:,1);
 Data(:,:,t,2) = im(:,:,2);
 Data(:,:,t,3) = im(:,:,3);
end
```

```
% Creating the 3D-YCbCr matrix
for t = 1:tau
 im = getfield(mov,{1,i},…
         'cdata'));
 Ycbcr = rgb2ycbcr(im));

 Lum = Ycbcr(:,:,1);
 Cb = imresize(Ycbcr(:,:,2),0.5);
 Cr = imresize(Ycbcr(:,:,3),0.5);
 Data(1:N,1:M,i) = Lum;
 Data(N+1:3*N/2,1:M/2,i) = Cb;
 Data(N+1:3*N/2,M/2+1:M,i) = Cr;
end
```

Fig. 3. Data organization. Left: data is organized in a 4D tensor where the first two dimensions are space, the third dimension is time, and the fourth dimension is color. Right: data is organized in a 3D tensor where the first two dimensions are spatial and color information, while the third is time.

5 Model Size

Tab. 1 resumes the number of model coefficients. They are obtained by retaining r_1 and r_2 singular values to represent the spatial dimensions, n for time, and 3 for color in the tensor decomposition. We omitted the coefficients needed to represent model matrices A and B, since they do not depend on the decomposition and are considerably smaller compared to the number of the other coefficients. Note that the two-way SVD has always the higher cost. In fact, the dynamic texture dimension N, M, and the order n of the model are multiplied, thus producing a large number of coefficients. Conversely, the number of coefficient in the Tucker decompositions is the product of smaller factors, since $r_1 r_2 \ll MN$.

Table 1. Number of model coefficients

Data Representation	4D-RGB	3D-YC_bC_r
Two-way SVD	$3MNn$	$1.5MNn$
Tucker	$Nr_1 + Mr_2 + n\tau + 3r_1r_2n$	$1.5Nr_1 + Mr_2 + n\tau + r_1r_2n$

6 Synthesis Cost

Synthesis is done according to Eq.(1): at each time index k the state vector $x[k]$ is updated and a new frame $z[k]$ is computed. The cost of updating the state vector $x[k]$ is $2n(n + n_\nu)$ operations[2], which corresponds to the cost of computing $Ax[k]$ and $B\nu[k]$. This cost is independent of the decomposition used in the analysis and it is not taken into consideration here, since it is irrelevant when compared to the cost of computing $z[k]$.

The computation of $z[k]$ in the case of a two-way decomposition corresponds to $3MNn$ operations., i.e., the cost of computing $Cx[k]$. In the case of tensor decomposition, the cost is given by the cost of the tensor multiplication:

$$\mathcal{S} \times_1 U^{(1)} \times_2 \ldots x[k] \ldots \times_L U^{(L)}. \tag{4}$$

The cost of this product is $I_1 I_2 \ldots I_n$, i.e., the product of the dimension of the core tensor. In the 4D-RGB case, the cost is thus $3MNn$, while for 3D-YC_bC_r the cost is $1.5MNn$ operations. Note that the cost of the synthesis using tensor decomposition is thus equal to the cost of synthesis of the two-way case.

7 Performance Evaluation

We evaluated the performance of the tensor decomposition in terms of prediction error. This is defined as the average PSNR between the frames of the original dynamic texture and their prediction obtained from the model. Calling W the

[2] With "operations" we mean simultaneous multiplications and additions.

load matrix of the tensor decomposition along the time axis (same notation as in Fig. 2(b)), the prediction $\hat{z}[k]$ of the k-th frame is given by:

$$\hat{z}[k] = \mathcal{S} \times_1 U^{(1)} \times_2 \ldots \hat{x}[k] \ldots \times_L U^{(L)}, \tag{5}$$

where $L = 4$ in the case of 4D-RGB data representation, $L = 3$ for 3D-YC_bC_r and $L = 2$ for the two-way case. The vector $\hat{x}[k] = Aw_{k-1}$ is the state vector prediction, where A is the matrix estimated from W according to Eq.(2) and w_{k-1} is the $(k-1)$-th column of matrix W. The tensor $\mathcal{S}$ is the core tensor and it reduces to Σ in the case of the two-way decomposition.

As test sequences, we considered 4 dynamic texture videos, respectively called "Flame," "Grass," "River," and "Sparkle." Fig. 4 shows the comparison between the performance obtained using the two-way and the Tucker decompositions on those video, indicating the name and respective dimension.

For the Tucker decomposition, we set $r_1 = r_2$. Thus, we performed a spatial dimension reduction using the same number of singular values for the horizontal and vertical spatial content respectively. r_1 was varied from 10 to $\min(N, M)$, where N and M are respectively the vertical and horizontal image dimensions. The dimension n of the state vector was varied from 10 to τ (the length of the video sequence) with an interval of 10, both for Tucker and two-way decompositions.

As illustrated in Fig. 4, the two-way SVD decomposition requires a greater number of coefficients for both (RGB and YC_bC_r) data representation than the Tucker method. The dashed curves corresponding to the Tucker decomposition, always lie to the left of the curves corresponding to the two-way decomposition. Using Tucker decomposition, we thus need less coefficients to obtain a model with the same performance in terms of prediction (PSNR). Interestingly, the two-way SVD decomposition of YC_bC_r data also needs less coefficients than both RGB representations for the same PSNR [19].

In Tab. 2 we compare the different representations by listing the relative number of coefficients necessary to have a PSNR of approximately about 30dB. The numbers refer to the Tucker decomposition of the 3D-YC_bC_r data, which is the representation that required the least number of coefficient. Note that the other decompositions require 1.3 to 5.4. times the number of coefficients.

While PSNR is an often used tool to evaluate the performance of different analysis and synthesis algorithms, it does not necessarily equate to a measure of visual quality. We compared the "naturalness" of the different synthesized videos in the lab and found them to approximately correspond to the results found above.[3]

We considered the performances on different image size to show that results can change according to image content and image size. The smaller the images, the less we gain comes from a multidimensional decomposition. Moreover, we limited the model order because the time and memory requirement necessary to compute higher order decompositions exceeded our computational capabilities. Despite having the same computational complexity in the synthesis step, tensor decomposition is in fact much more expensive than two-way decomposition in

[3] Synthesis results can be found at http://lcavwww.epfl.ch/~costanti/ISVC06.htm

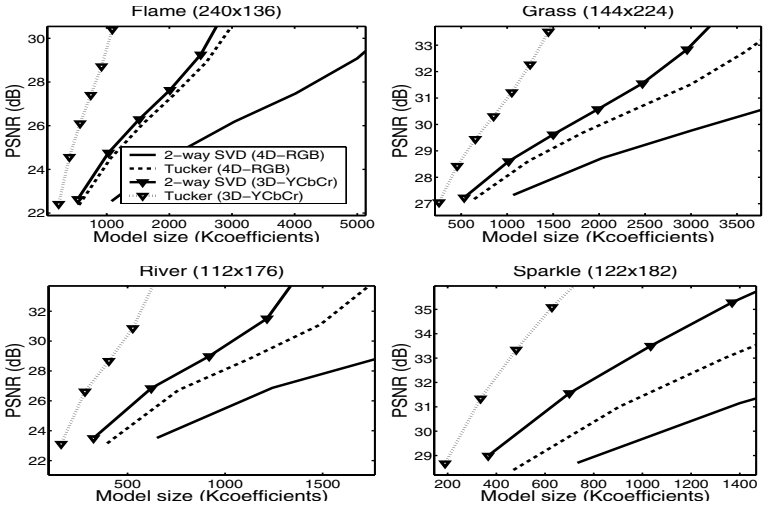


Fig. 4. Performance comparison for the 4 test video sequences

Table 2. Relative number of coefficients of the model that ensure a PSNR level of abotu 30dB. The reference decomposition, whose value is set to 1, is the Tucker on 3D-YC_bC_r data.

	Flame		**Grass**		**Sparkle**		**River**	
	240x136	120x68	144x224	72x112	122x182	80x120	112x176	56x88
Tucker (YC_bC_r)	1	1	1	1	1	1	1	1
Two-way (YC_bC_r)	2.6	2.1	2.2	1.2	1.9	1.8	2.1	1.3
Tucker (RGB)	2.9	2.2	2.7	2	2.9	2.8	2.7	2.3
Two-way (RGB)	5.4	3.4	4.3	2.4	4.1	3.9	4.1	2.7

the analysis step. We did not consider here any data sampling techniques [24] that could potentially reduce the computation of the tensor decomposition.

In Fig. 5 we show some prediction frames obtained using the Tucker decomposition of the 3D-YC_bC_r data representation.

8 Conclusions

We showed that the linear model based on tensor decomposition ensures the same dynamic texture synthesis quality at the same cost than the one based on two-way SVD decomposition, but resulting in a much smaller number of model coefficients. Using Tucker decomposition, we obtain a model with only about one third of the coefficients compared to the two-way SVD model. The low number of coefficients allows applying such multidimensional decomposition techniques in dynamic texture synthesis applications where memory constraints play an

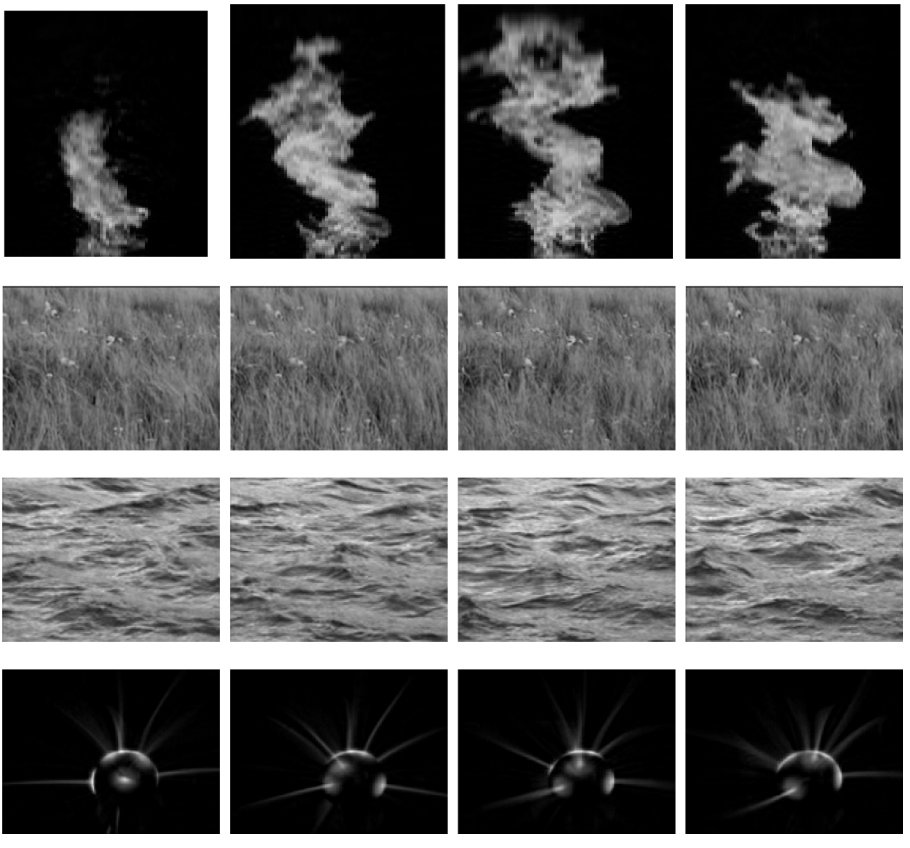


Fig. 5. Predicted dynamic textures frames, at a PSNR of about 30dB. First row: "Flame"; Second row: "Grass"; Third row: "River"; Fourth row: "Sparkle"

important role, such as in portable devices. One limitation of these methods is the analysis cost, but this can be decreased using data sampling techniques to find an approximate solution to the tensor decomposition.

Acknowledgments

This project is supported by the Swiss National Science Foundation (SNF) under grant number 21-067012.01. The authors thank Cristian Carmeli and Joos Vandewalle for fruitful discussions on the use of tensor decomposition.

References

1. Xu, Y., Guo, B., Shum, H.: Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report, Microsoft Research **22** (2000)
2. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. (Proceedings of SIGGRAPH 2000) 479–488

3. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. (Proceedings of SIGGRAPH 2001) 341–346
4. Schödl, A., Szeliski, R., Salesin, D., Essa, I.: Video textures. (Proc. of SIGGRAPH 2000) 489–498
5. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. (Proc. of SIGGRAPH 2003) 277–286
6. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. (ACM Transactions on Graphics, SIGGRAPH 1995) 229–238
7. Zhu, S.C., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. Int. Journal of Comp. Vis. **27** (1998) 107–126
8. Portilla, J., Simoncelli, E.: A parametric model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision **40** (2000) 49–71
9. Alata, O., Olivier, C.: Choice of a 2-d causal autoregressive texture model using information criteria. Pattern Recognition Letters **29** (2003) 1191–1201
10. Szummer, M., Picard, R.: Temporal texture modeling. IEEE Int. Conf. on Image Process. (1996) 823–826
11. Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Texture mixing and texture movie synthesis using statistical learning. IEEE Trans. on Visualization and Computer Graphics **7** (1993) 120
12. Wang, Y., Zhu, S.: A generative method for textured motion: Analysis and synthesis. Proc. of ECCV'02 **1** (2002) 583–597
13. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. Int. Journal of Computer Vision **51** (2003) 91–109
14. Doretto, G., Soatto, S.: Editable dynamic textures. In: CVPR '03. Volume 2. (2003) 137–142
15. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR '01. Volume 2. (2001) 58–63
16. Yuan, L., Wen, F., Liu, C., Shum, H.Y.: Synthesizing dynamic texture with closed-loop linear dynamic system. European Conf. on Comp. Vision, 2004 (2004)
17. Liu, C.B., Lin, R.S., Ahuja, N.: Modeling dynamical textures using subspace mixtures. (ICIME 2005)
18. Abraham, B., Camps, O.I., Sznaier, M.: Dynamic texture with fourier descriptors. Proc. of the 4th Int. Workshop on Texture Analysis ans Synthesis (2005) 53–58
19. Costantini, R., Sbaiz, L., Süsstrunk, S.: Dynamic texture synthesis: Compact models based on luminance-chrominance color representation. (To appear in Proc. ICIP 2006, Atlanta, GA, USA)
20. Lathauwen, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. Siam J. Matrix Anal. Appl. **21** (2000) 1253–1278
21. Lathauwen, L.D., Moor, B.D., Vandewalle, J.: On the best rank-1 and rank-(r_1,r_2, ..., r_1) approximation of higher-order tensors. Siam J. Matrix Anal. Appl. **21** (2000) 1324–1342
22. Tucker, L.: Some mathematical notes on three-mode factor analysis. Psychometrika **31** (1966) 279–311
23. Tucker, L.: Relations between multidimensional scaling and three-mode factor analysis. Psychometrika **37** (1972) 3–27
24. Mahoney, M.W., Maggioni, M., Drineas, P.: (Tensor-cur decompositions for tensor-based data) Proc. 12-th Annual SIGKDD 2006.

Color Pair Clustering for Texture Detection*

Lech Szumilas and Allan Hanbury

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9-11/1832, A-1040 Vienna, Austria
{lech, hanbury}@prip.tuwien.ac.at

Abstract. A novel approach to the extraction of image regions of uniform color and its application to automatic texture detection is discussed. The method searches for alternating color patterns, through hierarchical clustering of color pairs from adjacent image regions. The final result is a hierarchy of texture regions, described by their boundaries and a set of features, detected at multiple accuracy levels. The results are presented on some images of natural scenes from the Berkeley segmentation dataset and benchmark.

1 Introduction

Automatic texture detection can play an important role in many image analysis tasks, such as image segmentation or object recognition.

One of the most fundamental properties appearing in many textures is the presence of some regularity in a contiguous image region [1], which may manifest itself as a spatially repeating color pattern or shape. It is however not possible to find a strict regularity criterion which would discriminate between textures and non-textures. Several segmentation algorithms, either automatic or semi-automatic, aim at dividing an image into uniform color and/or texture regions [2,3,4,5,6,7]. The primary problem that these methods face is related to a fixed criterion for texture discrimination (usually texton based). In reality however, the texture similarity depends on the particular application or associated knowledge (see Figure 1).

The *Feature Co-occurrence Texture Detector* (FCTD) proposed in this paper detects potential texture regions at various "regularity levels", which can be later used for texture classification. The advantage of such an approach is the ability to detect less or more regular patterns automatically and then to make a knowledge based selection.

A very common pattern found in textures is the alternation of two or more colors or luminance levels. Therefore the FCTD at present uses only color features to do a coarse texture detection. We assume that the texture consists of spatially alternating patches of uniform color and simple shape, which we call *texture elements*. The general idea is to segment the image into small regions with a relatively uniform color and then to find alternating color patterns among those segments. The pattern we are looking

* This work was supported by the European Union Network of Excellence MUSCLE (FP6-507752) and the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04).

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 255–264, 2006.
© Springer-Verlag Berlin Heidelberg 2006

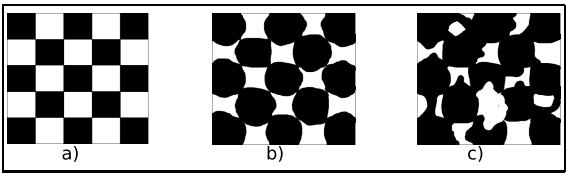


Fig. 1. All three images above may be classified as different or identical textures depending on the texture discrimination criteria. If only the color pattern is considered then all textures are identical, however when the geometrical pattern is also considered then these textures become different.

for is a group of similar color segments neighboring with another group of segments (or potentially more groups) with different colors. The result of the FCTD is a set of textures for which boundaries and color patterns are known.

We also present a novel image segmentation method, the Dominant Color Image Segmentator (DCIS), which aims at extracting uniform color regions in the image. The FCTD method uses the DCIS segmentation to extract color features.

The following sections describe all the steps leading to detection of texture regions: image segmentation (Section 2), feature extraction (Section 3), feature clustering (Section 4) and texture detection (Section 5).

2 Image Segmentation

The primary goal of the image segmentation step is to generate uniform color regions approximating the texture elements, which will be used for extracting color pairs from adjacent regions. We do not aim at obtaining a perfect representation of the texture elements, which would not be possible. We nevertheless want to avoid too much over-segmentation, which could lead to a region of almost uniform color being internally divided into more than one segment. Such an over-segmentation would destroy the texture element adjacency information that we use.

Several well known methods, which perform a general color segmentation of the image are available:

- Watershed [8] – uses the image gradient as a topographical representation of the image. It is however sensitive to noise and weak gradients, which often lead to a significant over-segmentation.
- Waterfall [9] – allows the reduction of the number of segments generated by the Watershed, however the final result depends on the number of iterations executed, which is rather difficult to estimate for particular image.
- Mean-shift segmentation [10] – is capable of delivering segments which are a good representation of uniform color regions in the image, but the final result is very sensitive to a set of input parameters. There is no easy way to adjust them to suit each particular image.
- Normalized Cuts - in [11] it is used to create a “superpixel” or segments which aim at being local, coherent and representing the image structure at higher than pixel

level. There are two ways to control the number of regions produced: specifying the number of regions in advance [12] or choosing a threshold on the normalized cut value [12]. Neither of these methods is optimal for finding the texture elements.

It is not possible to list all image segmentation methods here, but so far we did not find any method which provide a good representation of uniform color regions for an arbitrary image using stable set of parameters.

We propose to use a novel image segmentation method called "Dominant Color Image Segmentator" (DCIS), which is less sensitive to noise and adaptively discards weak gradients in the image using locally dominant colors for segment boundary detection.

DCIS can be divided into four steps:

1. Selecting uniform color region central points (also called markers).
2. Obtaining adjacency between region central points
3. Detecting locally dominant colors for each point and its neighborhood
4. Generating region labels

The final segmentation depends in a large degree on the first step, the selection of region central points. For this purpose a radial symmetry measure is calculated over the square area surrounding each pixel and region central points are aligned with local symmetry maxima (see Figure 2). The radial symmetry measure is defined as:

$$S(x,y) = -\sum_{i=-R}^{R}\sum_{j=0}^{R}\left|\boldsymbol{I}(x+i,y+j) - \boldsymbol{I}(x-i,y-j)\right| \tag{1}$$

where $\boldsymbol{I}(x+i,y+j)$ is an image pixel at coordinates $(x+i,y+j)$ and R defines the image window size used for symmetry measure calculation to be a $(2R+1)\times(2R+1)$ rectangle.

The symmetry measure has several useful properties which help to achieve convex shaped and low color gradient regions:

- The symmetry measure s reaches a maximum (equal to 0) if all corresponding pixel pairs (x_c+i,y_c+j) and (x_c-i,y_c-j) are identical.
- Symmetry is maximized at the center of radially symmetric shapes (like a filled circle, star, etc.) or along their symmetry axis (for elongated shapes).
- An edge irregularity produces more symmetry maxima and more segments, which in turn prevents the generation of segments with complex boundaries.

The proposed symmetry measure tends to generate an excess of local maxima points, but it does not miss any symmetrical regions in the image (assuming sufficiently large R). Other symmetry transforms exist [13], intended primarily as interest point detectors. They therefore do not detect as many interest points as is usually needed for representing texture elements.

The second step of DCIS is obtaining adjacency between markers, which is needed for detecting locally dominant colors. Because of dense marker distribution, adjacency can be captured by using a Voronoi tessellation of markers and analysis of the boundaries between tessellation regions. As a result, each marker is assigned a list of adjacent markers with which it shares a boundary between corresponding tessellation regions.

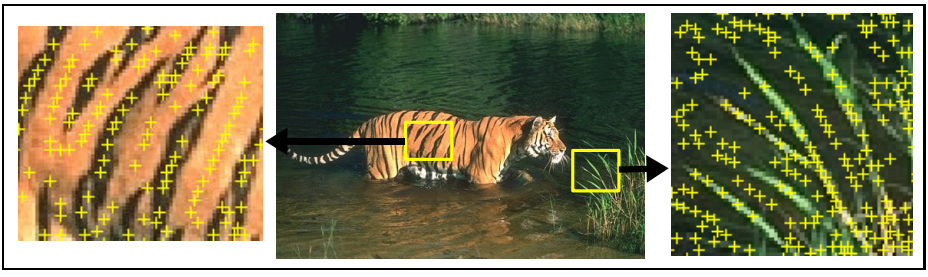


Fig. 2. Example of symmetry maxima location

Extraction of locally dominant colors is performed for each marker separately by agglomerative hierarchical clustering [14] of the colors within each tessellation region and its immediate neighborhood. The clustering uses an average linkage and the Euclidean distances to calculate distance between clusters. The inputs to the clustering algorithm are the colors of all pixels belonging to the tessellation region containing the marker and all adjacent regions. We assume that in the majority of images of natural scenes, the number of dominant colors around each marker is very low, between two and four colors. Therefore for each marker only three or four clusters are used to represent locally dominant colors. The clusters are extracted from the clustering tree level below the one containing two clusters, which can contain three or four clusters depending on the input data - the number of clusters is dictated by the automatically built clustering tree structure.

The process of region boundary selection for a single marker can be best illustrated on the example in Figure 3. The marker m_i has seven adjacent markers $m_a(i, 1..7)$, obtained through marker tessellation. Markers m_i, $m_{a(i,1)}$ and $m_{a(i,2)}$ are located within the uniform color 1 area, while the rest of the markers are located within the uniform color 2 area. Let us also assume that there is a transition area between color 1 and color 2, as is frequently the case in images of natural scenes. After color clustering of pixels belonging to all tessellation regions we obtain three clusters, corresponding to color 1, color 2 and the color transition areas. Cluster 1 is assigned to marker m_i. Since markers $m_{a(i,1)}$ and $m_{a(i,2)}$ also belong to the cluster 1 the tessellation boundaries between region m_i and these markers are preserved. Thus any weak gradients between these regions are ignored. However, the other markers belong to a different cluster, which means that they represent areas with significantly different color. The final boundary of the m_i region is therefore a combination of the cluster 1 boundary with tessellation boundaries inside cluster 1. This boundary selection is performed for each marker in turn.

The boundary selection strategy may leave some pixels unclaimed by both segments, which is usually the case for pixels belonging to the color transition area or small areas with significantly different color not represented by the marker. It is an advantage as it allows for more precise average segment color estimation, since outliers and boundary values are automatically rejected. It does not however allow one to obtain the adjacency of the segment. Fortunately it is relatively easy to assign unclaimed pixels to the segments adjacent to unclaimed pixel regions. In such cases each pixel in the unclaimed

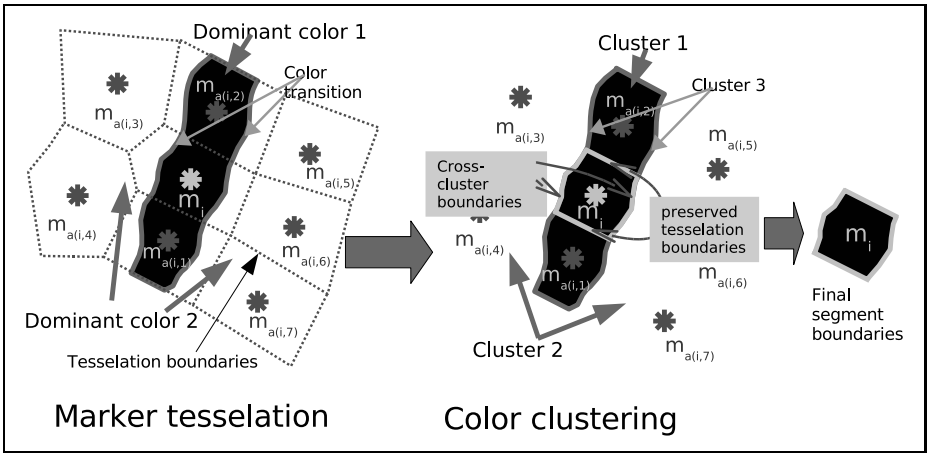


Fig. 3. Example of DCIS segment boundaries selection

region is assigned to the most similar neighboring segment producing a refined segmentation. The refined segmentation is not used for segment feature extraction.

Figure 4 shows differences between the Watershed and DCIS segmentations. One can see that the Watershed produces many more segments and that their boundaries react to weak gradients/noise. DCIS segments are much better suited to the extraction of color pairs related to texture elements. Even though DCIS also over-segments images, e.g. stripes contain multiple similar color segments, each segment of a dark stripe neighbors with a segment of a bright stripe, which is sufficient for extracting texture element related color pairs. It is also worth noticing that the DCIS refined segmentation is a de-noised version of the original image, while the Watershed preserves the noise in the form of small segments.

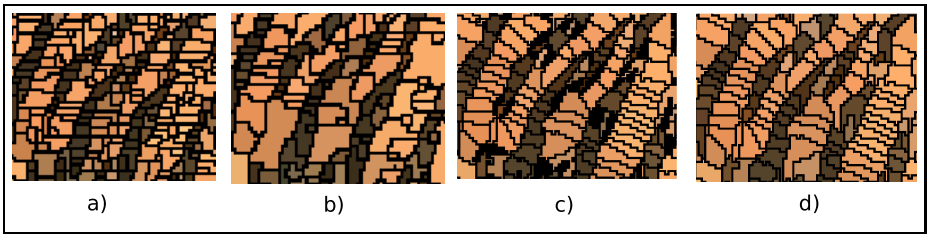


Fig. 4. Examples of various segmentation methods (crop of the tiger body from Figure 2): a) Watershed, b) symmetry marker based Watershed, c) DCIS, d) refined DCIS

Let us summarize the advantages of the DCIS algorithm:

- insensitive to noise (can be used for image de-noising) and locally weak gradients
- segments and extracts local dominant colors in a single pass

– Requires a single parameter R, but in practice it is adjusted to the image resolution. In all our tests we use $R = 5$ for an image resolution of 480×320 pixels.

Hierarchical clustering is computationally expensive. The DCIS execution time for 480x320 pixel images reached approximately 20 seconds on 2.4GHz AMD processor which is significantly longer than the Watershed.

3 Feature Extraction

A feature vector is assigned to each segment resulting from the DCIS segmentation. The feature vector contains the color of the segment to which it is assigned and the color or colors of segments surrounding it. Such features allow one to analyze alternating color patterns. There are several possibilities as to how colors are calculated, but in our case we use dominant colors generated by the DCIS segmentation. DCIS extracts 3 or 4 clusters for selecting segment boundaries, but 2 clusters can easily be extracted from the next level in the clustering tree. Those two clusters represent two colors, one related to the current segment and adjacent segments containing similar colors, and the second color representing all adjacent segments containing color which is not similar to the current segment. Although it is possible to use more than two colors, all our experiments were carried out with two color feature vectors.

4 Feature Clustering

Color clustering has been applied in the past for color reduction and segmentation [15] as well as other problems, but it was always performed in three dimensional space (single color clustering). Since our goal is the detection of color patterns we cluster color pairs, which means using a six dimensional space.

Before the clustering starts, color pairs are sorted by their luminance. The color with the lower luminance value always occupies the first three elements. This prevents two clusters per pattern forming due to different color orders.

FCTD uses agglomerative hierarchical clustering [14] based on the average linkage to build a cluster hierarchy of color pairs (and corresponding DCIS segments). Hierarchical clustering performs a cluster reduction by pairing nearest clusters when progressing from level N to level $N - 1$. This process is repeated until only two clusters remain at level 2.

Figure 5 is an example of a color-pair cluster hierarchy. At clustering level 2 the whole image is divided into two large regions. More significant clusters appear at level 4. Also a number of segments on the boundary between the tiger and water form a separate regions. At the tiger boundary bright tiger segments and dark water segments create the most different color pairs. It is possible to re-attach separated boundary clusters to the tiger body based on statistical analysis of the segment features and their position. We know that a relatively small number of separated boundary segments have a very similar color to the segments belonging to the tiger body. Also most of the boundary segments are adjacent to the segments belonging to the tiger body, representing both

dark and bright stripes. This allows us to assume that boundary segments belong to the tiger body as well.

This example shows that different textures may be better detected at different clustering levels. This is caused by the fact that feature spread varies between textures and that feature spread inside clusters decreases at higher clustering levels. For example, at clustering level 3, the water is represented by only a single region, but at clustering level 4 it is divided into two regions and at level 5 into 5 regions. At the same time the tiger body is divided into smaller regions at higher clustering levels, which makes them less usable in this case.

The main advantage of hierarchical clustering is that it enables texture detection at different clustering levels. We can also observe that some image regions remain stable in a number of different clustering levels, i.e. belong to a single cluster across multiple clustering levels, like most of the tiger body or a number of water regions, which may be a useful tool for texture detection and classification.

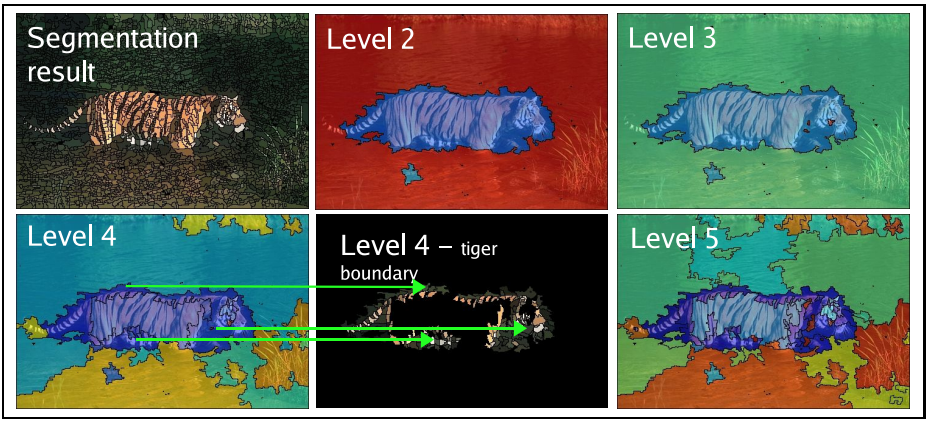


Fig. 5. Example of a cluster hierarchy ($R = 5$)

5 Texture Detection

The texture detection step attempts to discover color patterns in each cluster separately. It means that only features belonging to a single cluster are clustered again, but this time color pairs in feature vectors are not sorted. This approach divides a texture into two clusters whose segments neighbor each other. To distinguish between the two types of clustering done, we refer to the clustering level obtained by the algorithm of Section 4 as *image clustering level* and the clustering level obtained by the clustering described in this section as the *texture clustering level*.

Following this path we can say that if two or more clusters have a high percentage of segments neighboring each other, then we should treat them as a texture. The texture is detected if the length of the boundary B_{kl} in the image between groups of pixels

belonging to two clusters k and l, relative to their total boundary length B_k and B_l exceeds a cluster co-occurrence ratio threshold τ (in range 0 to 1):

$$\frac{B_{kl}}{B_k} \geq \tau \quad \wedge \quad \frac{B_{lk}}{B_l} \geq \tau \tag{2}$$

Typical values for τ vary in the range from 0.7 to 0.9 due to the fact that clusters are usually of different size and that we also allow for a small number of segments in both clusters to not be neighbors of the segments in the other cluster.

Re-clustering is performed until the adjacency condition in Equation 2 is fulfilled or the maximum clustering level is reached, which means that no textures were detected. If the adjacency condition is fulfilled then a texture consisting of two adjacent clusters is detected. The texture detection is continued further in an attempt to divide the detected texture (two adjacent clusters) into 2 textures (4 clusters altogether). It means that up to 3 textures can be detected from a single image cluster. This way we provide a choice of less and more generalized results for further processing.

The texture detection step produces multiple textures from all clustering levels. Figure 6 shows examples of the most representative textures detected at various clustering levels on images from the Berkeley segmentation dataset and benchmark [12]. Further results are presented in [16].

The final result of the texture detection depends on two parameters: R and τ. All results shown were obtained using $R = 5$ and $\tau = 0.7$. These parameters primarily affect the boundary of the detected textures. Figure 6 shows only the most representative textures detected.

The use of color-pair based features sometimes leads to inaccuracies in the detected texture boundaries, when segments at the texture boundary have other neighbors with more different colors, not belonging to the texture. This problem however can be solved as discussed in Section 4.

FCTD in many cases provides similar texture boundaries to the segments produced by methods described in [3,7] on the same image.

6 Conclusion

We have presented two novel algorithms performing color image segmentation (DCIS) and texture region detection (FCTD).

DCIS is a general method for finding uniform dominant color regions. It has a single stable parameter, is noise insensitive and adapts to a local contrast.

FCTD is a detector of textures made up of alternating colors in still images. It is based on a novel idea of color pair hierarchical clustering, where color pairs are extracted from neighboring uniform color segments. The method proved capable of detecting most textures consisting of alternating color patterns, at different regularities, from natural scenes. FCTD is based on an idea of detecting textures at different regularity levels, which is the primary difference with respect to existing methods already cited in this paper. This flexibility allows for detection of less and more regular textures within the same image. For example many existing methods cannot detect both the tiger and the water in Figure 2 at the same time. Though currently FCTD uses only color features,

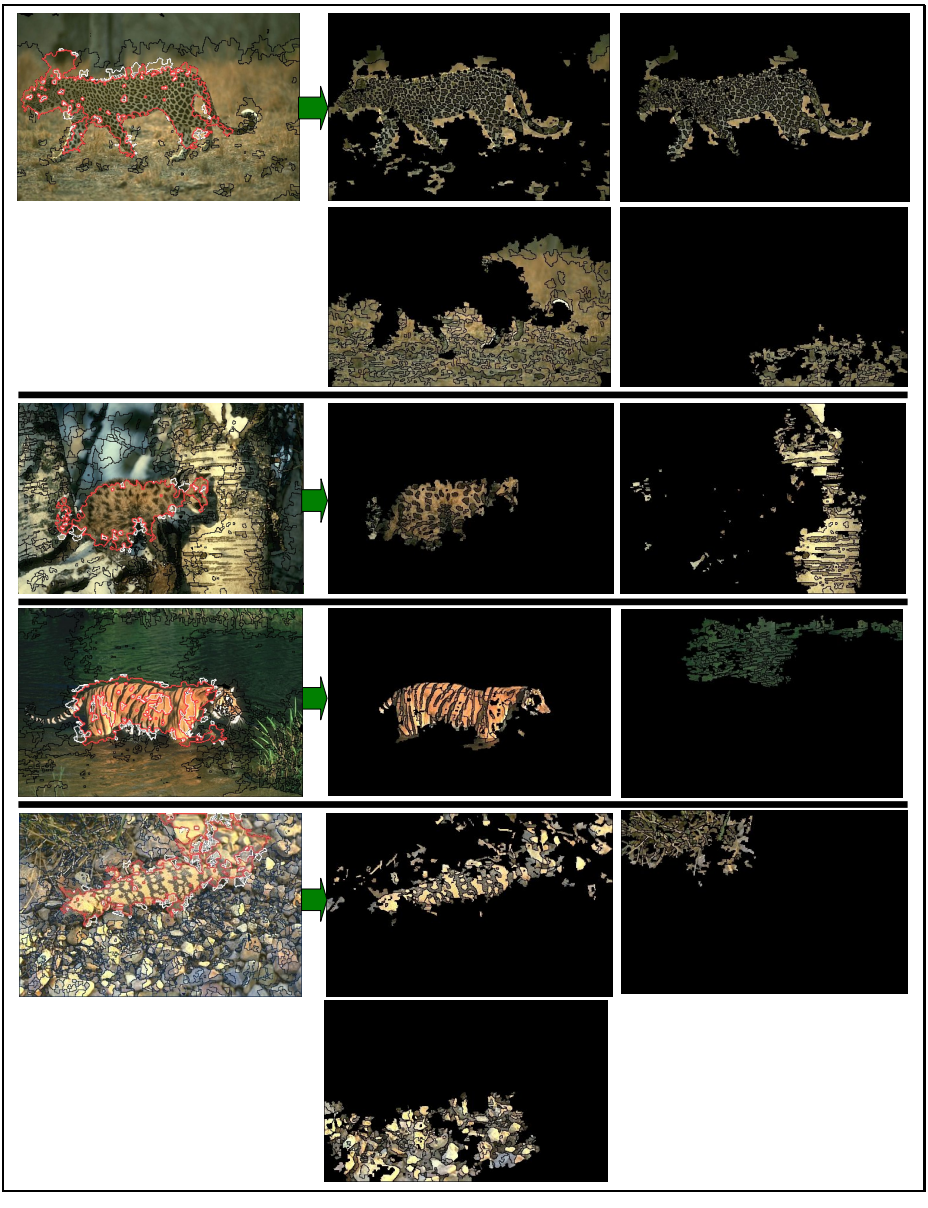


Fig. 6. An example of texture detection from images of size 481x321 pixels (in the first column). The regions corresponding to the detected textures are shown in the second and third column.

which limits its application as a texture detector, the method can be extended to use geometrical features.

Future work includes the application of geometrical features in addition to color, to make FCTD more insensitive to luminance changes and particularly shadows. We are

also going to use detected texture regions for texture classification and use the classification results for selecting regions best representing textures.

References

1. Chetverikov, D.: Pattern regularity as a visual key. Image and Vision Computing **18** (2000) 975–985
2. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. International Journal of Computer Vision **43** (2001) 7–27
3. Mičušík, B., Hanbury, A.: Steerable semi-automatic segmentation of textured images. In: Proc. Scandinavian Conference on Image Analysis (SCIA). (2005) 35–44
4. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22**(8) (2000) 888–905
5. Fowlkes, C., Martin, D., Malik, J.: Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In: Proc. CVPR. (2003) II: 54–61
6. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. IJCV **46**(3) (2002) 223–247
7. Sumengen, B., Manjunath, B.S., Kenney, C.: Image segmentation using curve evolution and region stability. In: Proc. International Conference on Pattern Recognition. Volume 2. (2002)
8. Soille, P.: Morphological Image Analysis. Springer (2002)
9. Marcotegui, B., Beucher, S.: Fast implementation of waterfall based on graphs. In: Mathematical Morphology and its Applications to Image Processing, Proc. ISMM'05. (2005) 177–186
10. Comanicu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. In: IEEE Trans. Pattern Anal. Machine Intell. Volume 24. (2002) 603–619
11. Ren, X., Malik, J.: Learning a classification model for segmentation. In: International Conference on Computer Vision. (2003) 10–17
12. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision. Volume 2. (2001) 416–423
13. Loy, G., Zelinsky, A.: Fast radial symmetry for detecting points of interest. Pattern Analysis and Machine Intelligence, IEEE Transactions on **25**(8) (2003) 959–973
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience (2000)
15. Li, G., An, C., Pang, J., Tan, M., Tu, X.: Color image adaptive clustering segmentation. In: Proc. Image and Graphics Conference. (2004) 104–107
16. Szumilas, L.: Feature co-occurrence texture detector. Technical report, PRIP-TR-099. Pattern Recognition and Image Processing, Vienna University of Technology (2005)

CPU-GPU Multithreaded Programming Model: Application to the Path Tracing with Next Event Estimation Algorithm

Christophe Cassagnabère, François Rousselle, and Christophe Renaud

Laboratoire d'Informatique du Littoral
Université du Littoral Côte d'Opale
BP 719 - 62228 Calais cedex - France

Abstract. Today's hardware includes powerful devices such as graphics process units (GPU) that are not always used to their maximum capacities. Our main goal is to take advantage of these unused resources. To achieve this, we abstract GPUs as SIMD streaming coprocessors and use them within the framework of a multithreaded parallel model. Thus we aim to use all the computing power of a modern PC for speeding up a global illumination simulation software.

The global illumination of a virtual scene can be estimated with stochastic methods such as Path Tracing. These methods however remain costly in terms of rendering time, because of the high sampling required to produce good quality frames. The most part of the rendering time is spent performing intersections tests between rays and triangles. We propose to speed up the rendering of a frame, by using all the available CPUs and GPUs. Our work is based on the *ray engine* developed by Carr et al. for ray tracing, and is mapped to our parallel programming model.

1 Introduction

Since few years, graphics display hardware has known a really fast evolution, from a static locked pipeline model to a more general and flexible model, including programmable vertex and fragment processors. Despite that these new functionalities has been introduced at first for real-time shading calculations, it has been observed that they can be used for general purpose calculations, extending to far beyond rendering and displaying.

It has been shown by Peercy et al. [1] that GPU and OpenGL architectures can be seen and used as SIMD streaming processors. Chan et al. [2] presented an algorithm that automatically subdivides large shaders into a set of smaller and simpler ones, in order to circumvent problems induced by shader length and input constraints. This programmability of GPUs leads to many sorts of use [3]. Thus, applications to linear algebra [4], physical simulations [5] have successfully been adapted to a GPU execution.

The graphics community elaborated new solutions for faster rendering processes. Carr's ray engine [6] computes ray/triangles intersections on GPU. Purcell

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 265–275, 2006.
© Springer-Verlag Berlin Heidelberg 2006

et al. [7] showed that the whole raytracing process can be mapped to GPU, using their stream programming model. Both of these works led to several others raytracing implementations on GPU including accelerating structures management, such as Kd-tree [8,9]. A radiosity algorithm for GPU has even been presented [10].

Thomson et al. [11] explored the GPU utilisation by assimilating it to a general-purpose vector processor and by implementing a software layer on top of graphics libraries which permit arithmetic calculations on arrays of floating point variables. Nowadays, libraries such as Brook for GPUs [12] allow a fast and easy development of software adapted to GPU processing.

Nevertheless, it seems that all works tend to implement the whole rendering process on the GPU. No approach takes benefit from the implicit parallelism between a GPU and a CPU on a hardware architecture. However, removing all calculations from the CPU and overloading the GPU is probably not the most efficient solution to be considered. Furthermore the GPU is not well adapted for some kind of tasks such as recursion, loops, etc (even if loops are no longer a problem in ATI's newest generation GPU). A GPU can thus be less efficient than a CPU in some processes, even if it is faster for SIMD massive calculations. Using each processor according to which kind of treatment it is the most efficient for, is a challenging approach that has never been studied so far.

Finally, none of already presented approaches deals with factorizing and allocating calculations on several different GPUs in a unique software execution context.

All these ideas make sense if we take into account that most of the computing/rendering engines under development are multithreaded in order to be adapted to recent multicore systems. In such a case, abstracting an available GPU as any other processor becomes particularly interesting. Moreover, some motherboards can handle several graphics boards simultaneously (and so permit multiple screen displays management). Recently, SLI technology has even been proposed by Nvidia (ATI owns the concurrent CrossFire technology). This technology intends to increase 3D display performances by coupling two identical graphics boards and merging their computing capacities in order to obtain a faster unique display output.

Nonetheless, we will show that the gain of SLI technology in terms of parallelism can be improved. We present in this paper, a general and modular multithread programming model which can handle any number of CPUs and GPUs. In the following part of this article, we present the general architecture of our model, and we detail the factorization of tasks and the inner synchronisations. In the third part, we present an application of our model to a global illumination rendering algorithm. Finally, we present the results we obtained on a specific hardware and we evaluate the gain of our solution, then we compare it to SLI technology.

2 Our Multi GPU Processing Model

We have considered using several graphics boards as so many SIMD processors for any kind of complex computing. Associated to available CPU processors, our architecture can be seen as a heterogeneous parallel multiprocessor computer.

2.1 Multithreading Model

Our implementation is based on a relatively classical "on demand" task distribution model. A main thread manages a tasks queue. A client thread is created for each available processor. Each of these client threads asks for a task to the main thread as soon as they complete the previous task. This process continues as long as some task remains in the main queue. Results of each task are progressively collected in a common final data array. When there is no more task to be computed, the main thread commands the termination of all client threads and displays final results.

Managing several GPUs. Two main problems occur when using several GPUs at the same time, in the same program. First of all, we have to specify precisely the GPU we want to access. In a multi-windowed application, if the display is split on different screens, managing the correct GPU is automatically done when the coordinates of a window are specified. However in our approach, GPUs are used exclusively for calculations and no window is displayed.

Under X window system (on Unix-like OS), using OpenGL, our solution is to specify which GPU will be used thanks to its screen ID parameter while calling the glXChooseFBConfigSGIX function. The second problem is that the management of OpenGL calls resides in global variables, functions and methods. While several graphic units are used simultaneously by different threads, several contradictory OpenGL calls may occur, as it is not possible to manage simultaneously different rendering contexts.

The solution is to split the program into several independent sub-programs using the fork instruction. Each one of these sub-programs can handle his own independent OpenGL context, and thus access any GPU with no constraint.

Shared memory for data exchange. Unix-like systems provide a useful shared memory mechanism. It allows to define blocks of reserved memory that are common not only for threads, but also for sub-programs spawned using the fork instruction. Any sub-program is able to read/write data defined in the shared memory zone. All input/output data needed for computations done on GPUs have to be located in this area.

Scheduling and synchronizing. Our parallel model consists of:

- A main program that manages the scheduling, synchronisations and the general dispatching of tasks through classic multithread parallelism.
- Several sub-programs that drive a GPU each.
- As many threads that are dedicated to interfacing communications between the main program and a sub-program each.

This model implies two levels of synchronization. A first one gets involved between the main program and client threads in order to set up an "on demand" model (see figure 1). Every thread asks the main program for a job each time they are free and sends the results back. The second one takes place between

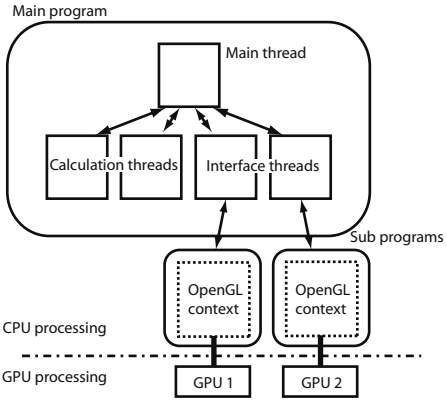


Fig. 1. Organization of processes in our model. Calculations are split between calculations threads running on CPUs, and sub-programs controling GPUs. These sub-programs are synchronized thanks to interface threads.

sub-programs and client threads running as interfaces. It is in charge with controlling data communications between programs via shared memory, triggering calculations on GPU, and retrieving results in shared memory.

We use this two-level model of synchronisations because of the different involved communication protocols. In fact, sub-programs cannot collect signals coming from thread calls. If we intend to synchronise both threads and sub-programs on the same signal, then an interface thread for each sub-program (in charge with inter-programs communications) simplifies greatly the algorithm and the signals management.

Threads and sub-programs driving GPUs do not represent a heavy computation load (less than 1% of the global computations). So if the architecture we use also contains multiple CPUs, we can create more supplementary client threads that will use available computation time on CPUs. On the figure 1 we can identify two types of client threads. Some perform calculations on CPUs, and others are just interfaces with sub-programs. These sub-programs actually control calculations on GPUs, through a local OpenGL context.

2.2 Load Balancing and Modularity

Due to the "on demand" model of our implementation, we do not have to control the load balancing. All we need to guaranty an equitable load balancing is to factorize the global process into tasks that are sufficiently small. However, tasks have to be consistent enough to ensure an efficient SIMD processing on GPUs. We will discuss this issue more in details while presenting our application to a path tracing with next event rendering process.

Thanks to this model, the client threads management is exactly the same for a thread running exclusively on a CPU, or for a thread controlling a GPU via a sub-program. Data transfers, and synchronisations are exactly the same. This two-level model with interface threads simplifies the management of synchronisations.

Moreover it generalises the model to any type of architecture with any number of CPUs and/or GPUs.

3 Application to a Rendering Engine

Using our GPU/CPU programming model, we implemented a rendering engine based on stochastic rendering methods (path tracing, path tracing with next event, bidirectional path tracing).

An often-quoted statistic reported by Whitted [13] is that more than 95% of the time used by ray tracing rendering can be spent performing intersections tests in complex environments. This statistic also applies to algorithms based on path tracing. In this context, using a parallel implementation for performing intersection tests makes sense if we intend to accelerate the rendering process.

In order to optimise intersections tests for a SIMD processing on a GPU, we had to find an effective way of grouping these tests into blocks of treatments that could be performed in one time. We used the factorization model presented by Pharr et al. [14] for his coherent ray tracing approach. It is designed to manage very complex scenes in an out-of-core rendering context. The triangles have to be stored on the hard drive and the computations have to be scheduled and reordered in order to make memory loads more efficient. However, Pharr's approach is not designed to perform parallel calculations, so we had to make important modifications in order to adapt it to our specific model. This also implied modifications of the classical PTNE algorithm.

3.1 Factorizing Intersection Tests

We propose to classify rays, not by their origin or their direction, but according to partitioning volumes (or voxels) they pass through the scene during the rendering process. While Pharr uses a regular voxels grid, we rather use an adapted BSP tree structure. In order to guaranty that each of our calculation jobs is consistent enough, we have to provide a sufficient number of primitives to be intersected with. When using regular grids, some voxels contain very few triangles. This could make our approach ineffective.

Moreover, Pharr's data structure uses a global reordered queue of coherent rays. But if the ray tracing algorithm can easily be transformed in such a way to maximize the exploitation of rays coherency, it is not the case for PTNE. When dealing with global illumination, more complex reflection models are used and the coherency between neighbour rays decreases dramatically.

In our approach, the concept of the BSP structure is extended to classifying the rays that cross the scene. Whenever a ray crosses a leaf-cell during PTNE, it is stored in a local queue. Each leaf-cell manages both a list of triangles that are contained by the volume of the cell (voxel), and a queue of rays that actually cross it (whatever their direction). The main point is to associate a subset R of rays, to a subset T of triangles (that share a neighbour position). Every ray of R may intersect any triangle in T (see figure 2).

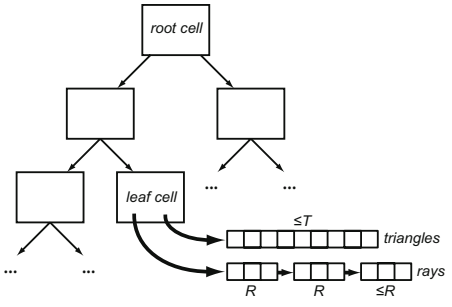


Fig. 2. Each leaf-cell of the BSP-tree contains a maximum of T triangles and a queue of rays packed in tables of size R

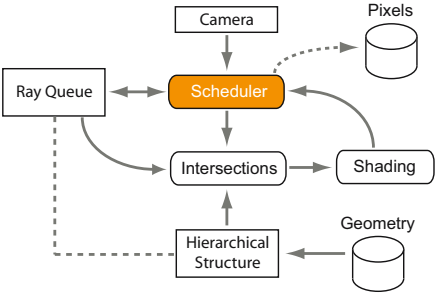


Fig. 3. Our rescheduled PTNE algorithm. Intersection tests are temporized. Rays are classified and queued in the hierarchical structure. The PTNE is no longer recursive.

The main advantage of this classification is that it is now possible to treat simultaneously all intersection tests induced by the association of R and T. We however have to tune precisely BSP parameters. Performances of this approach resides in the cardinal of T. If there are too many triangles in T, the acceleration structure is ineffective. If there are too few triangles, the SIMD parallelism is ineffective.

3.2 Rescheduling Intersection Tests

Finding associations between two given sets R and T implies modifying the classical execution of the path tracing rendering process. In the PTNE algorithm, paths are treated sequentially one after another. We have rescheduled the sequential generation of eye-paths.

Our algorithm. Primary rays are spawned and stored in the queue corresponding to the first leaf-cell they cross. Whenever the scheduler detects a sufficient number of stored rays in a leaf-cell, it interrupts the generation of primary rays and starts treating the concerned cells queue (see figure 3). Treating a cell means proceeding intersection tests between each element of R and each element of T in this voxel. For each ray in the R queue :

- If we don't find any intersection located in the current cell, we reclassify the ray in the following cell it will cross.
- If we find an intersection, the shading is computed and reflected, refracted and shadow rays are spawned. Every of these new rays are stored in the current cell.
- Whenever a ray intersects a light source, its contribution to the final image is computed and stored.

Adding new rays in other cells, may trigger the treatment of another leaf-cell whenever its threshold is reached. So, a waiting queue mechanism for cells to be

treated is implemented. The chain reaction of cell processing stops when every cell contains less rays than its threshold and when all primary rays have been generated through the scene. To terminate the process we launch a last flushing step of the BSP structure since some remaining rays still have to be treated.

Computing intersection tests. Given a set T and a set R, we can perform effective ray-triangle intersection tests computation in a SIMD fashion using the *ray engine*, presented by Carr et al. [6]. One of the main point here is to minimize latencies between GPUs and the CPU. A first latency time is due to data formatting before sending them to the GPU. A second latency time occurs after tests have been performed to read back data from the GPU and compute the shading.

We evaluated that the time used for performing intersection test on GPU represents up to 82% of time used for the complete calculations of an entire queue of rays. So the real latency corresponds to 18% of a BSP cell processing time. It is hardly compressible without thinking of deporting more calculations on GPUs and/or on other CPU calculation threads.

4 Results

4.1 Setting Up the Ray Engine

In spite of a calculations overload due to major modifications in the path tracing process and to the supplementary management of many rays queues, using GPU for intersection tests leads to interesting performances. The tests we performed present a gain of about 1.7 over the results we obtained with the processing of a more classical PTNE algorithm on a standard CPU. Speedup however relies on the BSP tree depth and on the size of the rays buffers.

Figure 4 presents the results we obtained on various configurations of BSP tree structure. The computer we used for these tests is built on an AMD Athlon64 X2 dual core processor, and two Nvidia GeForce 6600GT graphics boards. Our test scene contains 14702 triangles and 32 light sources (see figure 7). The output images are in 512x384 pixels and are processed with 10 samples per pixel. For this first test phase, we only use an unique GPU during all performing. This permits us to evaluate directly the gain coming from SIMD parallelism on a GPU processing.

We tested many thresholds of ray queues. In fact, intersection tests are only triggered if the number of rays in the queues exceeds this threshold. We noticed that the ray engine is more efficient as long as the number of rays that are coded in textures corresponds to a multiple of the GPU cache size ($N \times 128$, see figure 5). We also put in evidence a dramatic increase of memory needs as we increase the threshold. In fact, it is necessary to store more rays before performing intersection tests. Thus the more rays are stored and scheduled, the more memory is required. In our algorithm, we estimate that an ideal efficiency/memory needs ratio corresponds to a value of 512 rays ($N = 4$) for the threshold. From now, this value

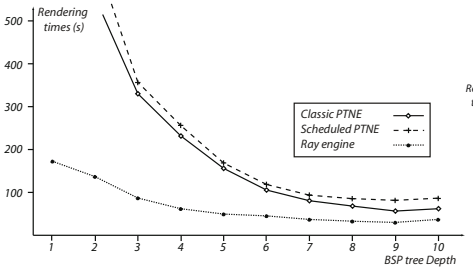


Fig. 4. BSP tree depth tests

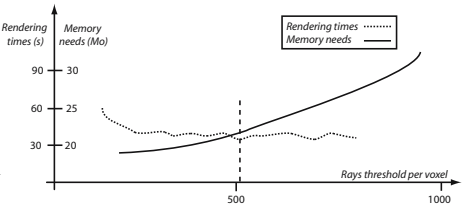


Fig. 5. Rays threshold tests

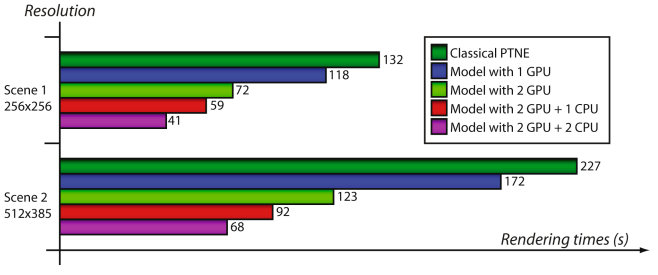


Fig. 6. Rendering times obtained testing different configurations on our hardware

will be used in all of our tests. As a consequence, the dimensions of the textures representing rays in our ray engine implementation are defined to 64×8.

Results are interesting but slower than we could have expected. There must be other places to be optimised. For example, all of the triangles contained in a leaf-cell are sent to the graphics board each time we intend to test intersections with a set of rays. We could use vertex arrays in order to load the whole scene on the graphic board before any calculation and to then avoid sending the same triangles sets several times.

4.2 Multithreaded Results

The figure 6 presents the results we obtained testing several configurations on our system. We used the first test scene we already talked about, and another smaller one (see figure 7). The second scene contains only few thousands triangles and a unique area light source. The BSP-tree is also smaller and implies fewer and smaller leaf-cells, as the tree depth is correlated to the number of triangles in leaf-cells. The gain is limited because of the limited numbers of triangles in cells, and also because the load balancing is less efficient with fewer cells.

Presented results are rendering times obtained for classical PTNE algorithm, with a unique GPU, then with two GPUs, 2 GPUs plus 1 CPU, etc. We can observe a performance gain each time a supplementary processor is used for intersection testing. However the gain obtained adding a CPU to the process is lower than when adding a GPU. According to the precedent testing about ray

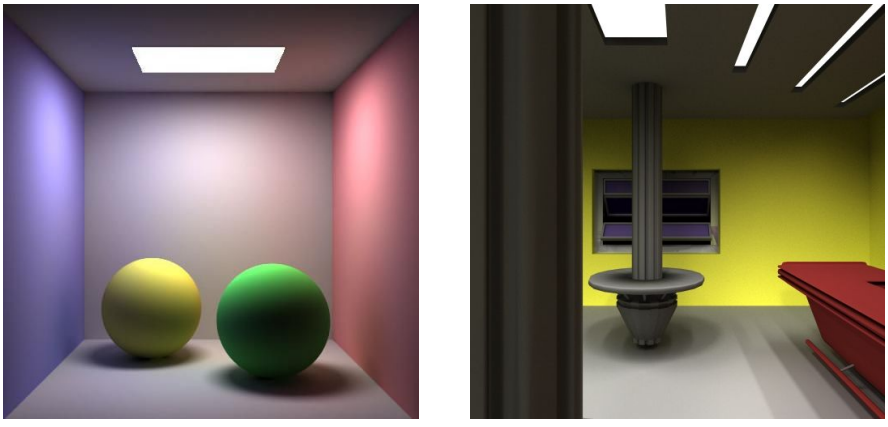


Fig. 7. Two scenes used for testing our approach. The Cornell box contains few thousands triangles. The bar scene contains about 15,000 triangles and 32 light sources.

engine, it seems natural. Each GPU runs to full capacity the most of the time. Some free computing time remains on the CPU and can be used for running a supplementary client thread. Thus the bottleneck of our system appears to be the GPU. In particular feeding the GPUs with all required data more efficiently appears to be the great challenge.

4.3 Comparison with Nvidia SLI Technology

Three SLI [15] rendering modes are possible: Alternate Frame Rendering (AFR), Split Frame Rendering (SFR) and SLI anti-aliasing (SLIAA). These three modes dispatch rendering tasks between two GPUs in a different manner. The choice of the optimal mode depends on the 3D software.

In our case, we base our algorithm on rendering a unique frame in the frame buffer, without any real-time display, or any anti-aliasing. The most adapted mode is thus SFR. In this mode, each GPU is in charge with one part of the same frame: the GPU0 renders the upper part of the frame, and GPU1 renders the lower part. The splitting of each frame tends to balance the computation load at best between the two GPUs. Nevertheless, the SFR mode implies more communications between GPUs, and so limits the time gain.

We set up our rendering application for 1 GPU, and launched the software with SLI mode enabled. Thus we assume that the assignation of tasks on the 2 GPUs is automaticaly managed by the system. We compared the obtained result to the one obtained with our application set up for 2 GPUs with SLI mode disabled. We obtained a 35% gain in the better cases using SLI when our method is close to 40%.

Moreover some problems appear when using SLI. First of all, it automatically requires and uses two GPUs. It is not the case of our solution. We can select only some GPUs for calculations and leave some free, if necessary. Extending this principle, we easily can launch several independent applications, one on

each available GPU. This is not possible with SLI. Furthermore, we have to note that in SFR mode, both graphics boards need a copy of the needed data for rendering the current frame. Data have to be duplicated in both cache memories and the available memory on two SLI boards is not larger than on a single board. This is not the case with our approach. As each board performs independent calculations, it can load and store required data without any constraint. Any card may load different data and available cache memory size is doubled.

To conclude, for now, SLI is limited to coupling two identical boards (but ATI CrossFire technology successes in handling two different boards). Whatever our approach is easily adaptable and scalable to any type and any number of GPUs.

5 Conclusion and Future Work

We presented a programming model for CPU-GPU multithreading and an implementation of Path Tracing With Next Event Estimation illumination algorithm, based on it. This sped up the rendering process for an equivalent global illumination quality. However, our goal is not to compare these results with other dedicated hardware approaches or with some cluster implementation but rather to optimise the possibilities of recent hardware by using as much available computing ressources as possible.

Today's hardware tends to include more and more powerful devices such as GPUs that are not always used to their maximum capacities. The model we presented proposes a way to develop a multithreaded software which could use GPUs as any other CPU in a transparent manner. For now, we are thinking about developing a library which would encapsulate all needed functions and variables in order to make our approach even more useful.

Concerning our path tracing implementation, we are currently working on making it faster and more efficient. For intersection tests on GPU, instead of sending to the graphics board all of the triangles contained in a leaf-cell, we considered to define vertex arrays before any calculation. With all data already loaded on boards, the intersection tests would be faster as we suppress a major part of communications between GPUs and CPUs.

Finally, we rebuilt the path tracing algorithm around a BSP-management base. It also would be interesting to try different approaches (such as ray classification, which is more selective in grouping rays) and test their behaviour in our multithreaded model.

References

1. Peercy, M.S., Olano, M., Airey, J., Ungar, P.J.: Interactive multi-pass programmable shading. In: SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. (2000)
2. Chan, E., Ng, R., Sen, P., Proudfoot, K., Hanrahan, P.: Efficient partitioning of fragment shaders for multipass rendering on programmable graphics hardware. In: HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. (2002)

3. David Luebke, Mark Harris, J.K.T.P.N.G.I.B.C.W., Lefohn, A.: Gpgpu: general purpose computation on graphics hardware. In: GRAPH '04: Proceedings of the conference on SIGGRAPH 2004 course notes. (2004) 33
4. Krüger, J., Westermann, R.: Linear algebra operators for gpu implementation of numerical algorithms. ACM Trans. Graph. **22** (2003) 908–916
5. Harris, M.J., Baxter, W.V., Scheuermann, T., Lastra, A.: Simulation of cloud dynamics on graphics hardware. In: HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. (2003) 92–101
6. Carr, N.A., Hall, J.D., Hart, J.C.: The ray engine. In: HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2002) 37–46
7. Purcell, T.J., Buck, I., Mark, W.R., Hanrahan, P.: Ray tracing on programmable graphics hardware. ACM Transactions on Graphics **21** (2002) 703–712 ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).
8. Moreno-Fortuny G., M.M.: Unified stream processing raytracer. Poster at GP^2: The ACM Workshop on General Purpose Computing on Graphics Processors, and SIGGRAPH 2004 poster (2002)
9. Foley, T., Sugerman, J.: Kd-tree acceleration structures for a gpu raytracer. In: HWWS '05: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, New York, NY, USA, ACM Press (2005) 15–22
10. Coombe, G., Harris, M.J., Lastra, A.: Radiosity on graphics hardware. In: GI '04: Proceedings of the 2004 conference on Graphics interface. (2004)
11. Thompson, C.J., Hahn, S., Oskin, M.: Using modern graphics architectures for general-purpose computing: a framework and analysis. In: MICRO 35: Proceedings of the 35th annual international symposium on Microarchitecture. (2002) 306–317
12. Buck, I., Foley, T., Horn, D., Sugerman, J., Fatahalian, K., Houston, M., Hanrahan, P.: Brook for gpus: stream computing on graphics hardware. ACM Trans. Graph. **23** (2004) 777–786
13. Whitted, T.: An improved illumination model for shaded display. Commun. ACM **23** (1980) 343–349
14. Pharr, M., Kolb, C., Gershbein, R., Hanrahan, P.: Rendering complex scenes with memory-coherent ray tracing. In: Computer Graphics (Proceedings of ACM SIGGRAPH 97). (1997) 101–108
15. Nvidia-SLI http://uk.slizone.com/page/slizone_learn.html: (SLI Technology Overview)

Real-Time and Robust Monocular SLAM Using Predictive Multi-resolution Descriptors*

Denis Chekhlov, Mark Pupilli, Walterio Mayol-Cuevas, and Andrew Calway

Department of Computer Science, University of Bristol, UK
{chekhlov, pupilli, wmayol, andrew}@cs.bris.ac.uk

Abstract. We describe a robust system for vision-based SLAM using a single camera which runs in real-time, typically around 30 fps. The key contribution is a novel utilisation of multi-resolution descriptors in a coherent top-down framework. The resulting system provides superior performance over previous methods in terms of robustness to erratic motion, camera shake, and the ability to recover from measurement loss. SLAM itself is implemented within an unscented Kalman filter framework based on a constant position motion model, which is also shown to provide further resilience to non-smooth camera motion. Results are presented illustrating successful SLAM operation for challenging hand-held camera movement within desktop environments.

1 Introduction

Recent years has seen the emergence of vision-based algorithms for estimating the pose of a moving monocular camera in real-time [1,2,3,4]. Many of the advances can be attributed to the successful transfer of simultaneous localisation and mapping (SLAM) algorithms from robotics [5,6], allowing localisation to proceed away from known 3-D maps [1,4]. This avoids the restriction of pre-calibrated environments in real applications such as Augmented Reality (AR).

Nevertheless, there remains some way to go before such applications can be realised in practice. The algorithms described in [1,4] have demonstrated SLAM operation when camera motion is smooth and consistent. However, real applications will require that operation is maintained even when camera motion is discontinuous and erratic; if applications are to be usable, then it is critical that SLAM operation recovers quickly and in a stable manner following such movement. Unfortunately, the methods described in [1,4] fail catastrophically under such conditions, and although the use of more general stochastic filtering methods, such as particle filtering, can give increased resilience, the approach is difficult to transfer to full SLAM operation [7]. The problem therefore remains a stumbling block to full utilisation of monocular SLAM.

We assert that previous methods have two key problems. First, feature detection is generally based on correlating with reference templates. This has the drawback that detection is not invariant to viewing direction and although local predictive warping can alleviate the problem [8,4], such methods are always

* This work was funded by ORSAS UK and the UK EPSRC Equator IRC.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 276–285, 2006.
© Springer-Verlag Berlin Heidelberg 2006

likely to be of limited utility. More importantly, correlating templates is known to be highly susceptible to detection ambiguity, especially in cluttered environments, and this makes reliable detection difficult to achieve, particularly when search regions grow due to large uncertainty in camera localisation. Secondly, to minimise the potential for ambiguity, the methods in [1,4] adopt highly predictive motion models producing small search regions. Although extremely effective during periods of smooth motion, these lead to failure when motions violate the assumed model, as they are very likely to do in real-world applications.

The algorithm described in this paper addresses both of these issues. We base feature detection on the descriptors used by Lowe in the Scale Invariant Feature Transform (SIFT) [9]. These have significant invariance to view direction and are also highly discriminative [10]. However, their primary use has been for off-line object recognition and computational demands have previously prohibited their use for real-time applications. For example, in [11], Gordon and Lowe report frame rates of 4 fps for a camera tracking algorithm. To overcome this, we adopt two important differences to that used in the SIFT. First, potential features are located using a fast saliency operator, assisted by search regions derived from the covariance within the SLAM filter. This avoids the computationally expensive task of constructing scale space representations for each frame. Second, scale invariance is achieved by constructing descriptors over multiple resolutions for each feature *only when it is first detected*. Crucially, the mean and covariance estimates for camera position and orientation are used to guide feature and resolution matching during SLAM operation, leading to fast and reliable detection. The only comparable approach is that of [12], but here GPS, rather than camera pose, guides the multi-resolution search. The discriminative power of the descriptors allow us to use a constant position motion model, which 'slackens' the filter and makes it more responsive to discontinuous and erratic camera motion. We use the unscented Kalman filter (UKF) [13] to achieve SLAM, which gives similar performance to that of the extended KF, but has several benefits in terms of ease of implementation. The following sections outline the key elements of the method and results are presented in Section 5.

2 Algorithm Overview

An overview of our SLAM algorithm is illustrated in Fig. 1. At its core is the UKF, providing frame by frame estimates of the mean camera position, orientation, and the 3-D locations of detected features. The uncertainty of the current camera pose and 3-D feature estimates is maintained in a single full-covariance matrix, which is critical for stable SLAM operation, especially loop closing [6]. Details of this filter are given in Section 4. The estimates of mean camera pose and scene structure are then passed to an AR engine, enabling 3-D graphical annotations to be superimposed onto the video stream. This allows visual evaluation of SLAM performance.

The filter measurements are estimated projections of mapped features in the scene. On initial detection of a suitable feature, a local descriptor representing

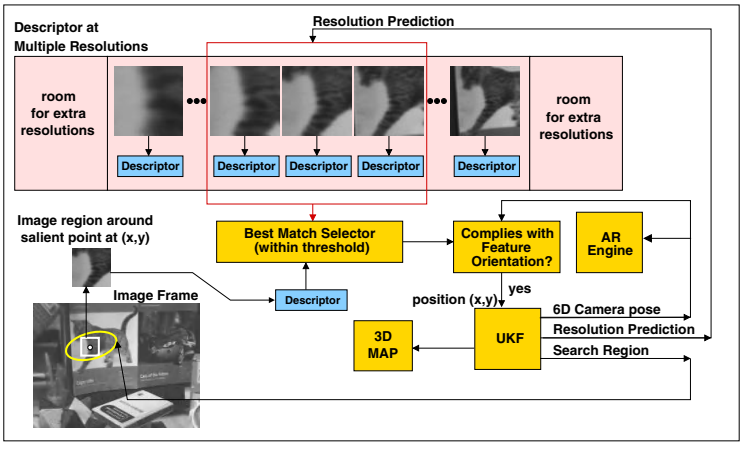


Fig. 1. Schematic overview of SLAM algorithm (see text)

the surrounding image region is built at multiple resolutions, as illustrated at the top of Fig. 1. We use the descriptor adopted by Lowe in [9] and compute it at multiple resolutions. As SLAM proceeds, we seek to detect the same features in subsequent frames by matching descriptors only at appropriate resolutions. In a new frame, descriptors for potential features are constructed at a fixed resolution and then a match is sought with those computed within initialisation frames. Importantly, this matching process is aided by the SLAM filter in three ways. Firstly, we use the predicted measurement covariance for feature locations to constrain the image search for each feature. Secondly, the estimated camera location and the uncertainty in the map are used to predict the likely range of resolutions at which to match the feature and hence further constrain the search. For example, if the camera has moved further from the scene since the feature was initialised, then we would expect to match the corresponding feature at a lower resolution, i.e. constructed within a larger region at initialisation. Finally, the mean and covariance of the estimated camera orientation is used to ensure that the change in dominant orientations of the matched descriptors are consistent with the change in camera pose. This combination of highly discriminative features and constrained matching provides reliable and fast measurements for the filter, even when uncertainty in camera pose becomes large, such as occurs following discontinuous or erratic motion. We are not aware of any other SLAM system that performs multi-resolution matching in such a top-down framework.

3 Feature Detection, Description, and Tracking

We separate detection from description following the work in [2]. Initially, candidate feature matches in a frame are detected using the Shi-Tomasi operator [14] which gave sufficient repeatability: features which are to be tracked must be detected as salient points in many, if not all, frames.

3.1 Feature Description

Having detected suitable feature locations, we then build descriptors to enable recognition of the same feature in subsequent frames. This is based on computing histograms of spatial gradients within the region surrounding the feature. Such descriptors have been shown to have good invariance properties to luminance changes, rotations and local warp [9,10]. To account for changes in viewing distance, however, we also need to incorporate scale invariance. In the SIFT, this is achieved by detecting extrema and building descriptors within a discretely sampled scale space representation [9]. Such bottom-up approaches are computationally impractical for real-time SLAM. Instead, we initially build descriptors at multiple resolutions around salient points and then seek to match those descriptors to those detected at a fixed resolution in subsequent frames. The key difference is that we rely on the saliency operator to achieve repeatability and then use the descriptors to characterise regions in the image, rather than in scale space. This formulation makes efficient use of computational resources in a top-down fashion.

The multi-resolution descriptors are constructed as follows. We parametrise resolution by r, denoting the sampling interval in pixels, along each dimension of a region about the feature point. The number of samples, N_D, is fixed and then we vary r between r_l and r_u with step r_s to generate descriptors over multiple resolutions as illustrated in Fig. 2(left). Thus, for $r = 1$, the descriptor is built at the same resolution as the current frame, whilst for larger and smaller r resolution is decreased and increased, respectively. The descriptor contains a dominant orientation and a normalised vector of orientation histograms [9]. Some features have several dominant orientations, in which case a separate descriptor is created for each.

3.2 Feature Tracking

Tracking amounts to identification of salient features in the subsequent frames using their descriptors. In a new frame we obtain a predicted covariance of a particular feature location from the SLAM filter. A gating of this covariance defines search regions in which salient points are detected. All detected points are potential candidates for the sought feature. We extract a patch of size N_D around each candidate and compute descriptor(s) from this patch. A key point is that we only build this descriptor at a single resolution, the resolution of the image. In order to discover the best match for the current feature we compute Euclidean distances between the candidates' descriptor and the feature's descriptors. Let i represent a frame number and without loss of generality assume that we initialise the feature in the first frame, $i = 0$. Let $\bar{c}_i$ be the mean camera pose estimate in ith frame and $\bar{d}_i$ be the mean distance from the camera to the feature in ith frame. We suggest to compute the resolution of ith frame relative to the resolution of the first frame as follows: $\bar{r}_i = \frac{\bar{d}_i}{\bar{d}_0}$. We can then limit the search to just a few descriptors whose corresponding resolutions are close to $\bar{r}_i$. However, this formulation neglects the uncertainty in the mean estimates. The effect of uncertainty in the system is illustrated in Fig.2(right). To attend to this we

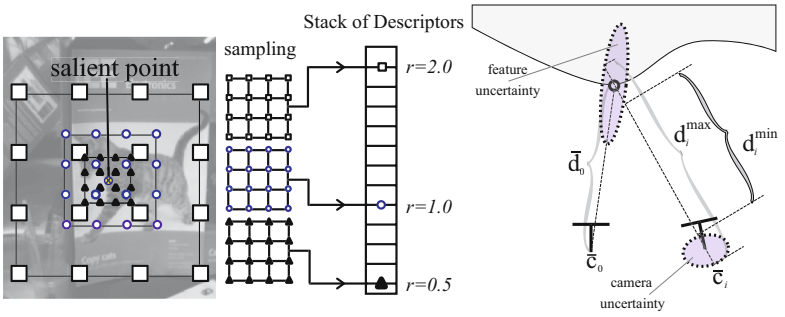


Fig. 2. Left: Same size regions are sampled around a salient point using different sampling intervals r. Descriptors are created for each sampling and saved in a list. **Right:** Uncertainty is used to define the interval of likely distances.

use an unscented transform (see section 4.4) to compute the mean distance estimate $\bar{d}_i$ and a variance for this estimate $\sigma^2_{d_i}$. By taking a gating of the variance we define an interval in which the matching resolution should be found with high probability. The distance interval is denoted as $[d_i^{min}, d_i^{max}]$, where $d_i^{min} = \bar{d}_i - 3\sigma_{d_i}$ and $d_i^{max} = \bar{d}_i + 3\sigma_{d_i}$. The corresponding resolution interval is $[r_i^{min}, r_i^{max}] = [\frac{d_i^{min}}{\bar{d}_0}, \frac{d_i^{max}}{\bar{d}_0}]$. The result of this top-down approach is that when the uncertainty of the camera is small the system decreases the number of descriptors it needs to check, but when the uncertainty grows the system automatically expands the search range of descriptors so that all the descriptors, whose resolutions are within the interval $[r_i^{min}, r_i^{max}]$, are tested. When we find the closest descriptor we check if the Euclidean distance is below some specified threshold τ and if it is then we proceed to the orientation consistency test.

As well as predicting the distance to a particular feature we can predict how orientation should have changed using the difference in position and orientation of the camera in the current frame and at the first frame when the feature was initialised. We use this for a simple orientation consistency check: if the predicted change in orientation and that of the of the descriptor differ by greater than some threshold we reject the match. We found this test important for extra robustness due to the increased discrimination it provides.

Another important component of the algorithm is enrichment of the feature description with new resolutions. When the closest candidate matches against the feature descriptor which lies on the boundary of resolution range, we repeat the procedure described in section 3.1 to build new descriptors. We adjust r_l and r_u accordingly depending on whether we are adding new higher resolutions or new lower resolutions. The new descriptors are added to the corresponding end of the list.

4 SLAM Using the Unscented Kalman Filter

Our formulation of the SLAM filter follows closely that used by Davison [1], although differs in terms of the motion model adopted and the use of the UKF. We

provide an overview here and recommend that readers refer to [13] for detailed information on the UKF.

4.1 State Space Model

The SLAM filter provides estimates of the 3-D position and orientation of the camera (localisation) in tandem with positions estimates for 3D feature points in the scene (mapping). The state vector in the filter therefore consists of two partitions, one for the camera pose and one for the structure. The former is parametrised by $\boldsymbol{x} = [\boldsymbol{t}, \boldsymbol{q}]^\top$, where $\boldsymbol{t}$ is the translation vector and $\boldsymbol{q}$ is a quaternion representing the relative position and orientation with respect to a reference camera frame. In mapping structure we follow the approach of Davison [1], where features are initially concatenated with the state vector as a line representation $\boldsymbol{l} = [\boldsymbol{c}, \boldsymbol{v}]^\top$ and then converted to a Cartesian representation, $\boldsymbol{z} = [z_1, z_2, z_3]^\top$, once they converge after a period of factored sampling (Section 4.4). In the line representation, $\boldsymbol{c}$ is the position of the camera when the feature was first seen, and $\boldsymbol{v}$ is a vector which points from $\boldsymbol{c}$ in the direction of the feature. Assuming all features have converged from their line representation, our entire state space model has the following form: $\boldsymbol{p} = [\boldsymbol{x}, \boldsymbol{y}]^\top$, where $\boldsymbol{y} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M]^\top$ and M is the number of mapped features.

4.2 Observation and Process Models

Observations are produced by transforming mapped 3-D points into the current camera co-ordinate system followed by perspective projection onto the image plane. For the ith point this gives a 2-D point in the image plane $\boldsymbol{u}(\boldsymbol{z}_i, \boldsymbol{x}) = \mathit{\Pi}(R(\boldsymbol{q})(\boldsymbol{z}_i - \boldsymbol{t}))$, where $R(\boldsymbol{q})$ denotes the rotation matrix corresponding to the normalised quaternion $\boldsymbol{q}$ and $\mathit{\Pi}()$ denotes standard pin-hole camera projection for a calibrated camera. The observation model is then the concatenation of the projected points plus noise:

$$h(\boldsymbol{p}) = [\boldsymbol{u}(\boldsymbol{z}_1, \boldsymbol{x}), \ldots, \boldsymbol{u}(\boldsymbol{z}_M, \boldsymbol{x})]^\top + \boldsymbol{r}_k \tag{1}$$

where $\boldsymbol{r}_k$ is a noise vector of dimension $2M$ with a normal distribution. This serves as the model for the locations of detected features obtained from the tracking stage.

The state-space process model defines a motion model which describes how we expect the camera to move from one time step to the next. The feature components, $\boldsymbol{y}$, are assumed to be rigid and hence the process for this partition of the state vector is the identity. To be robust to erratic motion we use a constant position motion model which is more general than constant velocity typically used in SLAM. This implies that our camera pose undergoes a first order random walk. Unfortunately, this cannot be written using an additive noise model without removing the quaternion from the state vector and accumulating it separately as in [15]. In SLAM the quaternion must be in the state vector so that its uncertainty is fully represented, hence, our process model must incorporate a non-linear noise component:

$$f(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{y}) = [\boldsymbol{t} + \boldsymbol{\tau}, \mathit{\Delta}\boldsymbol{q}(\boldsymbol{\omega}) \otimes \boldsymbol{q}, \boldsymbol{y}]^\top \tag{2}$$

where $\Delta\boldsymbol{q}(\boldsymbol{\omega})$ is the incremental quaternion corresponding to the Euler angles $\boldsymbol{\omega}$ as defined in [15] and $\otimes$ denotes quaternion multiplication (composition of rotations). Here, we have used a 6-D noise vector, $\boldsymbol{n} = [\boldsymbol{\tau}, \boldsymbol{\omega}]^\top$, which is assumed to be from a multi-variate normal distribution.

4.3 Unscented Kalman Filter

The UKF is a variant of the Kalman filter which is specifically aimed at problems with non-linear process and observation models, and also non-additive noise models such as that in Equation 2. It is based upon the unscented transformation (UT) which computes the effect of a non-linear function upon a mean and covariance. It operates by computing a deterministic sample set from the covariance which are then propagated through the non-linearity. The weighted mean and covariance of the propagated samples give us the transformed statistics.

For example, the UKF can easily handle the non-linear process noise model in Equation 2. This is achieved by inserting the mean and covariance of the process noise into the stochastic map:

$$\boldsymbol{p}_k = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{n}_k \\ \boldsymbol{y}_k \end{bmatrix} \qquad \Sigma_{pp} = \begin{bmatrix} \Sigma_{\boldsymbol{xx}} & 0 & \Sigma_{\boldsymbol{xy}} \\ 0 & \Sigma_{\boldsymbol{nn}} & 0 \\ \Sigma_{\boldsymbol{yx}} & 0 & \Sigma_{\boldsymbol{yy}} \end{bmatrix} \tag{3}$$

and then performing the computation:

$$\{\boldsymbol{p}^-_{k+1}, \Sigma^-_{\boldsymbol{pp}}\} = UT(\boldsymbol{p}_k, \Sigma_{\boldsymbol{pp}}, f) \tag{4}$$

which propagates the mean, $\boldsymbol{p}_k$, and covariance, $\Sigma_{\boldsymbol{pp}}$, through the non-linear process f. Following this, the predicted measurement and measurement covariance can be computed more simply using the additive noise version of the UT:

$$\{\boldsymbol{h}^-_{k+1}, \Sigma'_{\boldsymbol{hh}}\} = UT(\boldsymbol{p}^-_{k+1}, \Sigma^-_{\boldsymbol{pp}}, h) \tag{5}$$

where the predicted covariance is then $\Sigma^-_{\boldsymbol{hh}} = \Sigma'_{\boldsymbol{hh}} + \Sigma_{\boldsymbol{rr}}$ and $\Sigma_{\boldsymbol{rr}}$ is the measurement noise covariance (pixel noise).

4.4 Mapping New Features Using the Unscented Transform

When inserting new features into the map it is important that the covariance matrix of the expanded map is initialised correctly. In the work of Davison [1], the off-diagonal covariance is computed using local linearisations. By using the UT we can insert new features without linearisation errors.

When we detect an image feature at a location u_x, u_y in the image plane we know the vector $\boldsymbol{u}^c = [u_x, u_y, f]^\top$ which points in the direction of the feature in the camera's local coordinate system. Our uncertainty about $\boldsymbol{u}^c$ has covariance $\Sigma_{\boldsymbol{uu}} = \mathrm{diag}(\sigma_x, \sigma_y, \sigma_f)$, where σ_x and σ_y are the errors in the image plane location of the feature (typically $<< 1$ pixel since we are specifying the interest point) and σ_f is the variance of the focal length estimate (calibration error) and is also typically very small (1×10^{-6}cm)). The line representation to be added

to our map $\boldsymbol{l} = [\boldsymbol{c}, \boldsymbol{u}^w]^\top$ is in world coordinates. The current estimate of camera pose $\hat{\boldsymbol{x}}$ implies that $\boldsymbol{c} = \hat{\boldsymbol{t}}$ and $\boldsymbol{u}^w = R(\hat{\boldsymbol{q}})^\top \boldsymbol{u}^c$, but these estimates are uncertain. To insert $\boldsymbol{l}$ correctly we first augment the stochastic map with $\boldsymbol{u}^c$ and $\Sigma_{\boldsymbol{uu}}$:

$$\boldsymbol{p}' = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}^c]^\top \qquad \Sigma_{\boldsymbol{p'p'}} = \mathrm{diag}(\Sigma_{\boldsymbol{pp}}, \Sigma_{\boldsymbol{uu}}) \tag{6}$$

We can then view the concatenation of $\boldsymbol{l}$ to $\boldsymbol{p}$ as the transformation of the augmented map $\boldsymbol{p'}$:

$$\phi(\boldsymbol{p}') = \phi([\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}^c]^\top) = \left[\boldsymbol{x}, \boldsymbol{y}, \left[\hat{\boldsymbol{t}}, R(\hat{\boldsymbol{q}})\boldsymbol{u}^c\right]\right]^\top \tag{7}$$

Given this, it is easy to compute the new map using $UT(\boldsymbol{p}', \Sigma_{\boldsymbol{p'p'}}, \phi)$. The unscented transform accurately computes the correct covariance.

Similarly, we can perform the conversion from line representation to Cartesian representation using the transform function:

$$\psi(\boldsymbol{p}'') = \psi(\left[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l}_i, \bar{d}_i\right]^\top) = \left[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c}_i + \bar{d}_i \boldsymbol{u}_i^c\right]^\top \tag{8}$$

where the map is first augmented with $\bar{d}_i$ and σ_{d_i}, the depth mean and variance estimated from factored sampling. The sampling is performed using a prior distribution that is uniform in inverse depth as in [16].

5 Results

The systems runs in real-time on an ordinary 2.8 GHz Pentium 4 PC. The implementation is in C++ using OpenGL for visualisation. The code is not fully optimised and there is much potential for improvement. We use a calibrated Fire-wire camera with FOV of 42^o which feeds video at 30 fps at a resolution of 320x240. Fig. 3 shows frames from a typical run. We will provide commentary row by row, left to right, but the reader should note that performance of the system can only be properly appreciated from the example videos.[1]

Top row: start of mapping. The first image shows an external view of camera location with its uncertainty indicated by wire-frame ellipsoids (green: predicted covariance, yellow: estimated covariance). Red ellipsoids are uncertainties in the 3D map of features. The second image shows a frame with elliptical search regions overlaid (green: matched, red/yellow: no match found). The blue boxes indicate how many resolutions are being searched for each feature. Because features are newly mapped their uncertainties are large and hence the search is over many scales. In the third image the map has converged and so searches are performed over fewer resolutions. The final image shows a timing graph for the SLAM filter. Spikes coincide with introduction of new features (grey dashed lines) each causing an increase in processing.

Second row: erratic motion. The first image shows an external view during rapid camera shake. No features are matched because of motion blur hence

[1] http://www.cs.bris.ac.uk/Research/Vision/Realtime/

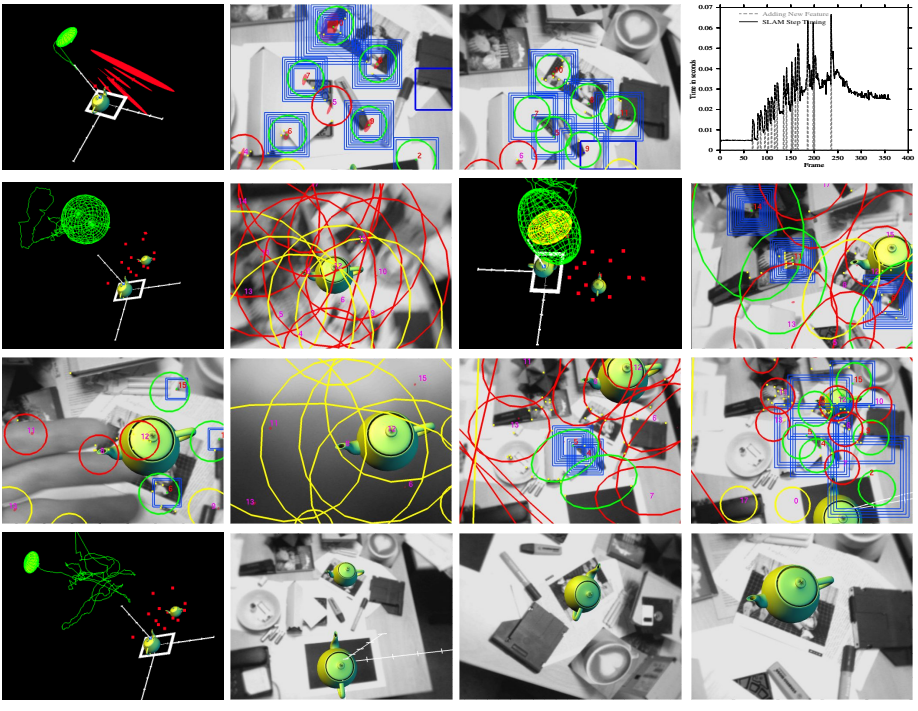


Fig. 3. Frames from a typical real-time run

the camera pose uncertainty has grown large. Image search regions grow in concert with this (second image). Once the shaking stops the UKF corrects the large uncertainty (third image) and the system automatically matches over many scales (final image) in order to recover.

Third row: occlusion. A hand was moved in front of the camera whilst simultaneously moving the camera away from the desk so that there is a scale change unknown to the system. The system recovers much in the same way as it did from the camera shake by successfully matching over many scales.

Bottom row: rotation and scale changes. The system is in normal operation, with converged features and small localisation covariance (first image). The camera moves with large cyclo-rotations and scale changes. Note the correct relative pose of the AR teapots, indicating accurate tracking despite the wide variation in camera angle.

6 Conclusion

By utilising the discriminatory power of SIFT-like features in an efficient top-down manner we have introduced new standards of robustness for real-time, hand-held, monocular SLAM. Such robustness to severe motion blur, total

occlusion, and erratic motion, is critical to real-time applications such as augmented reality where unpredictable human interactions are involved.

References

1. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proc. Int Conf on Computer Vision. (2003)
2. Nistér, D.: Preemptive ransac for live structure and motion estimation. In: Proc. International Conference on Computer Vision. (2003)
3. Chiuso, A., Favaro, P., Jin, H., Soatto, S.: Structure from motion causally integrated over time. IEEE Trans on PAMI **24** (2002) 523–535
4. Eade, E., Drummond, T.: Scalable monocular slam. In: Proc. Int Conf on Computer Vision and Pattern Recognition. (2006)
5. Davison, A.J., Murray, D.W.: Mobile robot localisation using active vision. In: Proc European Conference on Computer Vision. (1998)
6. Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., Csorba, M.: A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotics and Automation **17** (2001) 229–241
7. Pupilli, M., Calway, A.: Real-time visual slam with resilience to erratic motion. In: Proc. Int Conf on Computer Vision and Pattern Recognition. (2006)
8. Molton, N., Ried, I., Davison, A.: Locally planar patch features for real-time structure from motion. In: Proc. British Machine Vision Conference. (2004)
9. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int Journal of Computer Vision **60** (2004) 91–110
10. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans on PAMI **27** (2005) 1615–1630
11. Gordon, I., Lowe, D.: Scene modelling, recognition and tracking with invariant image features. In: Int Symp on Mixed and Augmented Reality. (2004)
12. Hutchings, R., Mayol, W.: Building recognition for mobile devices: incorporating positional information with visual features. Tech. Report, Univ. of Bristol (2005)
13. Wan, E., van der Merwe, R.: The unscented kalman filter. In Haykin, S., ed.: Kalman Filtering and Neural Networks. Wiley (2001)
14. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, Seattle (1994)
15. Azarbayejani, A., Pentland, A.P.: Recursive estimation of motion, structure, and focal length. IEEE Trans on PAMI **17** (1995) 562–575
16. Lemaire, T., Lacroix, S., , Sol, J.: A practical 3d bearing only slam algorithm. In: In IEEE International Conference on Intelligent Robots and Systems. (2005)

A Faster Graph-Based Segmentation Algorithm with Statistical Region Merge

Ahmed Fahad and Tim Morris

School of Informatics, University of Manchester
Manchester, UK, M60 1QD
ahmed.fahad@postgrad.manchester.ac.uk,
tim.morris@manchester.ac.uk

Abstract. The paper presents a modification of a bottom up graph theoretic image segmentation algorithm to improve its performance. This algorithm uses Kruskal's algorithm to build minimum spanning trees for segmentation that reflect global properties of the image: a predicate is defined for measuring the evidence of a boundary between two regions and the algorithm makes greedy decisions to produce the final segmentation. We modify the algorithm by reducing the number of edges required for sorting based on two criteria. We also show that the algorithm produces an over segmented result and suggest a statistical region merge process that will reduce the over segmentation. We have evaluated the algorithm by segmenting various video clips Our experimental results indicate the improved performance and quality of segmentation.

1 Introduction

The problem of image segmentation remains a great challenge for computer vision. A myriad of segmentation techniques have been developed over the years with considerable progress in improving execution time and efficiency. The literature is full of segmentation techniques some of which have showed promising segmentation results. The Normalized Cut algorithm [1] constructs a graph in which the nodes represent the pixels in the image and labelled edges represent affinities between nearby pixels. The image is segmented by minimizing the cost of cutting the graph into sub-graphs where the intra affinity within the sub-graph pixels is greater than the inter affinity of pixels between the sub-graphs. However, this algorithm does not run in linear time with regard to the number of pixels, which makes it an unsuitable algorithm for real time processing of realistic sized images. A fast multi-scale algorithm [2] uses an algebraic multi-grid to approximate the solution to the normalized cut problem using recursive graph coarsening in order to produce an irregular pyramid of region encoding. Cour, et. al. [3] present a multi-scale spectral image segmentation algorithm that solves a cross scale constraint matrix which processes the different spatial scales in parallel by forcing the system to seek an 'average' segmentation across all scales. Macaire, et. al. [4] proposed a pixel classification scheme that constructs a set of classes based on the analysis of the spatial-colour compactness degree which indicates the likelihood of a colour subset, defined by a colour domain, corresponding to an actual region in the

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 286 – 293, 2006.
© Springer-Verlag Berlin Heidelberg 2006

image. This is done by using a selection procedure that will identify a colour subset as a region in the image by maximizing an objective function which in turn will maximize the connectedness and colour homogeneity properties of the colour subset. The JSEG algorithm proposed by [5] applies a criterion of good segmentation that measures ratio of the distances between different classes to the distances within each class. Mean-Shift segmentation [6] estimates the gradient of the probability density of a colour occurring jointly in the spatial and colour domains of the colour image such that each pixel is associated with the closest local mode of the density distribution. An algorithm that integrates the watershed algorithm with the chain code is used, it simulates raining to generate connected components using chain codes and then simulates flooding to label catchment basins by tracing the chain codes [7]. Seeded Region Growing (SRG), which was proposed by Adams and Bischof [8], starts with assigned seeds, it then grows regions by merging a pixel with the nearest neighbouring seed region. [9] Showed how to automatically select initial seeds as pixels having high similarity to their neighbourhoods which do not fall on the boundaries of two regions. Felzenszwalb and Huttenlocher [10, 11] presented a graph segmentation algorithm based on pairwise region comparison. The comparison predicate considers the minimum weight edge between two regions as an external difference while the minimum spanning tree edges of each component is considered as an internal difference. The predicate merges two regions if their internal difference exceeds their external one. Our algorithm exploits the fact that once two regions are merged, the in-between edges of their boundaries are no longer required and hence the number of edges required for sorting is reduced. We also exploit the fact that the majority of regions' internal weights exceed a fraction of the edge weights which allow us to introduce a preprocessing step to merge regions with quite small edge weights. Although this preprocessing step violates the theoretical aspect of the segmentation algorithm, experimental results show that it speeds up the process and with the statistical region merge it produces better segmentations.

The rest of the paper is organized as follows: in the next section, a brief review of the Felzenszwalb and Huttenlocher graph algorithm is presented. This is followed by a description of our faster segmentation algorithm in section 3. In section 4 we use a statistical region merge predicate to minimize the oversegmentation of the graph algorithm. In section 5, experiments on a variety of image data sets are presented with a conclusion.

2 Graph-Based Image Segmentation

Felzenszwalb and Huttenlocher [10] presented a segmentation algorithm that represented the image as an undirected graph $G = (V,E)$ where each node v_i belonging to V corresponds to a pixel in the image and pairs of pixels are connected by edges $e_{ij} \in E$, $i \neq j$. A segmentation, S, is a partition of V into components such that each component (or region) $C \in S$ corresponds to a connected component in a graph $G' = (V, E')$, where $E' \subseteq E$. The internal difference of a component $C \subseteq V$ is the largest weight in the minimum spanning tree of the component, MST(C, E). That is,

$$Int(C) = \max_{e \in MST(C,E)} w(e). \quad (1)$$

The *difference between* two components $C_1, C_2 \subseteq V$ is the minimum weight edge connecting the two components. That is,

$$Dif(C1, C2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j). \quad (2)$$

The segmentation evaluates a predicate for the evidence of a boundary between a pair of components by checking whether the difference between the components, $Dif(C_1,C_2)$, is large relative to the internal difference within at least one of the components, $Int(C_1)$ and $Int(C_2)$. A Euclidean colour distance metric is used to represent the edge weight where a small value indicates strong affinity between pixels. The number of edges in E, using 8-adjacency graph, is four times the number of pixels in an image and the overall time complexity is E log E for sorting. Knowing that most of the computation time is devoted to sorting the edges, we show in the next section how to reduce the number of edges required for sorting. We will refer to this algorithm as F&H.

3 Method

F&H's main processing step is building a minimum spanning tree, MST, using Kruskal's algorithm by sorting the graph edges. Kruskal's algorithm starts by sorting all edges in non-decreasing order then scans the edges in this order adding an edge to the MST if either of its vertices do not belong to the MST; otherwise the edge is not included in the final MST. Similarly, two regions that are merged will mean two MSTs will merge into one MST. There is at least one edge connecting two pixels/vertices from different regions/MSTs and the edge with the minimum weight is tested for merging. Once the merge occurs, the two MSTs merge into one MST by adding the edge with the minimum weight and excluding all remaining edges connecting the different MSTs. We exploit the fact that once we merge two regions, the edges connecting their boundaries are no longer required; therefore they can be removed from the set of edges E'. The algorithm computes the graph edge weights as a distance measure in the RGB colour space instead of the absolute intensity difference because experimentally this gives better results [10]. The weights are real values. The edge weights can be sorted into buckets using the integral part of the weight as a bucket index, therefore each bucket holds greater edge weights than the previous one. In other words, the first bucket holds edge weights that are smaller than the other buckets. The weights in a bucket are not sorted. The graph algorithm is applied as normal to the first bucket's weights by sorting the edge weights by non-decreasing order and then applying the merging predicate. Next, all edges in the next bucket whose ending vertices belong to the same region/MST derived from the previous merging step are removed from the bucket (Fig. 1 presents a simple example illustrating the edge removal step). Sorting and merging follow the edge removal step and the



(a) (b) (c) (d)

Fig. 1. Example of edge removal (a) Four vertices belong to different sets (b) The four vertices are merged into two different sets (c) Thick line indicates edge about to merge (d) dashed edges representing removed edges in the next steps

algorithm repeats over all the buckets. We notice that the method does not affect the quality of segmentation i.e. the segmentation is not too coarse and not too fine.

In addition, we add a preprocessing step that will further speedup the algorithm, however this step may affect the quality of segmentation. Pixels belonging to regions that have negligible intensity variation can be merged into small components, thus reducing the number of edges that are required to be sorted. Hence, we perform a region merging process where any two vertices are merged into one component if their edge weight is less than a threshold. The threshold value is quite small in order not to affect the segmentation too much i.e. not too fine and not too coarse segmentation. We show in the experiments section, that this step hardly changes the quality of segmentation. In summary, the segmentation algorithm is:

0. Set each vertex to represent its own component.
1. If the edge weight e connecting any two components is less than a threshold then merge the two components and remove edge e from E.
2. Else add the edge weight to the bucket with the index value equal to the weight without the fraction.
3. Sort the bucket with the minimum index.
4. Let v_i and v_j denote the vertices connected by the edge e_k. The component to which v_i belongs is merged with the component to which v_j belongs if the edge weight e_k is less than or equal to the internal weight of either component and the two vertices belong to different components.
5. Remove all edges from next bucket if the vertices at the edge's two ends belong to the same component.
6. Sort the bucket in non-decreasing order and process step 4.
7. Iterate until all buckets are processed, or until the next bucket's index is greater than all the component's internal weights.
8. Build a spatial adjacency graph and apply statistical region merge.
9. Return the set of components.

4 Region Merge

The segmentation results obtained from F&H algorithm are oversegmented; we apply inference techniques, that are widely accepted in the vision community [12], to minimize this. Fig. 3(a) shows a comparison of the number of regions obtained from graph

segmentation versus a manual segmentation over 100 images. In order to reduce the oversegmentation, step (8) builds a spatial adjacency graph for the segmented regions and applies the merging predicate that tests if two regions C_1 and C_2 of which $C_1 \cup C_2$ come from the same statistical region of a perfect scene [13]. The perfect scene is not known however it is assumed that each pixel is represented by a set of distributions from which each observed colour channel i.e. image to segment, is sampled. The assumptions underlying the merging predicate of a perfect scene are:

- The pixels from the same region have the same expectation for each separate colour channel.
- The expectations of adjacent regions differ by at least one colour channel.

The merging predicate is:

$$P(C_1, C_2) = \left\{ \begin{array}{ll} \text{true} & \text{if } |\overline{C_2} - \overline{C_1}| \leq \sqrt{b^2(C_1) + b^2(C_2)} \\ \text{false} & otherwise \end{array} \right\}, \tag{3}$$

$$b(C) = g\sqrt{1/(2Q\,|C\,|))\ln(|C_{|C|}\,|/\delta)},$$

where each colour channel belongs to the set {1,2,…,g}, Q is the number of random variables representing each colour channel making it a parameter to quantify the statistical complexity of the perfect scene. $|\bullet|$ stands for cardinal and δ is the probability error.

5 Experimental Results

We have designed our experiments to compare the effectiveness of the proposed method with the existing methods. First we demonstrate that the preprocessing step improves the performance of our method without affecting the segmentation results. We applied the preprocessing step on three real videos with 120×100, 352×240 and 240×250 frame size respectively. We will first verify the reduction in processing time by applying the preprocessing step of the graph-based segmentation presented in section 3 on the first 50 frames extracted from each video. All experiments were performed on Pentium M processor 1.5GHz laptop. Fig. 2 (a), (b) and (c) shows the average number of graph edges sorted, the average processing time in milliseconds and the average number of regions obtained using different threshold values. Figure 2 (d) shows the average error of the difference obtained in the number of regions for the threshold values. In all experiments, the segmentation parameters were fixed except for the threshold values.

Second, we demonstrate the enhanced performance of our method in terms of speed and number of segments obtained. Fig. 3(b) demonstrates the over segmentation of the graph algorithm by comparing the results of 160 images manually segmented to the results obtained from graph segmentation. Fig. 3(c) demonstrates the enhanced speed of the algorithm: it is almost twice the speed of the original algorithm.

Finally, we compare quantitatively and qualitatively the results of our method with F&H method and SRM. In Table 1, we show the results of different empirical evaluation methods [14] that have been applied to 100 images segmented manually, with F&H algorithm, SRM algorithm, and our method. The table shows the reduced average number of regions obtained by the statistical region merge. The third column shows average intra-region uniformity where small values indicate better region uniformity. The fourth column shows that our method has the highest average inter-region contrast (adequate segmentation should have higher contrast across adjacent regions). F&H algorithm concentrates on keeping regions' internal variances less than their external variances at any moment which makes region uniformity measure and contrast to be good tools for accessing the quality of the segmentation. Fig. 4 shows some of the segmented images.

We experimentally show that the error incurred from the preprocessing step is very small and generally improves the segmentation result when incorporated with statistical region merge. Furthermore, the statistical region merge reduced the effect of over-segmentation and enhanced the quality of segmentation.

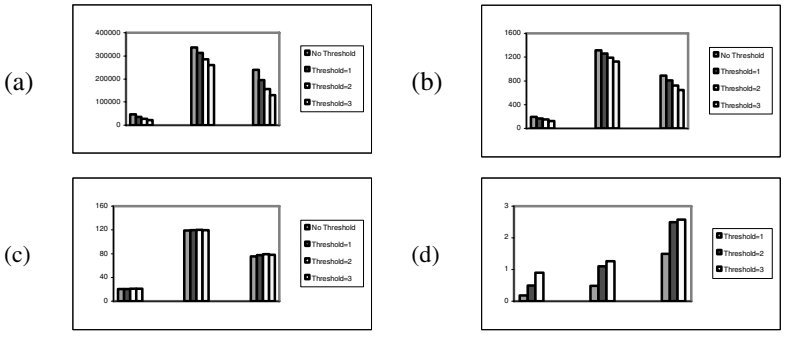


Fig. 2. Comparison of results using different threshold values for the three video clips, 'No Threshold' indicates F&H algorithm. (a) Average number of graph edges sorted (b) Average speed in millisecond (c) Average number of regions (d) Average error in the number of regions.

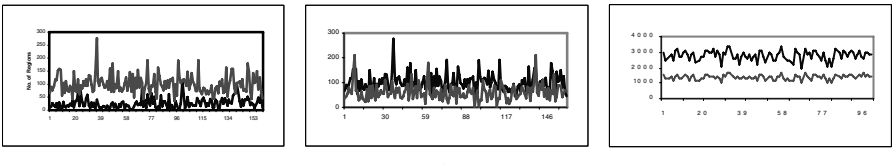


Fig. 3. (a) Regions obtained from F&H algorithm (oversegmentation) compared with manual segmentation (b) Number of Regions obtained from F&H algorithm compared with out method (c) Speed comparison of F&H algorithm and our algorithm in milliseconds over 160 images

Table 1. Results of applying empirical evaluation methods. GU: intra-region uniformity, GC: inter-region contrast, ND: discrepancy based on the position of mis-segmented pixels, D-PE: discrepancy based on the number of mis-segmented pixels.

Average of	Regions	GU	GC	ND	D-PE
F&H	100.59	0.888	1.651	0.587	25.52
SRM	46.55	0.692	1.828	0.662	39.30
Our Method	61.37	0.803	1.885	0.571	28.75



Fig. 4. Segmented Images

References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence. 22 (2000) 888-905
2. E. Sharon, A. Brandt, and R. Basri: Fast multiscale image segmentation. In IEEE Conference on Computer Vision and Pattern Recognition. 1 (2000) 70-77
3. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In IEEE Conf. on Computer Vision and Pattern Recognition. 2(2005) 1124-1131
4. L. Macaire, N. Vandenbroucke, and J.-G. Postaire: Color image segmentation by analysis of subset connectedness and color homogeneity properties. Computer Vision and Image Understanding. 102 (2006) 105-116
5. Y. Deng, B. S. Manjunath, and H. Shin: Color Image Segmentation. In IEEE Conference on Computer Vision and Pattern Recognition. 2 (1999) 446-451
6. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (2002) 603 – 619
7. H. Sun, J. Yang, and M. Ren: A fast watershed algorithm based on chain code and its application in image segmentation. Pattern Recognition Letters. 26 (2005) 1266-1274
8. R. Adams and L. Bischof: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence. 16 (1994) 641-647
9. F. Y. Shih and S. Cheng: Automatic seeded region growing for color image segmentation. Image and Vision Computing. 23 (2005) 877-886
10. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. Int. J. of Computer Vision 59 (2004) 167-181
11. Haxhimusa, Y., Kropatsch, W.: Segmentation graph hierarchies. Joint IAPR Int. Workshops SSPR and SPR, Springer-Verlag, Berlin Heidelberg (2004) 343-351
12. Forsyth, D.A., Ponce, J.: Computer vision a modern approach. Prentice Hall (2003)
13. Nock, R., Nielsen, F.: Statistical region merge. IEEE Trans. on Pattern Analysis and Machine Intelligence 26 (2004) 1452-1458
14. Zhang, Y. J.: A survey on evaluation methods for image segmentation. Pattern Recognition 29 (1996) 1335-1346

Sensor Fusion Based Obstacle Detection/Classification for Active Pedestrian Protection System

Ho Gi Jung[1,2], Yun Hee Lee[1], Pal Joo Yoon[1], In Yong Hwang[1], and Jaihie Kim[2]

[1] MANDO Corporation Central R&D Center
413-5, Gomae-Dong, Giheung-Gu, Yongin-Si, Kyonggi-Do 446-901, Republic of Korea
{hgjung, p13468, pjyoon, iyhwang}@mando.com
[2] Yonsei University, School of Electrical and Electronic Engineering
134, Sinchon-Dong, Seodaemun-Gu, Seoul 120-749, Republic of Korea
{hgjung, jhkim}@yonsei.ac.kr

Abstract. This paper proposes a sensor fusion based obstacle detection/classification system for active pedestrian protection system. At the front-end of vehicle, one laser scanner and one camera is installed. Clustering and tracking of range data from laser scanner generate obstacle candidates. Vision system classifies the candidates into three categories: pedestrian, vehicle, and other. Gabor filter bank extracts the feature vector of candidate image. The obstacle classification is implemented by combining two classifiers with the same architecture: support vector machine for pedestrian and vehicle. Obstacle detection system recognizing the class can actively protect pedestrian while reducing false positive rate.

1 Introduction

There are two explicit trends of CW(Collision Warning)/CA(Collision Avoidance) development: driver behavior model [1][2][3] and pedestrian protection [4][5]. Because these factors use the information of driving situation to reduce annoying false positive action while achieving fast response and reliable operation, they could be understood from the viewpoint of situation awareness. Driver monitoring system observes driver and measures the probability of whether driver perceive upcoming dangerous situation or not. Driver's perception status is used to modify the risk assessment estimated by time to collision [2]. Driver behavior model can modify the driving situation criticality level by considering the potential risk of collision and the adequacy of the driver's behavior [3]. Pedestrian protection system, which protects vulnerable road user, i.e. pedestrian, from traffic accident, is thought to be the most efficient and important method to reduce the fatalities of traffic accident [4]. Intelligent night vision is a kind of pedestrian protection system providing enhanced forward image in nighttime and recognizing pedestrian automatically [5]. Therefore, next generation CW/CA system is expect to not only detect forward obstacle but also recognize its class. Obstacle classification makes it possible for CW/CA system to consider the kind of obstacle. Especially, active safety system is expected to protect pedestrian more actively. This paper is related with forward obstacle detection and classification.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 294 – 305, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Vehicle detection has been one of the most important sensing problems in active safety system and driver assistance system. Sun's recent survey successfully summarizes up-to-date research activities on vehicle detection [6]. He analyzes vehicle detection process into two phases: HG(Hypothesis Generation) and HV(Hypothesis Verification). For the hypothesis generation phase, three kinds of methods are listed: knowledge based methods, stereo based methods, and motion based method. For the hypothesis verification phase, two kinds of methods are listed: template based method, and appearance based method. Recently, almost vehicle detection systems adopt the appearance based method, which classifies the hypotheses into 'vehicle' class or 'non-vehicle' class. Among them, SVM(Support Vector Machine) with Gabor filter is reported to produce the best performance [7].

Comparatively, pedestrian detection has shorter history, but is supposed to be the most critical technology for the reduction of traffic accident fatality [8]. Pedestrian detection method can be classified into three categories: range sensor based methods, vision based methods, and sensor fusion based methods.

Pedestrian detection with range sensor detects clusters of neighboring range data, then determines whether the clusters are corresponding to 'pedestrian' or not based on several kinds of attributes. Motion and width of cluster tracked temporally can be fed into pattern classifier [9]. With mm-wave radar, relation between range distribution and speed distribution is used for obstacle classification [10]. In other words, pedestrian has little range distribution, but large speed distribution compared to vehicle and stationary object.

Vision based pedestrian can be classified into three categories again: rhythmic motion based method, contour matching based method, and appearance based method. Cristóbal Curio verifies the candidates by checking the periodic motion caused from gait [11]. D. M. Gavrila developed contour matching based pedestrian detection. His chamfer system proposed a tree structure of pedestrian contours and distance transform based matching [12]. Appearance based pedestrian detection is similar to the appearance based vehicle detection. A. Broggi used morphological characteristics and the strong vertical symmetry of the human shape [13]. Liang Zhao used stereo based segmentation and neural network based recognition [14]. As proved in vehicle detection, SVM is supposed to be the best classifier also in pedestrian detection. Stereo based segmentation, edge features, and SVM classifier showed good classification performance and efficient computation possibility [15]. As in vehicle detection case, wavelet feature and SVM classifier seem to produce the best performance and robustness [16][17].

Sensor fusion based methods focus on the reduction of HG workload and the increase of hypothesis reliability. Therefore, they generally select sequential data fusion method [18]. Range sensor efficiently narrows the search range, although it is not complete, such that real time implementation becomes possible. Actually, to some extent, sensor redundancy is required to meet the reliability needs in automotive field [19].

As shown in Fig. 1, we select sequential sensor fusion based method to meet real time requirement. Laser scanner provides range data including true 'vehicle' and 'pedestrian', which are confused with noise. Range data clustering and tracking find potential obstacle eliminating noise to some extent. Two pattern classifiers are designed: one for vehicle detection, and the other for pedestrian detection. These pattern

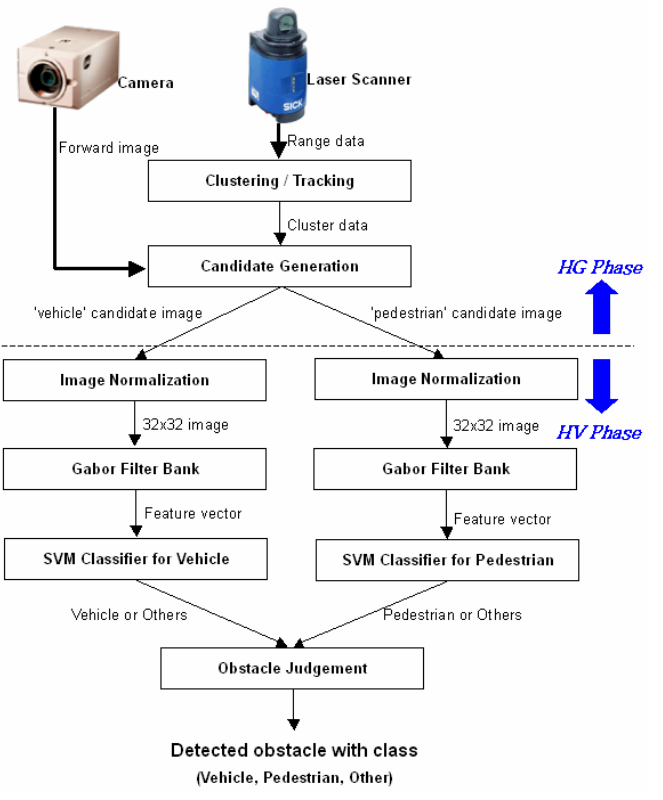


Fig. 1. Architecture of proposed sensor fusion based obstacle detection/classification system

classifiers are 'two class problem' solvers, i.e. true or not problem. Each classifier is implemented by SVM with Gabor filter bank, which is proved successful in various applications. At the decision phase, the outputs of two classifiers are simply integrated to determine the class of obstacle. In ambiguous situation, when vehicle classifier and pedestrian classifier output positive result simultaneously, the pedestrian classifier is selected to protect pedestrian actively. In this case, the system should accept the increase of false alarm rate. However, it can be justified by the fact that total error rate is sufficiently low and accident with pedestrian causes a fatal result. 'Other' class means a false obstacle caused by laser scanner noise or roadside object. Experiments show that proposed system can detect and classify obstacles successfully. Because Gabor filter bank - SVM is suitable for hardware implementation and the same architecture is used for two obstacle classes, our proposed system is expected to be a practical solution for mass production.

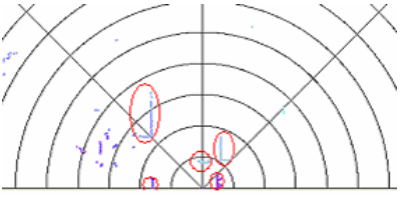
2 Candidate Generation

Laser scanner outputs range data, which are measured distance to objects with respect to azimuth angle. Objects can be identified by detecting clusters in the range data.

Fig. 2(b) shows the result of clustering adjacent data points with pre-defined distance, e.g. 0.3m. Detected clusters are tracked by Kalman filter to ensure the robustness of cluster detection. Assigned individual identification enables temporal signal processing. Distance, azimuth angle, changing rate of distance, changing rate of azimuth angle and cluster width are used as the state variables of Kalman filter.

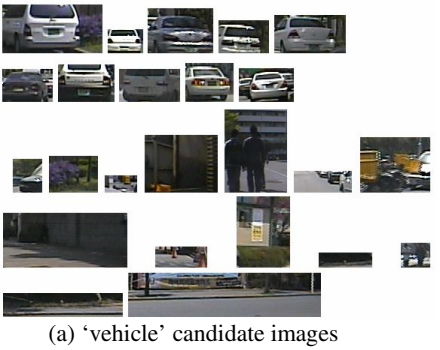


(a) Forward image from camera



(b) Range data from laser scanner and clusters

Fig. 2. Forward image and range data with detected clusters



(a) 'vehicle' candidate images



(b) 'pedestrian' candidate images

Fig. 3. Generated candidate images. In 'vehicle' candidate images, there are correct and incorrect cases. In 'pedestrian' candidate images, there are correct and incorrect cases.

Candidate image corresponding to each tracked cluster is extracted. Polar coordinate (r,θ) of recognized cluster can be converted easily to world coordinate $\boldsymbol{P_W}$ (X_W, Y_W, Z_W). Z_W plane is assumed to be the ground surface. Image coordinate $\boldsymbol{P_I}$ (X_I, Y_I) corresponding to the world coordinate $\boldsymbol{P_W}$ can be derived by perspective projection model. An image portion with the width of corresponding cluster and specific height is extracted as a candidate image. The specific height is 1.5m for vehicle and 2m for pedestrian. Therefore, one vehicle candidate image and one pedestrian candidate image are created for each cluster. Since each cluster creates two candidate images with two specific heights, vehicles or pedestrians with unusual height may hinder the following pattern recognition. However, the alignment of an obstacle may be maintained constantly since the height of extracted candidate image is determined from the ground surface. Fig. 3 shows the example of generated obstacle candidate image. Incorrect cases are caused by street side objects and laser scanner noise. The roles of classifiers are determining whether the candidate is correct or not.

Since segmented candidate images have various size and shape depending on width and distance, the images are required to be converted into uniform size. Both vehicle recognition module and pedestrian recognition module use 32x32 image as input. Fig. 4 illustrates the result of 'vehicle' candidate image normalization and 'pedestrian' candidate image normalization.



Fig. 4. Candidate image normalization procedure

3 Feature Vector Construction

Gabor filter is product of two-dimensional Gaussian function and sine wave [20]. Gabor filter can measure the image power of a specific frequency and direction at a specific location. σ_x and σ_y represent standard deviation of x-axis and y-axis. W is the frequency of spatial wave. Direction of sine wave is determined by rotating axis by θ. The definition of Gabor filter in frequency domain is shown in equation (2).

$$g(x,y)=\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x{}^2}+\frac{y^2}{\sigma_y{}^2}\right)+2\pi jWx\right] \tag{1}$$

$$G(u,v)=\exp\left[-\frac{1}{2}\left(\frac{(u-W)^2}{\sigma_u{}^2}+\frac{v^2}{\sigma_v{}^2}\right)\right] \tag{2}$$

Gabor filter bank is a group of Gabor filters with various shapes in a frequency domain. Since Gabor filter functions as band-pass filter for different frequencies, it is commonly used to create feature vector of images. We follow Manjunath's approach: equally dividing frequency space in phase angle and logarithmically dividing frequency space in magnitude direction [21].

Every filter is designed for its half height boundary to meet neighboring filters' boundaries. Our system uses 4 scales and 6 orientations: S(scale number)=4, K(orientation number)=6. Once u-axis directional standard deviation σ_u can be derived for a Gabor filter located on u-axis, other cases can be derived by rotating the result. In u-axis direction, average value and standard deviation of sth Gaussian function located on u-axis, G_s, are in the manner of geometrical progression with the average value and standard deviation of the first Gaussian function as shown in equation (3). U_l represents the lowest frequency, and U_h represents the highest frequency. Multiplication factor in logarithmic scale, a, is defined as in equation (4), which represents the frequency ratio of Gabor filters consecutive in scale.

$$G_s(u) \sim N(U_s, \sigma_u) = N\left(U_l a^{s-1}, \sigma_0 a^{s-1}\right) \tag{3}$$

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{S-1}} \quad \text{or} \quad \ln a = \frac{1}{S-1}\left(\ln U_h - \ln U_l\right) \tag{4}$$

Average value of sth Gaussian function located on u-axis, U_s, is defined in equation (5). Its u-axis standard deviation, σ_u, and v-axis standard deviation, σ_v, is defined in equation (6) and (7) respectively [21].

$$U_s = U_l \cdot a^{s-1} \tag{5}$$

$$\sigma_u = \frac{(a-1)U_s}{(a+1)\sqrt{2\ln 2}} \tag{6}$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right)\frac{\left(U_s - 2\ln 2\frac{\sigma_u^{\,2}}{U_s}\right)}{\sqrt{2\ln 2 - (2\ln 2)^2\frac{\sigma_u^{\,2}}{U_s^{\,2}}}} \tag{7}$$

The direction of 2-dimensional wave is set by substituting (x,y) by (x',y'), which is rotated to the kth section of K equal angular sections like equation (8). Especially, Gabor wavelet is created by multiplying the magnitude with a^{-s} to divide the power of high-pass filter into two child-filters recursively. Figure 5 shows the designed 24 Gabor filters in frequency domain. It is learned that because the adjacent Gabor filters contact each other at half height to minimize the information overlapping, each of designed filters functions as a band-pass filter for a specific frequency region [21].

In order to overcome segmentation deviation, only the statistical characteristic of Gabor filtering results of superposed sub-windows are used for the recognition process. Features are extracted for 9 overlapped sub-windows of candidate image. Candidate image with 32x32 size is divided into 9 sub-windows with 16x16 size. Each 16x16 sub-window overlaps a half of each other. Therefore, even if there is alignment deviation after the creation of candidate images, the extracted feature will bring similar results. The feature vector for recognition consists of mean, standard deviation, and skewness of convolutions between 9 sub-windows and 24 Gabor filters. Therefore, the dimension of feature vector is 648(=9x24x3).

$$g_s(x,y) = a^{-s}\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^{\,2}} + \frac{y'^2}{\sigma_y^{\,2}}\right) + 2\pi j U_s x'\right] \tag{8}$$

$$where, \begin{cases} x' = x\cos\left(\frac{k\pi}{K}\right) + y\sin\left(\frac{k\pi}{K}\right) \\ y' = -x\sin\left(\frac{k\pi}{K}\right) + y\cos\left(\frac{k\pi}{K}\right) \end{cases}, \ \sigma_x = \frac{1}{2\pi\sigma_u}, \ \sigma_y = \frac{1}{2\pi\sigma_v}$$

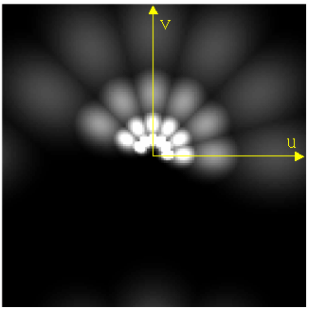


Fig. 5. Designed Gabor filter bank. 24 Gabor filters are overlapped in frequency domain.

4 Support Vector Machine Based Classification

SVM is a kind of classification method with hyperplane [22]. Decision boundary is defined by a *N*-dimensional vectors **w** and *b* defining a hyperplane as shown in equation (9). In the case of two-class problem, the decision function is defined by the sign of the projection of input vector **x** on the hyperplane as shown in equation (10).

$$(\mathbf{w}\cdot\mathbf{x}) + b = 0 \qquad , where\ \mathbf{w} \in \mathcal{R}^N, b \in \mathcal{R} \tag{9}$$

$$f(\mathbf{x}) = sign((\mathbf{w}\cdot\mathbf{x}) + b) \tag{10}$$

SVM assumes the maximum separating margin between two classes is the optimal condition. There are many hyperplanes dividing two classes. An optimal hyperplane should be designed as far as possible from two classes simultaneously. If margin m is defined as the minimal distance between two classes and hyperplane, designing optimal hyperplane is equal to maximizing m. If each of two classes has label +1 and –1, m can be expressed with the norm of $\mathbf{w}$ as illustrated in Fig. 6 and equation (11).

$$m = \frac{2}{\|\mathbf{w}\|} \tag{11}$$

SVM training can be explained by constrained optimization problem [23]. A data set $\{\mathbf{x_1},\ldots,\mathbf{x_n}\}$ and class label $y_i \in \{-1,1\}$ for $\mathbf{x_i}$ are known, decision boundary must satisfy $y_i(\mathbf{w}\ \mathbf{x_i}+b) \geq 0$ for all data. Maximizing m is equal to minimizing the norm of $\mathbf{w}$ as deduced from equation (11). Therefore, SVM training is the same as solving constrained optimization problem, equation (12).

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} && y_i \cdot (\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1 \quad \forall i \end{aligned} \tag{12}$$

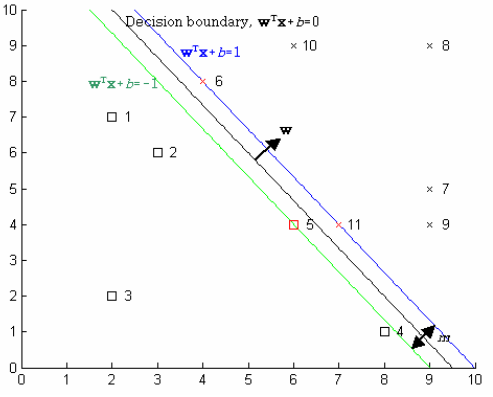


Fig. 6. Separation margin *m* and decision boundary

An optimization problem with multiple inequality constraints can be solved using Lagrange multiplier. If data $\mathbf{x_i}$, label y_i are known, SVM training is the same as solving constrained optimization problem with respect to $W(\boldsymbol{\alpha})$ as in equation (13). Here, $\boldsymbol{\alpha}$ is Lagrangian coefficient vector for every constraint. Since the problem is a type of QP(Quadratic Programming) problem, various QP problem tools can be used. Once $\boldsymbol{\alpha}$ is derived, **w** can be derived by equation (14). Therefore, *b* can be derived by utilizing data on the boundary. Because only data on decision boundary, i.e. support vector, have non-zero coefficients and contribute to boundary equation, decision function can be expressed only by support vectors. Nonlinear decision boundary can be learned by introducing kernel function.

$$\begin{aligned} &\text{maximize} && W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ &\text{subject to} && \alpha_i \geq 0, \qquad \sum_{i=0}^{n} \alpha_i y_i = 0 \end{aligned} \tag{13}$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{14}$$

The SVM of vehicle classifier is trained using 1,000 normalized images. 50% of the images are true images, and the others are false images. 648(=9x24x3) dimensional feature vector is acquired by applying 24 Gabor filters to 9 sub-windows and

estimating 3 statistic values. Used kernel is 1st-order linear function. The learning result using SVMlight [24] is shown in Table 1.

Table 1. Vehicle classifier learning result

Misclassified image	38/1000
Correct classification rate	96.2 %
False positive, P(Vehicle | Other)	5.6 %
False negative, P(Other | Vehicle)	2 %

The SVM of pedestrian classifier is trained using 1,200 normalized images. 700 of the normalized images are true images, and the others are false images. Used kernel is radial basis function. The learning result using SVMlight is shown in Table 2. It is noticeable that the system is able to recognize pedestrians of various sizes, colors and poses. Obstacles with various shapes and candidates from laser noise are correctly recognized as non-pedestrian images.

Table 2. Pedestrian classifier learning result

Misclassified image	63/1200
Correct classification rate	94.75 %
False positive, P(Pedestrian | Other)	8.8 %
False negative, P(Other | Pedestrian)	2.7 %

5 Obstacle Judgment

Each cluster of range data creates a 'vehicle' candidate image and a 'pedestrian' candidate image. Vehicle classifier and pedestrian classifier verify the images, respectively. Generally, one of two classifiers presents positive response or both classifier present negative responses. However, if both classifiers present positive responses, the system recognizes the candidate as pedestrian in order to be conservative. The correlation between classifier outputs and obstacle judgment is shown in Table 3.

Table 3. The correlation between classifier outputs and final decision

Pedestrian Classifier Output / Vehicle Classifier Output	Other	Pedestrian
Other	Other	Pedestrian
Vehicle	Vehicle	Pedestrian

6 Experiment Result

406 images that are not used for vehicle classifier learning are used to test performance of vehicle classifier. 50% of the images are 'vehicle' image, and the others are 'other'

images. The test result shows 95.07% of proper image recognition rate. Table 4 shows the test result of vehicle classifier. Fig. 7 shows correctly recognized candidate images. It is noticeable that various kinds of vehicles in different pose and size are successfully recognized.

Table 4. Test result of vehicle classifier

Misclassified image	20/406
Correct classification rate	95.07 %
False positive, P(Vehicle | Other)	5.4 %
False negative, P(Other | Vehicle)	5.4 %



(a) Correctly recognized vehicle images (b) Correctly recognized other images

Fig. 7. Candidate images correctly recognized by vehicle classifier

392 test images that are not used for pedestrian classifier learning are used to test performance of pedestrian classifier. 50% of the images are 'pedestrian' image, and the others are 'other' images. The test result shows 89.29% of proper image recognition rate. Table 5 shows the test result of pedestrian classifier. It is noticeable that correctly recognized candidate images include street crossing pose and street following pose.

Although developed recognition system successfully classifies vehicle and pedestrian in simple situations such as test track and inactive straight roadway, it fails to recognize in cluttered urban roadway because of complicated roadside objects. In such cases, clustering of range data becomes too noisy and leads to incorrect extraction of candidate image. These problems should be solved in the future.

Table 5. Test result of pedestrian classifier

Misclassified image	42/396
Correct classification rate	89.29 %
False positive, P(Pedestrian | Other)	12.7 %
False negative, P(Other | Pedestrian)	8.7 %



(a) Correctly recognized pedestrian images (b) Correctly recognized other images

Fig. 8. Candidate images correctly recognized by pedestrian classifier

7 Conclusion

This paper proposes a sensor fusion based obstacle detection and classification system. Obstacle's class information can be used for situation adaptive control system. Furthermore, it can reduce annoying false alarm rate of active pedestrian protection system such that users accept the new function as a practical daily tool. Experiments show that the same architecture can be applied for vehicle detection and pedestrian detection. Used Gabor filter bank and SVM are supposed to be suitable for hardware implementation. Our future works are improvement of range data clustering algorithm and implementation of system with FPGA to meet real-time performance.

References

1. Ardalan Vahidi and Azim Eskandarian, "Research Advances in Intelligent Collision Avoidance and Adaptive Cruise Control", IEEE Transaction on Intelligent Transportation Systems, Vol. 4, No. 3, Sep. 2003, pages: 143~153.
2. Akira Hattori, et al., "Development of Forward Collision Warning System Using the Driver Behavioral Information", SAE Paper No.: 2006-01-1462, The Society of Automotive Engineers, 2006.
3. Thierry Bellet, et al., "Driver behaviour analysis and adequacy judgement for man-machine cooperation: an application to anticollision", 5th European Congress and Exhibition on Intelligent Transportation Systems and Services, 1~3 June 2005.
4. Meinecke, et al., "SAVE-U: First Experiences with a Pre-Crash System for Enhancing Pedestrian Safety", 5th European Congress and Exhibition on Intelligent Transportation Systems and Services, 1~3 June 2005.
5. Takayuki Tsuji, Hideki Hashimoto, Nobuharu Nagaoka, "Intelligent Night Vision System-Nighttime Pedestrian Detection Assistance System", 12th World Congress on Intelligent Transport Systems, 6~10 November 2005.
6. Zehang Sun, George Bebis, and Ronald Miller, "On-Road Vehicle Detection: A Review", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 28, No. 5, May 2006, pages: 1~18.

7. Zehang Sun, George Bebis, and Ronald Miller, "On-Road Vehicle Detection Using Evolutionary Gabor Filter Optimization", IEEE Transaction on Intelligent Transportation Systems, Vol. 6, No. 2, June 2005, pages: 125~137.
8. Dariu M. Gavrila, "Sensor-Based Pedestrian Protection", IEEE Intelligent Systems, Vol. 16, Issue 6, Nov.-Dec. 2001, pages: 77-81.
9. Kay Ch. Fuerstenberg, and Jochen Scholz, "Reliable Pedestrian Protection Using Laserscanners", 2005 IEEE Intelligent Vehicles Symposium, 6-8 June 2005, Pages: 142-146.
10. Florian Fölster, Hermann Rohling, and Marc-Michael Meinecke, "Pedestrian Recognition based on automotive radar sensors", 5th European Congress and Exhibition on ITS and Services, 1-3 June 2005.
11. Cristóbal Curio, et al., "Walking Pedestrian Recognition", IEEE Transaction on Intelligent Transportation Systems, Vol. 1, No. 3, September 2000, pages: 155-163.
12. D.M. Gavrila, "Pedestrian Detection from a Moving Vehicle", LNCS 1843 (ECCV 2000), 2000, pages: 37-49.
13. A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, "Shape-based Pedestrian Detection", IEEE Intelligent Vehicles Symposium 2000, October 3-5, 2000, pages:215-220.
14. Liang Zhao, and Charles E. Thorpe, "Stereo- and Neural Network-Based Pedestrian Detection", IEEE Transaction on Intelligent Transportation Systems, Vol. 1, No. 3, September 2000, pages: 148-154.
15. Grant Grubb, et al., "3D Vision Sensing for Improved Pedestrian Safety", IEEE Intelligent Vehicles Symposium 2004, June 14-17, 2004, pages: 19-24.
16. Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio, "Example-Based Object Detection in Images by Components", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 23, No. 4, April 2001, pages:349-361.
17. Constantine Papageorgiou and Tomas Poggio, "A Trainable System for Object Detection", International Journal of Computer Vision, 38(1), 15-33, 2000, pages: 15-33.
18. Milch, S., Behrens, M., "Pedestrian Detection with Radar and Computer Vision", http://www.smart-microwave-sensors.de/html/publications.html, 2006.
19. L. Walchshäusl, et al., "Detection of Road Users in Fused Sensor Data Streams for Collision Mitigation", 10th International Forum on Advanced Microsystems and Automotive Applications (AMAA 2006), Berlin, Germany, April 25-26, 2006.
20. Javier R. Movellan, "Tutorial on Gabor Filters", http://mplab.ucsd.edu/tutorials/pdfs/gabor.pdf, 2006.
21. B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, August 1996, pages: 837-842.
22. Christopher J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 2, 121-167 (1998).
23. Martin Law, "A Simple Introduction to Support Vector Machines", http://www.cse.msu.edu/~lawhiu/intro_SVM_new.ppt, 2006.
24. Thorsten Joachims, "SVMlight: Support Vector Machine", http://svmlight.joachims.org, 2006.

Combinatorial Pyramids and Discrete Geometry for Energy-Minimizing Segmentation

Martin Braure de Calignon[1], Luc Brun[2], and Jacques-Olivier Lachaud[1]

[1] LaBRI CNRS UMR 5800
Université Bordeaux 1
351, cours de la Libération
33405 Talence cedex
{braure, lachaud}@labri.fr
[2] GreyC CNRS UMR 6072
Équipe Image - ENSICAEN
6, Boulevard du Maréchal Juin
14050 CAEN Cedex - France
luc.brun@greyc.ensicaen.fr

Abstract. This paper defines the basis of a new hierarchical segmentation framework based on an energy minimization scheme. This new framework is based on two formal tools. First, a combinatorial pyramid encodes efficiently a hierarchy of partitions. Secondly, discrete geometric estimators measure precisely some important geometric parameters of the regions. These measures combined with photometrical and topological features of the partition allow to design energy terms based on discrete measures. Our segmentation framework exploits these energies to build a pyramid of image partitions with a minimization scheme. Some experiments illustrating our framework are shown and discussed.

1 Introduction

The convergence of energy minimization and hierarchical segmentation algorithms provides a rich framework for image segmentation. This framework is based on an objective criterion, called *energy*, whose minimization defines a salient partition according to a given problem. The energy of a partition is generally decomposed by summation over each region R, where the energy is a weighted sum of two terms $E(R) = E_{img}(R) + \nu E_{reg}(R)$. E_{img} may be understood as a fit to the data within the region while E_{reg} corresponds to a regularization term. The parameter ν defines the respective weights of the two terms. The Mumford-Shah energy is a classical instance of this approach [1]. Such an equation may also be interpreted within the Minimum Description Length (MDL) framework [2], where the two energies E_{img} and E_{reg} represent respectively the encoding costs of the photometry and the geometry of a region.

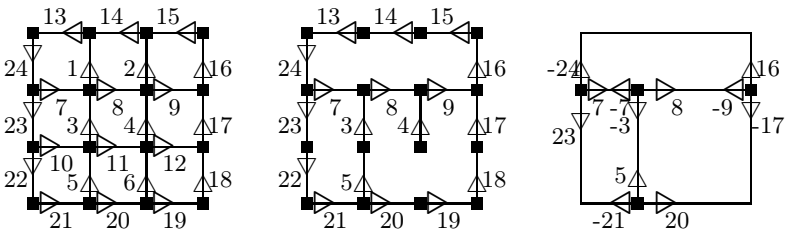
Several methods have been proposed in order to obtain a partition minimizing an energy. These methods include the level set approach [1], graph cuts [3] and the methods based on a region merging scheme [4,5,6,7]. The definition of a meaningful segmentation using an energy minimization framework and a merge

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 306–315, 2006.
© Springer-Verlag Berlin Heidelberg 2006

scheme supposes first to define a merge strategy. If the parameter ν is fixed, a near optimal strategy consists in merging at each step the two regions, the merging of which induces the greatest decrease of the energy until any merge would increase the energy. The obtained partition is said to be *2 normal* at the scale ν [4,5]. An alternative strategy [6] consists in merging at each step the two regions whose union would belong to the 2 normal partition of lowest scale. This reduction framework avoids the need to select a vector of ν parameters encoding *a priori* the different scales of interest. However, previous works [4,5,6] were based on a sequence of merge operations combined with a stopping criterion (number of regions, maximal value of $\nu \ldots$). Guigues *et. al.* [7] explicitly encode the hierarchy of partitions using a reduction scheme similar to [6] but uses the hierarchy in order to build for any value of ν, the optimal partition which may be defined from the hierarchy. Moreover, instead of starting from the grid of pixels like [6], their initial partition is an over partition of the image. It presents two advantages. First, the initial over segmented partition allows to compute reliable statistics on regions. Secondly, it restricts the set of possible partitions and thus reduces the risk to be trapped into a local minima.

The second problem that should be addressed by a segmentation algorithm is the correct design of the energy terms. For instance, the classical Mumford-Shah energy simply combines the squared error of each region together with the total length of the partition boundaries. However, as shown by several authors [7], more complex models (both geometrical and photometrical) may handle finer definitions of salient partitions. Their design requires to fit geometrical models onto regions. An efficient access to the set of boundaries of each region and to their geometry is thus compulsory. However, classical hierarchical segmentation frameworks are not adequate for this task. Adaptive pyramids based on graph [8] do not provide a 1-1 correspondence between the edges of the graphs and the geometrical boundaries: reconstructing the geometry of a region is then tricky. Dual graphs [9] behave better for this task but the explicit encoding of all reduced graphs restricts the number of merge steps.

This paper provides a new framework that addresses the design of new energy terms based on geometrical and photometrical features. The stack of successively reduced partitions is encoded using a combinatorial pyramid [10]. A very fine granularity for the hierarchy is then achieved since a new level of the pyramid is created for each merging of two regions. Geometrical features are computed on each partition of the hierarchy using discrete geometric estimators of normal and length. This framework offers then a compact and efficient encoding of the hierarchy together with an efficient access to the geometrical and topological properties of the partition. It came thus as a natural complement to methods searching for optimal partitions within a hierarchy. The paper is structured as follows. We present in Section 2 the combinatorial pyramid model. The application of this model to compute geometrical features on regions using discrete geometric estimators is presented in Section 3. We then present in Section 4 one energy based on discrete estimators together with some experiments.



(a) $\overline{G_0}$ (b) $\overline{G_1}$ contracted (c) $\overline{G_3}$ reduced

Fig. 1. A dual of a combinatorial map (a) encoding a 3×3 grid. The contracted combinatorial map (b) obtained after application of the contraction kernel (CK) $K_1 = \alpha^*(1, 2, 10, 11, 12, 6)$. The reduced combinatorial map (c) is obtained by the removal of the empty self loops $\alpha^*(4)$ and the removal kernel of empty double edges (RKEDE) $K_3 = \alpha^*(13, 14, 15, 19, 18, 22) \cup \{24, -16, 17, -20, 21, -23, 3, -5\}$. In order to avoid overloading of images, only positive darts are shown in (a) and (b).

2 Combinatorial Pyramids

Our approach is based on combinatorial maps [11]. A combinatorial map may be seen as a planar graph encoding explicitly the orientation of edges around a given vertex. To do so, each edge of a planar graph is split into two half-edges called *darts* (e.g. darts 16 and -24 in Fig. 1c). A combinatorial map is formally defined by a triplet $G = (\mathcal{D}, \sigma, \alpha)$ where $\mathcal{D}$ represents the set of darts and σ is a permutation on $\mathcal{D}$ whose cycles correspond to the sequence of darts encountered when turning counter-clockwise around each vertex. Finally α is an involution on $\mathcal{D}$ which maps each of the two darts of one edge to the other one (e.g. α maps 16 to -24 and -24 to 16 in Fig 1c). The cycles of α and σ containing a dart d will be respectively denoted by $\alpha^*(d)$ and $\sigma^*(d)$.

Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, its dual map is defined by $\overline{G} = (\mathcal{D}, \varphi, \alpha)$ with $\varphi = \sigma \circ \alpha$. The cycles of permutation φ encode the faces of the combinatorial map and may be interpreted as the sequence of darts encountered when turning clockwise around a face. The cycle of φ containing a dart d will be denoted by $\varphi^*(d)$.

2.1 Encoding the Image Grid with Combinatorial Map

Combinatorial maps can also code the low level geometry of image pixels. Indeed, Fig. 1a describes a dual combinatorial map $\overline{G_0} = (\mathcal{D}_0, \varphi_0, \alpha_0)$ encoding a 3×3 4-connected planar sampling grid. The φ, α and σ cycles of each dart may be respectively understood as elements of dimensions 0, 1 and 2 and formally associated to a 2D cellular complex [10]. More precisely, each α_0 cycle may be associated to a *linel* (sometimes also called *crack* or *surfel*) between two pixels. Each of the two darts of an α_0 cycle corresponds to an orientation along the linel. For example, the cycle $\alpha_0^*(1) = (1, -1)$ is associated to the linel encoding the right border of the top left pixel of the 3×3 grid (Fig. 1a). Darts 1 and

-1 define respectively a bottom to top and top to bottom orientation along the linel.

2.2 Construction of Combinatorial Pyramids

A combinatorial pyramid is defined by an initial combinatorial map successively reduced by a sequence of contraction or removal operations which are formal translations of region merges. Contraction operations are encoded by contraction kernels (CK). These kernels, defined as a forest of the current combinatorial map, may however create redundant edges such as empty-self loops and double edges. Empty self loops (edge $\alpha_1^*(4)$ in Fig. 1b) may be interpreted as region inner boundaries and are removed after the contraction step. The remaining redundant edges, called double edges, belong to degree 2 vertices in $\overline{G}$ (e.g. $\varphi_1^*(13)$, $\varphi_1^*(14)$, $\varphi_1^*(15)$) in Fig. 1b) and are removed using a removal kernel of empty double edges (RKEDE) which contains all darts incident to a degree 2 dual vertex. Further details about the construction scheme of a combinatorial pyramid may be found in [10].

2.3 Embedding of Region Boundaries

As mentioned in Section 2.1, if the initial combinatorial map encodes a planar sampling grid, the geometrical embedding of each initial dart corresponds to an oriented linel. Moreover, each dart of a reduced map that is not a self loop encodes a connected boundary between two regions. The embedding of such a boundary may be retrieved from the embedding of the darts of the initial map G_0. Let us consider the reduced combinatorial map $G_i = (\mathcal{D}_i, \sigma_i, \alpha_i)$ defined at level i and one dart $d \in \mathcal{D}_i$ which is not a self loop. The sequence $d_1 \ldots, d_n$ of initial darts encoding the embedding of the dart d is obtained using the following relation:

$$d_1 = d \, , d_{j+1} = \underbrace{\varphi_0 \circ \cdots \circ \varphi_0}_{m_j \text{ times}}(\alpha_0(d_j)) \tag{1}$$

where $\overline{G_0} = (\mathcal{D}_0, \varphi_0, \alpha_0)$ is the dual G_0 and m_j is the smallest integer q such that $\varphi_0^q(\alpha_0(d_j))$ survives at level i or belongs to some former RKEDE. The dart d_n is the first dart defined by Eq. (1) which survives up to level i. This dart also satisfies $\alpha_0(d_n) = \alpha_i(d)$ by construction of the receptive fields. Note that the tests performed on $\varphi_0^q(\alpha_0(d_j))$, $q \in \{1, \ldots, m_j\}$ to determine if it is equal to d_{j+1} or d_n are performed in constant time using the implicit encoding of combinatorial pyramids.

Let us consider the dart 16 in Fig. 1c. This dart encodes the border between the background and the first row of the 3×3 grid encoded by the σ_3 cycle $\sigma_3^*(16) = (16, 7, 8)$ of G_3. The sequence of initial darts encoding the boundary of the dart 16 is retrieved using Eq. (1) and is equal to: 16.15.14.13.24 (Fig. 1b). Since each initial dart is associated to an oriented linel, one may associate a sequence of Freeman's codes to each sequence of initial darts (Fig. 1b) and thus

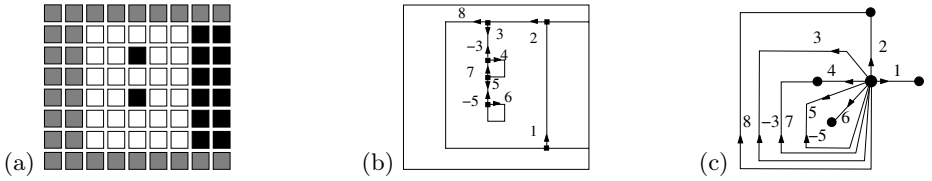


Fig. 2. The central white region $\sigma^*(1)$ (a) contains several subregions. Its boundary is thus split into several connected components connected by bridges in $\overline{G}$ (b). These edges correspond to self loops in G (c).

to each dart of a reduced combinatorial map G_i. The sequence of Freeman's codes associated to a dart d is called the *segment* associated to d (e.g.: the segment associated to the dart 16 is sequence $s_{16} = 1.2.2.2.3$).

3 Discrete Geometry over a Partition

As mentioned in Section 2, each edge $(d, \alpha_i(d))$ of a partition G that is not a self loop encodes a connected boundary between two regions and is called *separating*. On the other hand, a self loop corresponds to a bridge in the dual combinatorial map and is characterized by $\alpha_i(d) \in \sigma_i^*(d)$ (e.g. edge $(3, -3)$ or $(5, -5)$ in Fig. 2bc). Such edges, called *fictive*,

either connect the outer boundary to some inner boundary (e.g. edge $(3, -3)$ in Fig. 2) or connect two inner boundaries (edge $(5, -5)$ in Fig. 2).

Each separating edge is embedded as a 4-connected digital path, included in the interpixel digital plane (Section 2.3 and [10]).

When estimating the geometry of the boundary of the region, fictive edges do not play any role. More precisely the concatenation of only the separating edges defines a set of 4-connected digital loops. Each of these loops is either the outer boundary of the region or one of its inner boundaries. Given an initial dart d belonging to a separating edge, Algorithm 1 extracts a boundary between region $\sigma^*(d)$ and its complement (setting $L_{\text{in}} = \sigma^*(d)$) or between regions $\sigma^*(d)$ and $\sigma^*(d')$ and their complement (setting $L_{\text{in}} = \sigma^*(d) \cup \sigma^*(d')$). Its principle is to follow the boundary with σ except that it skips fictive edges and edges in-between $\sigma^*(d)$ and $\sigma^*(d')$. This method for tracking a boundary is easily understood on Fig. 2b, where for instance the algorithm tracks from dart 1, then $2 = \sigma(1)$, 3 is skipped since $-3 \in \sigma^*(1)$, then $8 = \sigma(-3)$ and terminates on $1 = \sigma(8)$ again. Extracting all the boundaries of a region is done in a similar way. All these algorithms can be implemented with a complexity linear with the number of boundary linels.

3.1 Geometry with Digital Straight Segments

We may now examine how geometric quantities can be estimated on a closed 4-connected digital contour C, which is some boundary of a region or two adjacent regions (computed as in the previous paragraph). We restrict ourselves

Algorithm 1. Algorithm to visit all the linels of the digital boundary encircling region(s) specified by their darts L_{in} and containing the dart d.

```
1 Function Map::boundary( dart d, darts L_in ) : Freeman chain
Ensure: Return a sequence of Freeman's codes that is a 4-connected loop.
Require: d ∉ L_in
2 list C ← ∅,   dart b ←d
  repeat
    C.append(s_b)
    b ←σ(b)
    while α(b) ∈ L_in do {Skip fictive or interior edges}
      b ←φ(b)
    end while
  until b = d
  return C
```

to pure discrete geometry tools based on *digital straight segment* (DSS) recognition. Several equivalent definitions of DSS exist together with several classes of algorithms to recognize them on digital curves (see for instance [12] for a recent survey). We chose here to present briefly the arithmetic point of view of digital lines, which leads to rather simple and efficient algorithms [13,14].

The set of points (x, y) of the digital plane verifying $\mu \leq ax - by < \mu + |a| + |b|$, with a, b and μ integer numbers, is called the *standard line* with slope a/b and shift μ. A standard line is always 4-connected. A sequence of consecutive points $C_{i,j}$ indexed from i to j of the digital curve C is a *digital straight segment (DSS)* iff there exists a standard line (a, b, μ) containing them. The one with smallest $a + b$ determines its *characteristics*, in particular its slope a/b. Any DSS Z thus defines an angle $\theta(Z)$ between its carrying standard line and the x-axis (in $[0; 2\pi[$ since a DSS is oriented), called the *direction of* Z.

The predicate *"$C_{i,j}$ is a DSS"* is denoted by $S(i, j)$. Incremental algorithms exist to recognize a digital straight segment on a curve and to extract its characteristics [13]. Therefore deciding $S(i, j + 1)$ or $S(i - 1, j)$ from $S(i, j)$ are $O(1)$ operations. Any DSS $C_{i,j}$ is called a *maximal segment* iff $\neg S(i, j + 1)$ and $\neg S(i-1, j)$. Maximal segments are thus the inextensible DSS of the curve (Fig. 3, left). Note that the set of all maximal segments of a curve can be computed in time linear with the number of curve points [14].

3.2 Tangent, Normal and Length Estimation

Several tangent estimators based on DSS recognition have been proposed. We propose to use the λ-Maximal Segment Tangent estimator (λ-MST) to approach the tangent direction at any point of the digital curve [15]. It was indeed shown to give good approximations even at coarse scale, to be rather independent from rotations and to be asymptotically convergent.

Fig. 3, right, gives the essential idea of this tangent estimator. Given a point, the direction θ_i of every maximal segment containing it is evaluated. The relative position e_i of the point within the maximal segment is also computed. The

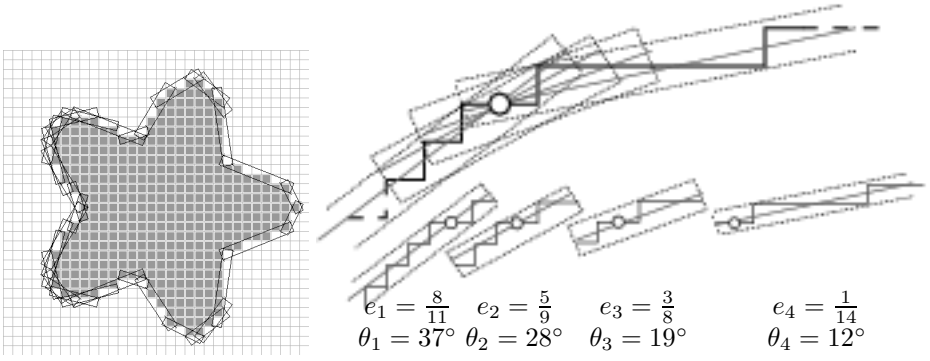


Fig. 3. Left: every maximal segment along this contour is drawn as its rectangular bounding box. Right: λ-Maximal Segment Tangent estimation at a given point.

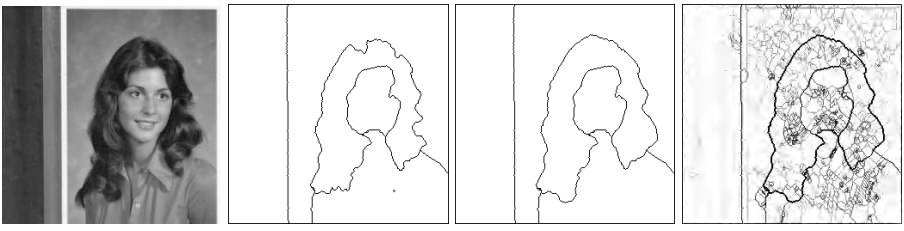
λ-MST tangent direction $\hat{\theta}$ is some weighted combination of the preceding parameters: $\hat{\theta} = (\sum_i \lambda(e_i)\theta_i)/(\sum_i \lambda(e_i))$. In our experiments, the mapping λ was defined as the triangle function taking base value 0 at 0 and 1, and peak value 1 at $\frac{1}{2}$. For further details, see [15].

The experimental average number of maximal segments per linel is between 3 and 4. Therefore computing the λ-MST direction is not costly and is a $O(1)$ operation on average. This technique of tangent estimation is easily extended to any real curvilinear abscissa along the digital contour. The tangent is thus defined at any linel, taking half integer abscissas.

The estimation of the *normal vector* at C_k is then simply the vector $\hat{\mathbf{n}}(k) = (-\sin(\hat{\theta}(k)), \cos(\hat{\theta}(k)))$. The *elementary length* $\hat{l}(k, k+1; C)$ of a linel $C_{k,k+1}$ is defined as $|\cos(\hat{\theta}(k+0.5))|$ for horizontal linels and $|\sin(\hat{\theta}(k+0.5))|$ for vertical linels. It corresponds to an estimation of the length of a unit displacement along the digital curve. The length of C is estimated by simple summation of the elementary length of its linels. This method of length evaluation was reported to give very good experimental results [16]. If C^b is boundary$(b, \sigma^*(b))$ as returned by Algorithm 1, then its *length* is $\hat{L}(b; G) = \sum_{k=1}^{|C^b|} \hat{l}(k, k+1; C^b)$. The total *perimeter* $\widehat{\mathrm{Per}}(R(\sigma^*(b)); G)$ of the region $\sigma^*(b)$ is the sum of the length of each of its boundaries.

4 Energy of a Partition and Pyramidal Segmentation

The geometrical features (normal, perimeter) defined in Section 3 may be computed on each region of a partition to provide different measures of its geometrical characteristics. Such measures may then be incorporated into a hierarchical segmentation algorithm based on an energy minimization scheme (Section 1). Such energy balances two terms: the goodness of fit term and a regularization term which penalizes unlikely or complex models. The energy of a partition encoded by the map G is simply called the *energy of the combinatorial map* G and



(a) (b) (c) (d)

Fig. 4. Influence of length penalization: (a) image Girl, (b) one level of the pyramid with $\hat{l}(k, k+1; C) = 1$, (c) same level of a pyramid built using discrete length estimators. All the boundaries of the pyramid which contains (c) are superimposed on (d). The darkest boundaries are those who survive at the highest levels.

is formally defined as follows: Let $G = (\mathcal{D}, \sigma, \alpha)$ be a combinatorial map with a geometrical embedding in the digital grid and an input image I over this grid. Let $\mathcal{D}_\sigma$ be the set of σ-cycles of $\mathcal{D}$. The energy of the combinatorial map G is

$$E(G) = \sum_{\sigma^*(d) \in \mathcal{D}_\sigma} E(\sigma^*(d)), \tag{2}$$

$$\text{with, } E(\sigma^*(d)) = E_{\text{img}}(\sigma^*(d)) + \nu E_{\text{reg}}(\sigma^*(d)). \tag{3}$$

Eq. (2) indicates that the global energy is decomposable over each region. This property helps in defining fast algorithms for region decimation. Eq. (3) balances the two energies, one dependent on the image (the image energy E_{img}), the other dependent only on the model (the regularization energy E_{reg}).

The parameter ν is often interpreted as a scale parameter, since it privileges the goodness of fit for low values (and over-segmentation) and *a priori* most likely regions for high values (and under-segmentation). The image energy used for our experiments is defined as follows:

$$E_{\text{img}}(\sigma^*(d)) = -\delta \sum_{C_k \in C^d} \|D\, I(C_k)\| \hat{l}(k, k+1; C^d) + \sum_{(x,y) \in R(\sigma^*(d))} \|I(x,y) - \mu_{\sigma^*(d)}\|^2 \tag{4}$$

where $\hat{l}(k, k+1; C^d)$ denotes the length estimate of a linel at point C_k, $I(x, y)$ denotes the color of the pixel (x, y) and $\|D\, I(C_k)\|$ the norm of the differential of I at point k. This last measure is equal to the norm of the gradient for grey level images. The term $\mu_{\sigma^*(d)}$ represents the mean color of the region encoded by $\sigma^*(d)$. The second sum of the above expression describes thus the squared error of the region fit. Finally, the term δ represents the respective weight of the gradient and squared error energies.

The regularization energy is defined from the estimate of the perimeter of the region as $E_{\text{reg}}(\sigma^*(d)) = \widehat{\text{Per}}(R(\sigma^*(d)); G)$ (Section 3.2). Given two possible merge operations inducing the same variation of the image energy, this choice for the regularization term favors the one which induces the simplest partition

with the lowest overall length of contours. The advantage of using discrete length estimators compared to a basic count of the linels is to make the segmentation process more independent on the alignment of components wrt some axes.

We tested the influence of length penalization on the classical Girl test image (Fig. 4). Two pyramids have been built on an initial partition encoded by a combinatorial map G_0. This initial partition is defined by a watershed algorithm applied on the gradient of the Girl test image. The parameter ν is fixed to 1.3 during the construction of both pyramids. Fig. 4(b) represents one significant level of the first pyramid built using a fixed length estimate equal to 1 for all linels, as classically done by many authors [7][3]. Fig. 4(c), represents the equivalent level in the second pyramid built using the discrete length estimator defined in Section 3.2. As shown by Fig. 4(c) the more accurate measure of the length given by the discrete length estimator provides the smoothest boundaries. Fig. 4(d) shows that the most significant regions survive at highest levels in the pyramid (darkest boundaries). So that pyramidal segmentation provides a multi-scale image segmentation.

Pyramidal Segmentation Algorithm

Our energy minimization method starts with an initial partition coded by a map, and merges at each step the two adjacent regions, the merging of which induces the greatest decrease (or the smallest increase) of the combinatorial map energy. This process may be interpreted as a gradient descent which continues when a local minima is reached in order to seek other minima. Note that our framework is not devoted to a specific strategy for energy minimization. Many alternative optimization heuristics could be used (e.g. the scale-climbing of Guigues *et. al.* [7]). The proposed approach is however sufficient to compare the respective advantages of different energies. Let us additionally note that using our strategy or the scale climbing of Guigues *et. al.*, only two regions are merged between two consecutive levels of the pyramid. This merge strategy does not induce a high memory cost due to the implicit encoding of the combinatorial pyramid ($O(N)$ for an image with N pixels). An explicit construction of all the reduced graphs using graph or dual graph pyramids would require a huge amount of memory with a lot of redundancy between graphs ($O(N^2)$).

5 Conclusion

We have presented a new framework for segmenting images with a pyramidal bottom-up approach using an energy-minimizing scheme. Our framework combines combinatorial pyramids, which can represent in the same structure all the levels of a hierarchy, and discrete geometric estimators, which provide precise geometric measurements and allow the definition of new regularization and image energy terms. A greedy algorithm for computing the hierarchy was also provided and some examples of segmentation were exhibited and discussed.

Our first experiments show that the length estimation can have a great influence on the regularization of the segmentation. Discrete geometric estimators

provide some smooth boundaries. However, they are useless if the over-segmentation gives irregular regions. In future works, we want to tackle this problem by using a smoother over-segmentation.

References

1. Chan, T.F.: Active contours without edges. IEEE Transactions on image Processing **10**(2) (2001) 266–277
2. Leclerc, Y.G.: Constructing simple stable descriptions for image partitioning. International Journal of Computer Vision **3**(1) (1989) 73–102
3. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on PAMI **26**(19) (2004) 1124–1137
4. Koepfler, G., Lopez, C., J.-M., M.: A multiscale algorithm for image segmentation by variational method. SIAM J. Numerical Analysis **31**(1) (1994) 282–299
5. Morel, J.M., Solimini, S.: Variational methods in image segmentation. Volume 14 of Progress in Nonlinear Differentiel Equations and Their Applications. Birkhäuser, Boston, Basel, Berlin (1995)
6. Reddings, N.J., Crisp, D.J., Tang, D.H., Newsam, G.N.: An efficient algorithm for mumford-shah segmentation and its application to sar imagery. In: Proc. Conf. Digital Image Computing Techniques and applications (DICTA). (1999) 35–41
7. Guigues, L., Cocquerez, J., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision, **68**(1) (2006.) 289 – 317
8. Jolion, J.M.: Data driven decimation of graphs. In Jolion, J.M., Kropatsch, W., Vento, M., eds.: Proceedings of 3^{rd} IAPR-TC15 Workshop on Graph based Representation in Pattern Recognition, Ischia-Italy (2001) 105–114
9. Kropatsch, W.G., Haxhimusa, Y., Lienhardt, P.: Hierarchies relating topology and geometry. In Christensen, H., Nagel, H.H., eds.: Proceedings of Cognitive Vision Systems (Seminar N° 03441 Dagstuhl). (2003)
10. Brun, L., Kropatsch, W.: Contains and inside relationships within combinatorial pyramids. Pattern Recognition **39**(4) (2006) 515–526
11. Lienhardt, P.: Topological models for boundary representations: a comparison with n-dimensional generalized maps. Computer-Aided Design **23**(1) (1991) 59–82
12. Klette, R., Rosenfeld, A.: Digital Geometry - Geometric Methods for Digital Picture Analysis. Morgan Kaufmann, San Francisco (2004)
13. Debled-Renesson, I., Réveillès, J.P.: A linear algorithm for segmentation of discrete curves. International Journal of Pattern Recognition and Artificial Intelligence **9** (1995) 635–662
14. Feschet, F., Tougne, L.: Optimal time computation of the tangent of a discrete curve: Application to the curvature. In: Proc 8th Int. Conf. Discrete Geometry for Computer Imagery (DGCI'99). Number 1568 in Lecture Notes in Computer Science, Springer Verlag (1999) 31–40
15. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Analysis and comparative evaluation of discrete tangent estimators. In et al., E.A., ed.: Proc. Int. Conf. Discrete Geometry for Computer Imagery (DGCI'2005), Poitiers, France. Volume 3429 of LNCS., Springer (2005) 140–251
16. Coeurjolly, D., Klette, R.: A comparative evaluation of length estimators of digital curves. IEEE Trans. on Pattern Anal. and Machine Intell. **26**(2) (2004) 252–257

Fast Dense Stereo Matching Using Adaptive Window in Hierarchical Framework

SangUn Yoon, Dongbo Min, and Kwanghoon Sohn

Dept. of Electrical and Electronic Eng., Yonsei University
134 Shinchon-dong, Seodaemun-gu, Seoul, 120-749, Korea
khsohn@yonsei.ac.kr

Abstract. A new area-based stereo matching in hierarchical framework is proposed. Local methods generally measure the similarity between the image pixels using local support window. An appropriate support window, where the pixels have similar disparity, should be selected adaptively for each pixel. Our algorithm consists of the following two steps. In the first step, given an estimated initial disparity map, we obtain an object boundary map for distinction of homogeneous/object boundary region. It is based on the assumption that the depth boundary exists inside of intensity boundary. In the second step for improving accuracy, we choose the size and shape of window using boundary information to acquire the accurate disparity map. Generally, the boundary regions are determined by the disparity information, which should be estimated. Therefore, we propose a hierarchical structure for simultaneous boundary and disparity estimation. Finally, we propose post-processing scheme for removal of outliers. The algorithm does not use a complicate optimization. Instead, it concentrates on the estimation of a optimal window for each pixel in improved hierarchical framework, therefore, it is very efficient in computational complexity. The experimental results on the standard data set demonstrate that the proposed method achieves better performance than the conventional methods in homogeneous regions and object boundaries.

1 Introduction

Stereo matching is one of the classical problems in computer vision. There are many potential applications including the robot navigation, teleconferencing, 3D modeling, and image based rendering, etc. The stereo matching problem is to compute the disparity map for the reference image using two or more images of the same scene. In binocular stereo, it is assumed that two input images are calibrated and rectified in advance, so that epipolar line becomes horizontal, but stereo matching is a difficult problem due to ambiguous region such as the occluded and textureless areas. To solve this problem, a number of algorithms have been proposed [1].

According to a recent taxonomy [1], stereo matching method can be roughly classified into two categories, namely, global and local methods. Global method define the smoothness constraint in order to resolve the ill-posed problem of

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 316–325, 2006.
© Springer-Verlag Berlin Heidelberg 2006

stereo matching, so that the problem of homogeneous regions can be handled successfully. Moreover, the discontinuity preserving smoothness constraint has been employed for the stereo model, and the energy function is minimized through various minimization techniques such as regularization method[9] and graph cut [11][12]. However, global methods have high computational complexity due to iterative scheme for energy minimizations and the complex constraints. Therefore, it is not appropriate to apply the method in real-time applications.

Local methods use dissimilarity between color or intensity in the neighboring window. It is implicitly assumed that all pixels in a support window are from similar depth in a scene. These methods can acquire correct results in highly textured regions, but often tend to produce noisy results in large textureless regions. Therefore, to obtain accurate results at depth discontinuities and in homogeneous regions, an appropriate window must be selected for each pixel.

Adaptive window algorithms [2][3][4] try to find optimal window for each pixels in accordance with changing the window size and shape adaptively. For this window cost, Kanade and Okutomi model [2] presented a method to select an appropriate window by evaluating the local variation of intensity and disparity. These methods are, however, highly dependent on the initial disparity estimation and is computationally expensive. Other popular methods are using the multiple windows [5][6][7]. For each pixel, a small number of different windows are evaluated, and the one with the best cost is retained. Generally, window size is constant, but shape is varied. This method has still some problems in homogeneous regions. A compact window algorithm [8] is one of the famous methods. Window cost is the average window error plus bias to larger windows. However, it is not efficient enough for real time implementation. Adaptive support weight approach [10] computes the matching cost based on the photometric and geometric relationship with the pixel under consideration. However, it is not efficient enough for real time approach due to the high computational complexity,either.

We propose a new adaptive window algorithm. The method estimates window sizes and shapes by using boundary information adequately. In order to estimate boundary and disparity information simultaneously, we use the hierarchical framework. The rest of the paper is organized as follows. First, we present the outline of proposed algorithm in section 2. Then, we explain how to obtain a boundary map, decide the direction of edges and post processing. Experimental results on various data sets are shown in section 4, and conclusions are drawn in section 5.

2 Outline of Proposed Algorithm

Generally, the performance of local methods depends on how well the window is selected adaptively at each pixel. The window must satisfy the following conditions [2].

a. A window should be large enough to have sufficient intensity variation.
b. A window should be small enough to contain pixels at approximately equal disparity only.

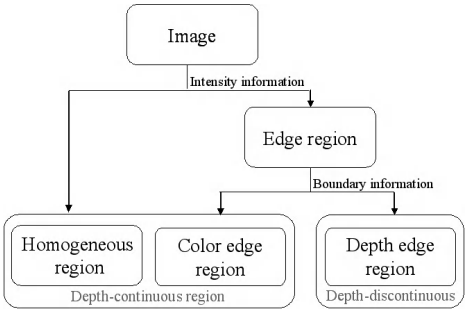


Fig. 1. Classification of Textureless and boundary region

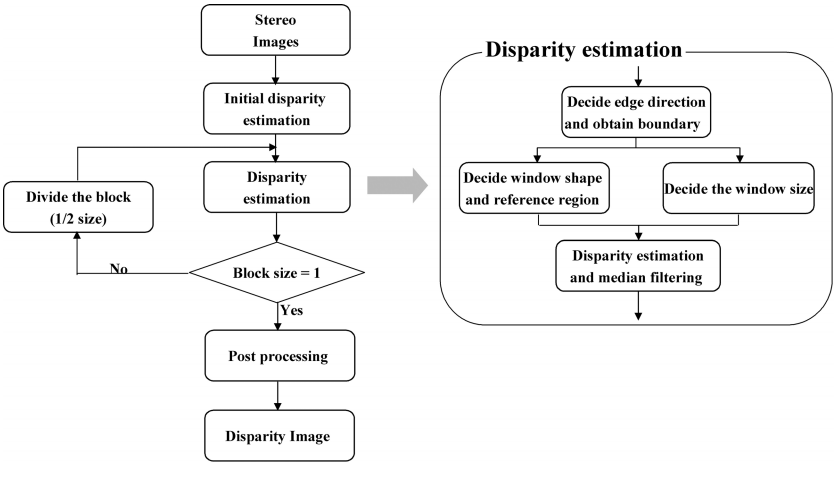


Fig. 2. The flow chart of proposed algorithm

As window size is increased from small to large, the results range from accurate disparity boundaries but noisy in low textured areas, to more reliable in low textured areas but blurred disparity boundaries. It is very difficult to obtain the optimal window for each pixel, which are reliable in both low textured areas and object boundaries. We classify an image into the depth-discontinuous and continuous region, and different approaches are used in each region as shown in Fig. 1.

To solve the matching problem in homogeneous regions, a window size increases until the window includes sufficient texture information or encounters the boundary map. Fig. 2 shows the flow chart of the proposed algorithm. The key technique in the process is the estimation of optimal window and boundary map. In boundary regions, the window is defined as 9 different shape models. Fig. 3 shows the 9 window models for each direction. Each window is decided according to the direction of the corresponding edge. The proposed algorithm is different from multiple window method [5]: Multiple window method changes the



Fig. 3. Definition of 9 window models

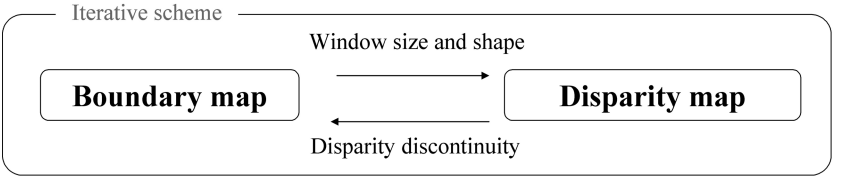


Fig. 4. Relation of disparity and boundary information

location of pixel and select the best position in the window. On the other hand, the proposed window method chooses one among the 9 window models by using direction of an edge. Moreover, unlike the multiple window using fixed window shape, window shape is different according to the edge direction. It makes the algorithm robust to object boundary regions.

The boundary information is necessary for optimal window selection. According to the boundary information, an image can be divided into the homogeneous and object boundary regions, direction of each pixel is calculated at boundary regions. Generally, the boundary regions are determined by the disparity information, which should be estimated, as shown in Fig. 4. Thus, to estimate the boundary and disparity information simultaneously, we use a hierarchical framework.

Fig. 5 shows the shape and size of window in 'tsukuba' image. We can confirm that an optimal window is estimated at each pixel adequately. The sum of absolute difference (SAD) is used as the matching cost and the disparity is finally selected by Winner-Takes-All(WTA) method in Eq (1).

$$E(x,y,d) = \frac{1}{||N(x,y)||} \sum_{(x',y')\in N(x,y)} |I_l(x',y') - I_r(x'+d,y')|$$
$$\hat{d}(x,y) = \arg\min_{d} E(x,y,d), \tag{1}$$

where I_l and I_r are input images and $N(x,y)$ represent an optimal window for each pixel. Finally, we obtain final disparity map through the proposed post-processing schemes.

3 Implementation

3.1 Improved Hierarchical Framework

In order to obtain boundary and disparity map simultaneously, we use hierarchical framework. The basic idea of a hierarchical approach is to start at the

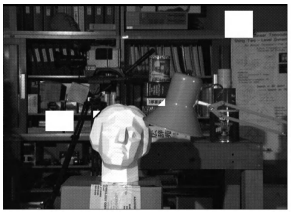


Fig. 5. Adaptive window : adaptive window in Tsukuba image

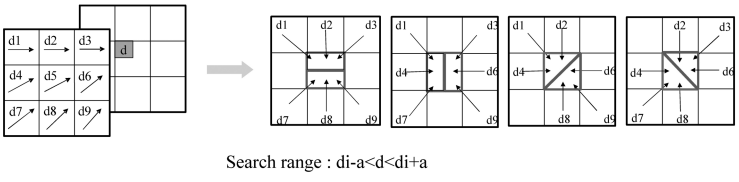


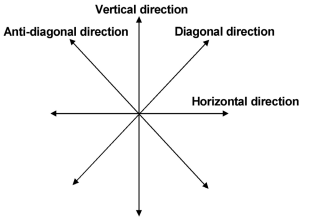
Fig. 6. Reference regions in hierarchical framework

highest level with a large measurement window and a maximum search range. Only a set of candidate displacement vectors from the higher level is considered at a lower level in order to decrease the potential displacements to be taken into account. The dimension of the measurement window is decreased as well. In order to keep implementation costs as low as possible, displacement vectors are only estimated with respect to some selected sampling positions at the highest level. However, in contrast with the conventional approach, we decide candidate vectors selectively according to the direction of each block and search the image to obtain optimal disparity near the candidate disparity value, as shown in Fig. 6. It is possible to enhance the accuracy of disparity map and reduce the computational complexity in this way.

3.2 Boundary Region Detection

It is possible for an image to classify edge region and homogeneous region using intensity information. However, edge region includes both color edge and depth edge. Since color edge regions are not object boundaries, we divide them according to disparity information as shown in Fig. 1. We use different methods at each class for accurate stereo matching. To estimate the direction of edges, we use the texture information for each block. We define four types of edge directions (horizontal, vertical, diagonal and anti-diagonal) as shown in Fig. 7 (a). To determine the type of block, we utilize four 'Sobel' masks, as shown in Fig. 7 (b). For each region, four types of 'Sobel'masks are applied and the directional type with the largest summation of absolute 'Sobel'value is selected. After four 'Sobel'operations, we sum the absolute 'Sobel'values of all the directional types in Eq. (2). We use this information to decide the location of the homogeneous region[13].

$$Sum = Sum_h + Sum_v + Sum_d + Sum_{ad}, \tag{2}$$



(a)

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}$$

(b)

Fig. 7. Definition of direction and four directional Sobel mask

where Sum_h, Sum_v , Sum_d and Sum_{ad} represent the sums of absolute 'Sobel' values in each direction, respectively.

The region of block is considered as a homogeneous region when the Sum is less than the threshold (E_{th}). In order to consider the characteristics of an image, the value of E_{th} is computed as follows:

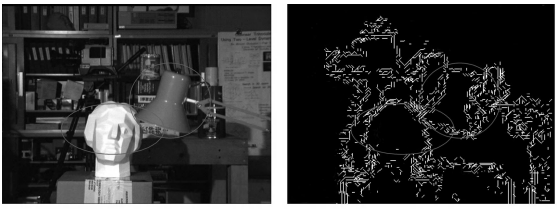
$$E_{th} = \frac{1}{N_x N_y} \sum_{(i,j) \in I_r} ((I_h(i,j) + I_v(i,j) + I_d(i,j) + I_{ad}(i,j)) \times c \ , \ c \in [0,1], \tag{3}$$

where $I_h(i,j)$, $I_v(i,j)$, $I_d(i,j)$ and $I_{ad}(i,j)$ are the output values for the each directional 'Sobel' masks. N_x, N_y indicate the image size, respectively. By adjusting the constant value c, we can control the proportion of the homogeneous region. Fig. 8 shows the results of boundary detection. The proper directions are correctly estimated especially in 'head' and 'lamp'.

The boundary information is updated in hierarchical framework. According to the information of edge direction and boundary, adaptive window size/ shape and the reference region are computed to execute accurate and fast disparity estimation. In boundary regions, to reduce 'overfitting' problem in the object boundary, we use adaptive shape windows. In homogeneous region, the different size of window is defined to overcome the problems in the textureless regions.

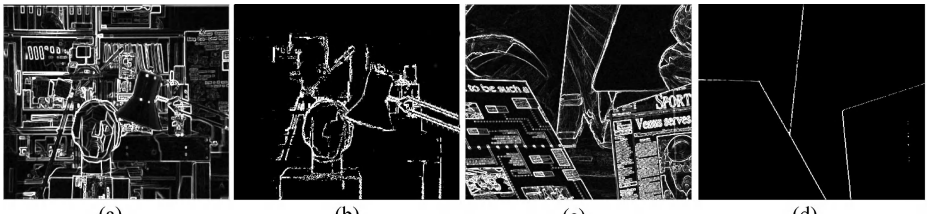
$$\begin{aligned} &the\ pixel\ is\ boundary\ if\ \ \frac{Max(diff_1, diff_2)}{Min(diff_1, diff_2)} > B_{th} \\ &diff_1 = |d' - d|, diff_2 = |d - d'| \ \{^{horizontal: d'(i,j) = d(i, j \pm k),}_{vertical: d'(i,j) = d(i \pm k, j)} \end{aligned} \tag{4}$$

where $diff_1$ and $diff_2$ are the difference between disparity of neighboring pixels and k is a window size in each level. The region of block is decided as object boundary when the rate of difference is more than the threshold (B_{th}). The process is repeated in each level, and we can obtain the boundary of a reference image as shown in Fig. 9.



(a) (b)

Fig. 8. 4x4 block directions at boundary regions: (a) Tsukuba reference image (b) Four direction



(a) (b) (c) (d)

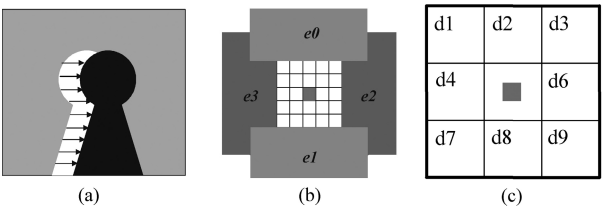
Fig. 9. Edge map and boudnary map: (a)(b) Tsukuba image, (c)(d) Venus image

3.3 Post Processing

According to the selection of adequate window in each region, problems of local method are reduced considerably. Nevertheless, there are still some problems in occluded and textureless regions such as 'foreground-fatting' problem in boundary region. In this section, we propose a post-processing scheme for improving the performance.

Occlusion handling

The occluded region exist in depth discontinuous regions nearby an object and disparity of this region is filled with the false values. Therefore, we should assign exact disparity values in occlusion region using boundary map. Disparity discontinuity exists at object boundary and occlusion regions are the left side of the object. Thus, disparities of the occluded regions is filled with the value of left side disparity at occluded regions as shown in Fig. 10 (a).



(a) (b) (c)

Fig. 10. Post processing : (a) Occlusion handling (b) Definition of boundary regions in a block (c) Definition of reference region

Disparity regularization
Although we use adaptive window, some outliers still exist in homogeneous regions. Thus, we propose a filter that calculate the optimal disparity value in the neighborhood disparities at the outlier while preserving the disparity discontinuity at object boundary. The proposed filter decides the outlier with mean (win_m) and variance (win_v) inside fixed window by equalization as shown in Eq(5), and correct value is computed as shown in Eq(6).

$$win_m + win_{\mathrm{v}} < d \quad or \quad win_m - win_{\mathrm{v}} > d \tag{5}$$

$$d_c = \arg\min_{d_k}\{(sum_e + sum_{ei}) \times |m_d - m_i|\} \quad i = e_0, e_1, e_2, e_3, \tag{6}$$

where m_d is mean of disparities in a window and m_is' are means of disparities (0,1,2 and 3 indicate top, bottom, left and right direction, respectively) as shown in Fig. 10(b). sum_{ei} denote the amount of edges at same place in boundary region of window and sum_e represent the amount of edge in a window. d_ks' are the candidate disparity vectors in neighborhood regions. d is exchanged by optical disparity, d_c. In boundary regions, due to disparity discontinuity, we use intensity values for the discrimination of best disparity at boundary, instead of window mean and variance. According to the boundary direction, we decide candidate disparities and check the optimal disparity value using Eq (7)

$$d_c = \arg\min_{d_k}(|I_l(i,j) - I_r(i,j+d_k)|), \tag{7}$$

where d_ks' are candidates of disparities in referred regions as shown in Fig (10)(c) ([horizontal:d_2,d_8],[vertical:d_4,d_6],[diagonal:d_1,d_9],[anti-diagonal:d_3,d_7]) .

4 Experimental Results

To evaluate the performance of our approach, we used a test bed proposed by Scharstein and Szeliski [1]. We evaluated the proposed algorithm on these test data sets, Tsukuba, Venus, Sawtooth and Map. For all the experiments, we set c=0.5, B_{th}=10, and minimum and maximum window sizes to 4 by 4 and 32 by 32 squares.

The performance of the proposed algorithm is measured by the percentages of bad matching (where the absolute disparity error is greater than 1 pixel). This measure is caculated in three different parts of an input image including the entire image(all), untextured (untex), and discontinuite (disc) regions.

Fig. 11 shows the results of stereo matching for the 4 standard stereo images. For comparison, we include the results of window based methods. The results show that the proposed algorithm achieves good performance in conventionally challenging areas such as object boundaries and untextured regions. Especially, the results of ambiguous regions and the preserving of disparity discontinuity are better than the conventional methods. This is due to the efficient selection of adaptive window and post-processing according to boundary map. The running times for the Tsukuba, Sawtooth, Venus, and Map scenes are 1, 2, 2, 1 seconds,

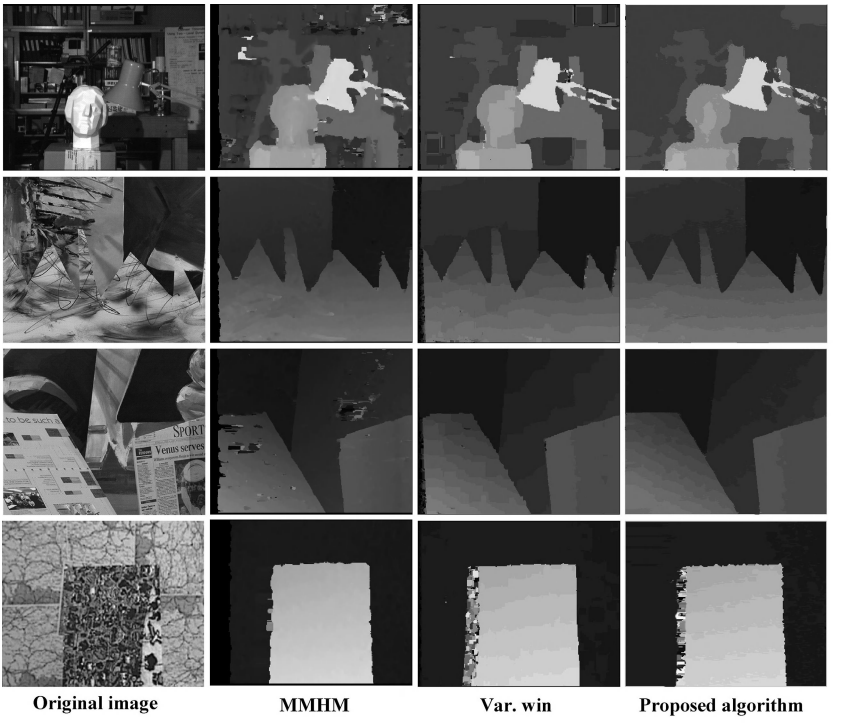


Fig. 11. Results for (from top to bottom) Tsukuba, Sawtooth, Venus, and Map image pairs

Table 1. Comparative performance of algorithms

	Tsukuba			Sawtooth			Venus			Map	
	all	untex	disc	all	untex	disc	all	untex	disc	all	disc
Proposed method	2.25	1.58	12.19	0.77	0.23	6.76	0.86	0.28	6.49	0.40	5.18
Adapt. weights[10]	1.51	0.65	7.24	1.14	0.27	5.48	1.14	0.61	4.49	1.47	13.58
Var. win[4]	2.35	1.65	12.17	1.28	0.23	7.09	1.23	1.16	13.35	0.24	2.98
Graph cut[11]	1.94	1.09	9.49	1.30	0.06	6.34	1.79	2.61	6.91	0.31	3.88
Tree DP[14]	1.77	0.38	9.48	1.44	0.84	6.87	1.21	1.41	5.04	1.45	13.00
Comp. win[8]	3.36	3.54	12.91	1.61	0.45	7.87	1.67	2.18	13.24	0.33	3.94
MMHM[15]	9.76	13.85	24.39	4.76	1.87	22.49	6.48	10.36	31.29	8.42	12.68

respectively on Pentium IV 3.0 GHz. Table 1. shows quantitative results for the stereo images using true maps. The results show that the proposed method has comparable performance with state-of-the-arts.

5 Conclusion

In this paper, we presented a new stereo matching algorithm using adaptive window and boundary map. By efficiently using the advantage of boundary

information with adaptive window, we could acquire comparable performance with state-of-the-arts.

The boundary map improves the performance of the proposed algorithm. We plan to investigate the more accurate boundary extraction method using disparity information with intensity value. We are also examining the real time computational issues related to the proposed technique.

Acknowledgement

This research was partially supported by the MIC, Korea, under the ITRC support program supervised by the IITA (IITA-2005-(C1090-0502-0027)) and was partially supported by the MOE, MOCIE and the MOLAB through the fostering project of the Lab of Excellency.

References

1. D. Scharstein and R. Szeliski: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, IJCV, Vol. 47 (2002) 7-42.
2. T. Kanade and M. Okutomi: A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiments, PAMI, vol. 16, no. 9 (1994) 920-932.
3. Y. Boykov., O. Veksler, and R. Zabih: A Variable Window Approach to Early Vision, PAMI, vol. 20, no. 12, (1998) 1283-1294.
4. O. Veksler.: Fast Variable Window for Stereo Correspondence using Integral Images, CVPR, vol. 1 (2003) 556-561.
5. A. Fusiello, V. Roberto and E. Trucco.: Efficient Stereo with Multiple Windowing, CVPR, (1997) 858-863.
6. A.F. Bobick and S.S. Intille.: Large Occlusion Stereo, IJCV, vol. 33, no. 3 (1999) 181-200.
7. S. B. Kang, R. Szeliski, and C. Jinxjang.: Handling Occlusions in Dense Multi-View Stereo, CVPR, vol. 1 (2001) 103-110.
8. O. Veksler.: Stereo matching by compact windows via minimum ratio cycle, ICCV01, (2001) 540-547.
9. H. Kim, Y. Choe and K. Sohn.:Disparity estimation using region-dividing technique with energy-based regularization, Optical Engineering, vol. 43, no. 8, pp 1882-1890, Aug. 2004.
10. K.-J. Yoon and I.-S. Kweon.: Locally Adaptive Support- Weight Approach for Visual Correspondence Search, CVPR, Vol.2 (2005) 924-931.
11. Boykov.Y., Veksler.O., Zabih.R: Fast approximate energy minimization via graph cut, PAMI23, (2001) 1222-1239.
12. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts, ICCV01, (2001) 508-515.
13. Y. Kim, J. H. Lee, C. Park and K. Sohn.:MPEG-4 compatible stereoscopic sequence CODEC for stereo broadcasting, IEEE Trans. on Consumer Electronics, vol. 51, no. 4, pp. 1227-1236 Nov. 2005.
14. O. Veksler.: Stereo correspondence by dynamic programming on a tree, CVPR, Vol. 2 (2005) 20-25.
15. K. Muhlmann, D. Maier, J. Hesser, and R. Manner.: Calculating dense disparity maps from color stereo images, an efficient implementation, SMBV, (2001) 30-36.

A New Photographing Apparatus for Skin Maps of Human Face Rendering

Haedong Kim, Howook Jang, and Inho Lee

Digital Contents Division, ETRI, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea

Abstract. Movie contents makers generally require high quality texture maps built of images to render a realistic face skin. However, it is difficult to capture the images of same facial expression from various directions because the expression can be changed easily. A new data capture apparatus using polarization properties of light is made to capture mapping images for human face. It captures images without specular reflection in a moment, and the images are used to make texture maps, bump maps and special characteristic maps. The maps are used to render a realistic human.

1 Introduction

A lot of movies have been made of computer graphical effects and those effects make digital actors come out more frequently in the movies. Like the movie, 2001's Final Fantasy: The Spirits Within, the role of a digital actor is similar to that of a human actor, such as human-like shape and human-like action. When rendering an actor, it is difficult to render the face skin of the actor because face is a main characteristic of him and the color of face skin is changed according to illumination condition.

There are various skin modeling and rendering techniques as follows. First, analytical skin reflectance models are proposed. Diffusion approximation is used, and intuitive model variations are allowed to map directly to variations in color [1], [2], [3]. Their methods are suitable for interactive modeling and rendering, and they have possibility to extend to numerical model when combined with proper scattering model to produce BRDF or BSSRDF.

Second, numerical skin reflectance models solve the full light transport in skin tissue, and so they can make high quality images [4], [5]. These models are not suitable for interactive applications, and the quality of their results depends directly on the skin anatomy model, respectively.

Third, image-based rendering technique captures effects such as self-shadowing, inter-reflections, subsurface scattering, etc., and so it is suitable for interactive applications [6]. Incident illumination of the technique is captured in the images. It is difficult to modify the results and high memory requirements.

Various skin modeling and rendering techniques have been proposed, but well-captured face image and well-fit modeling data need to make a realistic digital human face. Data capture of human face is, however, difficult because the captured data for facial expression can be changed during long capturing time. Generally movie contents makers require high quality texture map source data to render a realistic face

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 326 – 333, 2006.
© Springer-Verlag Berlin Heidelberg 2006

skin. Human face consists of several layers, and the face color is changed by the result of interacting lights with each layer, that is facial expression is changed as time. If the time of capturing images is longer, the environments captured each image are very different and the big difference among the images shows in color, facial expression. Because a face texture image is made by several images that are captured by various directions, the case of making a texture with captured data in long time is that we could lose a lot of information of source image, such as color, facial expression, and so on.

A method for the separation of reflection components using polarization is proposed, but it sometimes misses the specular reflection [7]. Tan is suggested good idea for one image, but 3D face generally requires several images [8], [9].

So we made a new photographing apparatus using polarization of light to get three images simultaneously. It captures images of a human face without specular reflection in very short time for providing human mapping source data. And the data are used to make texture maps, bump maps and special characteristic maps. They are used for rendering a human face with advantages of skin models, mentioned above.

2 A New Photographing Apparatus

A new photographing apparatus is introduced in this section. It provides mapping images for texture mapping and captures various optical characteristic data of a face in a moment.

2.1 System Overview

Our system consists of two parts: an illumination part and a capture part. The illumination part has light source, a polarizing film (or polarizer), and a control device. To adapt captured images to application environments, incandescent lamps, fluorescent lamps or halogen lamps are used as the light source. The polarizer uses to polarize beams of the light source. The control device changes beams of the light source to radiate either directly or through the polarizer. The capture part has a camera, a polarizer and a control device. The camera is used to capture a human face, and the polarizer is also used to pass light rays of same polarized direction. The control device in the capture part changes the direction of the polarizer, as well as plays a similar role mentioned above.

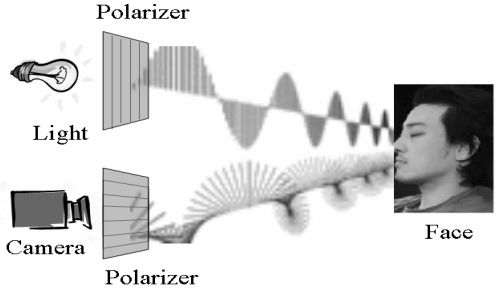
Fig. 1 illustrates our system. Fig. 1 (a) shows a full system that is operated by 3 units, and it can capture 3 images from each unit simultaneously. So, these images are captured as same facial expression in same lighting environments. Face texture maps can be made more easily by using them. Fig. 1 (b) is one unit of our system that has the illumination part and the capture part. Fig. 1 (c) presents the principle of our system, that is, polarized light from a light source is reflected by a human face and then captured by a camera with another polarizer. All camera and polarizers are simultaneously controlled by the control device.

2.2 The Properties of Polarization

We utilize the properties of polarization based on the fact that if linearly polarized light of a known orientation is reflected at oblique incidence from a surface then the



(a) Full system (b) One unit

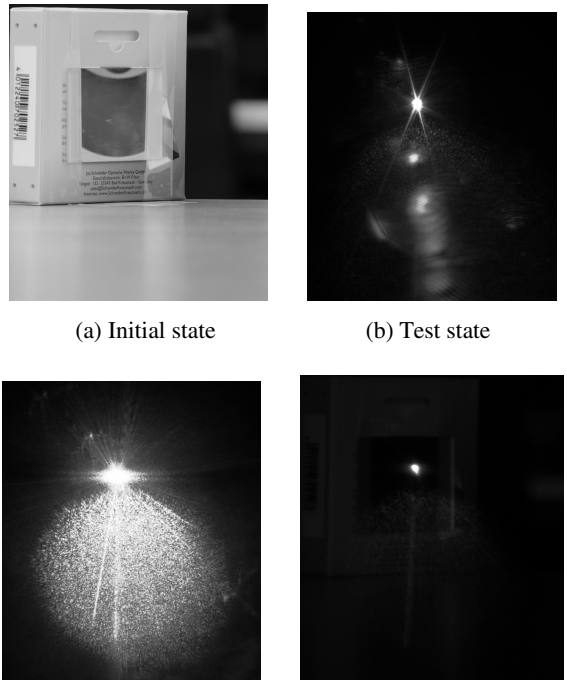


(c) Principle of our system

Fig. 1. Our System

reflected light is elliptically polarized. The shape and orientation of the ellipse depend on the angle of incidence, the direction of polarization of the incident light and the reflection properties of the surface.

Fig. 2 shows images as the results when the polarized light from a laser is reflected by a mirror and then captured by a camera with another polarizer. Fig. 2 (a) is testing environments for polarization of light. In the figure, there are a mirror centered on a box, a laser at the right of the box and a camera in front of the box. The angle between the normal vector of the box and the direction vector of camera is near 45 degree. Fig. 2 (b) illustrates a basic state, where the laser beam is radiated, reflected by the mirror and captured by the camera. But any polarizer is not used in each part. Fig. 2 (c) shows parallel test, where two polarizers are used, and the orientation of two polarizers are parallel with each other, that is, same direction. Fig. 2 (d) shows a perpendicular test, where the orientation of two polarizers is perpendicular with each other. In fig. 2, the exposure time of (b) and (c) is similar, that is, 0.8s and 0.6s each, but that of (d) is very long, that is, 30s. If we capture images in similar exposure time at the case of fig. 2 (d), we can only see a white point. It means that if we use the case of perpendicular direction of two polarizers, we can capture images without specular reflection, except for specular reflection. In cases of the fig. 2 (b) and (c), the image at the case of fig. 2 (c) is more exaggerated at specular reflection.



(a) Initial state (b) Test state

(c) Parallel test (d) Perpendicular test

Fig. 2. Polarization experiments

2.3 Use of Our System

Our system is used to capture source data for high quality texture maps. The data are also used to make bump map, and special characteristic maps. As a human is alive, his facial expression can change during image capture. It is difficult to capture many images with high quality. So, three units are used to capture images of a human face. In same lighting, three images are captured simultaneously and additional images in a moment.

Fig. 3 illustrates captured images by our system in the lighting environments that consist of incandescent lamps. The system can take 3 images of different directions simultaneously. Three Canon 1Ds Mark II cameras are used to capture these images by aperture-priority-auto mode and white-balance-auto mode. It usually takes less than 1 minute to capture three pairs of images. The images can be converted by white balance value of incandescent lamps as 2800K color temperature.

Fig. 4 presents three different images with different filters each. Fig. 4 (a) shows the image that are captured without any polarizer. The image at fig. 4 (b) is captured in perpendicular direction of two polarizers each, and the image at fig. 4 (c) is in parallel direction. As mentioned at fig. 2, the image at fig. 4 (c) is shiner than the image at fig. 4 (a). The images at fig. 4 (b) are captured without specular reflection, and so used for face texture maps. The images are captured in raw mode of the camera, and converted into other lighting conditions.



Fig. 3. Images of three different directions



(a) None (b) Perpendicular (c) Parallel

Fig. 4. Three different images with different filters



(a) Diffuse (b) Specular (c) Reddish part

Fig. 5. Processed images for texture maps

Fig. 5 shows processed images by using row images of fig. 4, respectively. The image at fig. 5 (a) is a diffuse reflection part and base data of a texture map. The image at fig. 5 (b) illustrates a part of specular reflection, and the image at fig. 5 (c) shows a reddish part of the front face. When rendering, these images may be used for additional characteristics of face skin.

3 Rendering with Maps

High quality cameras, such as Canon 350D or Canon 1Ds Mark II, are used for capturing source images. In the previous section, captured images are taken by using 3 units. In each unit, the model of camera is Canon 1Ds Mark II, and the light source consists of 4 incandescent lamps. Adobe photoshop® software is used to compose a face texture map by heuristic methods, shown at fig. 6.

Fig. 7 shows a rendered image by using the texture map of fig. 6 and a mental ray renderer for Maya® modeling software. The image shows the detail structure of face skin such as pores without specular reflection. Using a texture map data without specular reflection, new images can be made in new lighting environments.

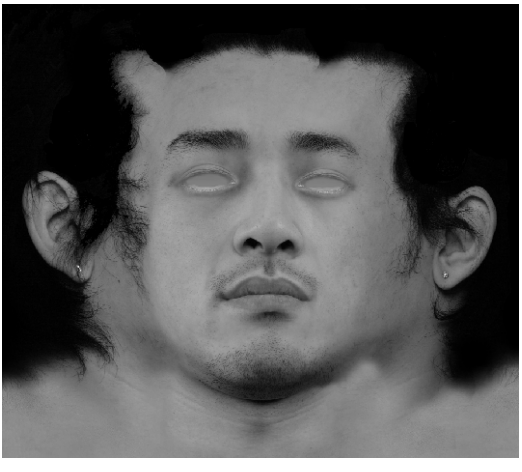


Fig. 6. A face texture map



Fig. 7. A rendered image by a mental ray renderer

4 Future Work

A new apparatus using polarization of the light for data-capture is introduced. It can capture images of a human face without specular reflection, that is, images with diffuse reflection in a moment. And it can also be able to capture optical characteristic data of a human face. The images and the characteristic data are used to make various maps and will be applied to render a digital human face in a movie.

The apparatus is designed to get high quality images without specular reflection for realistic rendering of a human face skin, and it is possible to extract specular reflection part and reddish part from the images. However, ellipsometry using polarization can give us more other properties of the skin, such as its wounded part. So, we need to research into ellipsometry which is a non-destructive optical technique for determining film thickness and optical properties.

The captured data are used to make a texture map, bump map and special characteristic maps. The maps have been made by heuristic methods. If they are based on bio-optical characteristics and rendered by using them, the results will be more realistic. And we need to study spectral rendering for realistic human skin rendering in real light.

Acknowledgements

The Ministry of Information and Communication (MIC) in KOREA has provided the support for this work. Professor Myoungjun Kim and Daehyun Kim at Ewha university help us make the system.

References

1. Stam, J.: An illumination model for a skin layer bounded by rough surfaces. Proceedings of the 12th Eurographics Workshop on Rendering, Springer-Verlag, London, UK (2001) 39-52
2. Jensen, H.W., Marschner, S.R., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. In Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques, ACM Press, New York (2001) 511-518
3. Jensen, H.W., Buhler, J.: A rapid hierarchical rendering technique for translucent materials. In Proceedings of SIGGRAPH'2002, ACM Press, New York (2002) 576-581
4. Hanrahan, P., Kreuger, W.: Reflection from layered surfaces due to subsurface scattering. In Proceedings of SIGGRAPH'1993, ACM Press, New York (1993) 165-174
5. Ng, C.S.-L., Li, L.: A multi-layered reflection model of natural human skin. In Proceedings of Computer Graphics International 2001, IEEE Computer Society, Washington DC (2001) 249-256
6. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In Proceedings of SIGGRAPH'2000, ACM Press New York (2000) 145-156
7. Nayar, S.K., Fang, X.-S., Boult, T.: Separation of Reflection Components Using Color and Polarization. International Journal of Computer Vision, Vol. 21 no. 3, Kluwer Academic Publishers, Hingham (1996) 163-186

8. Tan, R. T., Nishino, K., Ikeuchi, K.: Separating Reflection Components based on Chromaticity and Noise Analysis. IEEE Transaction on Pattern analysis and Machine Intelligence, Vol. 26, No. 10, IEEE Computer Society, Washington DC (2004) 1373-1379
9. Tan, R. T. , Ikeuchi, K.: Separating Reflection Components of Textured Surfaces Using a Single Image. IEEE Transaction on Pattern analysis and Machine Intelligence Vol. 27, No.2, IEEE Computer Society, Washington DC (2005) 178-193

Investigating the Dynamics of Facial Expression

Jane Reilly, John Ghent, and John McDonald

National University of Ireland, Maynooth

Abstract. This paper is concerned with capturing the dynamics of facial expression. The dynamics of facial expression can be described as the intensity and timing of a facial expression and its formation. To achieve this we developed a technique that can accurately classify and differentiate between subtle and similar expressions, involving the lower face. This is achieved by using *Local Linear Embedding* (LLE) to reduce the dimensionality of the dataset and applying *Support Vector Machines* (SVMs) to classify expressions. We then extended this technique to estimate the dynamics of facial expression formation in terms of intensity and timing.

1 Introduction

Darwin, in 1872, was the first to recognize the importance of facial expressions and the role that they play in human communication [1]. During the subsequent years as behavioral scientists sought a means to objectively measure facial expressions, many different techniques and methodologies for describing facial expressions were developed (see [2] for a comprehensive review). The *Facial Action Coding System* (FACS), created by Ekman and Friesen, in 1978, is the most comprehensive of these standards and is widely used in research. The FACS provides an unambiguous quantitative means of describing all movements of the face in terms of 46 *Action Units* (AUs) [3].

Recent research has shown that it is not just the expression itself, but also its dynamics that are important when attempting to decipher its meaning [4,5,6,7, 8,9,10]. The dynamics of facial expression can be defined as the intensity of the AUs coupled with the timing of their formation. Ekman et al. suggest that the dynamics of facial expression provides unique information about emotion that is not available in static images, namely the temporal information itself [11]. There is now a growing body of psychological research that argues that temporal dynamics of facial behavior - such as the timing, duration and intensity of facial activity - is a critical factor for the interpretation of the observed behavior [4]. This paper aims to not only classify expression but also capture the dynamics of facial expression formation.

In this paper we extend our previous work [12] to classify subtle changes in expression and capture the dynamics of facial expression. To achieve this we use *Locally Linear Embedding* (LLE). LLE was originally proposed in 2000 as a non-linear dimensionality reduction technique that computes low-dimensional neighborhood preserving embeddings of high-dimensional data by unfolding the underlying manifold [13]. We show that this non-linear dimensionality reduction

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 334–343, 2006.
© Springer-Verlag Berlin Heidelberg 2006

technique provides a means for classifying subtle changes in expression and allows for analysis of the dynamics of facial expression. In other words, if LLE can capture the underlying manifold of facial expression independent of identity, it would then provide information about the timing and intensity of that expression, hence capturing the dynamics of facial expression.

This paper is organized as follows. In Section 2, recent research on the importance of the dynamics of facial expressions is introduced from a behavioral science perspective. Following this we survey a number of computer vision based automated techniques for the analysis of both the appearance and dynamics of facial expression. Our proposed methodology is presented in Section 3. Experiments are presented in Section 4 . Conclusions and a discussion on the results of these experiments is provided in Section 5.

2 Related Work

In this section we first review the state-of-the-art in expression analysis from a behavioral science point of view. Subsequent to this we detail computer vision based approaches for the automated analysis of facial expression.

2.1 Behavioral Science

Since the importance of facial expressions was first established in 1872 [1], many studies have been carried out attempting to interpret their meaning. However, according to Ambadar et al. [5], only a few investigators have examined the impact of dynamics in deciphering faces. These studies were largely unsuccessful due to their reliance on intense facial expressions. While the primary reasoning behind the research carried out by Ambadar et al. was to establish the importance of dynamics in deciphering facial expressions, it also highlighted the fact that facial expressions are frequently subtle. They found that subtle expressions that were not identifiable in static presentations suddenly became apparent in dynamic display.

Schmidt et al. studied the normative characteristics and individual differences that occur in spontaneous smiles. They found that spontaneous smiles, in contrast to posed smiles, have a relatively fast onset, averaging slightly over 0.5 seconds. Approximately one third of smiles included in the study consisted of AU 12 (Lip Corner Puller)[1] on its own, while the remaining smiles had one or more co-occurring AUs appearing within one second. [14]

According to Cohn, further development within the area of Automatic Facial Action Unit Recognition will eliminate the need for manual FACS coding [15]. This would prove to be a significant step forward as currently it takes approximately 100 hours of training to become even marginally proficient with FACS, and it takes 1 hour for an experienced FACS coder to encode one minute of video footage [3].

[1] for more information on AUs see the FACS manual [11].

The results of studies carried out by Cohn et al. confirmed human perception of facial expressions is heavily influenced by dynamics [5]. A system that could automatically FACS code the dynamics of facial expression, would be an important advance in the field of facial expression analysis in that it would provide key information on the timing and intensities of the various AUs.

2.2 Computer Vision Approaches

There are many different techniques being currently applied to the problems of automatic facial expression recognition and automatic FACS coding such as: Expert Systems [16], Hidden Markov Models [7], Gabor filters [6, 17, 18, 19, 20, 21, 22, 23, 24] and Optical Flow Analysis [25]. For an overview of the current techniques see [26]. Facial expression dynamics, have also been used to assist the identification of individuals [4,9], where it was proposed that there are individual differences in how expressions are formed, it is suggested that this information could be used to augment facial recognition systems.

While facial expressions and their dynamics have received a lot of interest from various different research groups, in the past decade the techniques developed by Bartlett et al. have come to represent the state-of-the-art in the study of facial expression. In their most recent publication [24] they present a system that accurately performs automatic recognition of 20 AUs from near frontal-view face image sequences using Gabor filters and Support Vector Machines.

Despite the fact that this system has the potential to provide information about expression dynamics, it has not as yet been fully explored. However, a system that could fully analyze the dynamics of facial expression in terms of intensity and timing would represent a major advancement in the field of facial expression analysis.

3 Proposed Methodology

In this paper we aim to classify expression and extract information regarding the dynamics of expression by first projecting shapes of individuals portraying specific expressions into a lower dimensional *Locally Linear Embedding* (LLE) space in order to capture an underlying manifold of that expression as if forms. And secondly, to classify that expression using *Support Vector Machines* (SVMs). Information about the dynamic of facial expression is then represented in the LLE space. In this section both the LLE and SVM algorithm are briefly discussed.

3.1 Locally Linear Embedding (LLE)

LLE is a non-linear dimensionality reduction technique that is capable of generating highly non-linear embeddings, and its main optimization involves a sparse eigenvalue problem [27]. The LLE algorithm is based on simple geometric intuitions. Essentially the algorithm attempts to compute a low dimensional embedding with the property that nearby points in the high dimensional space

remain nearby and similarly co-located with respect to one another in the low dimensional space.

In 2000, the LLE algorithm was introduced by Saul and Roweis as an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embeddings of high dimensional data [13, 27, 28]. Many extensions and adjustments to this core algorithm have been proposed, ranging from Robust-LLE [29, 30] to supervised [31, 32] and semi-supervised versions of LLE [33], however in this paper we are using the original algorithm as defined in [13].

3.2 Support Vector Machines (SVM)

SVMs are a type of learning algorithm based upon recent advances in statistical learning theory [34,35,36,37] and are based on a combination of techniques. One of the principal ideas behind SVMs is the *kernel trick*, where data is transformed into a high dimensional space making linear discriminant functions practical. SVMs also use the idea of large margin classifiers. Suppose we have a dataset $(x_1, y_1), ..., (x_m, y_m) \in \mathbf{X} \times \{\pm 1\}$ where $\mathbf{X}$ is some space from which the x_i have been sampled. We can construct a dual Lagrangian of the form

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{1}$$

which are subject to the constraints

$$\alpha_i \geq 0 \quad \forall i \qquad and \qquad \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{2}$$

The solution to Equation 1 is a set of α values which defines a hyperplane that is positioned in an optimal location between two classes. See [38], for more information.

4 Experiments and Results

In this section we detail two experiments. The first is designed to classify which AU or combination of AUs a person is portraying, and the second is designed to extract information regarding the intensity and timing of a classified AU and hence provide a means for analysing the dynamics of facial expression. The data used for these experiments comes from the Cohn-Kanade [39] database.

4.1 Classifying Action Units

In this experiment we classify four lower facial expressions; AU20+AU25, AU12, AU10+AU20+AU25 and AU25+AU27. The effect that these have on the mouth is illustrated in Table 1. Our training data consisted of 141 images of an individual performing 8 sequences from neutral to extreme expression. A single individual was used to develop the LLE space as this ensures that any variance

Table 1. This table illustrates the effect of portraying four different expressions. The AUs portrayed, from left to right are; AU12, AU20+AU25, AU10+AU20+AU25, AU25+AU27.



in the LLE space is due to expression change rather than subject change. It should also be noted that two expressions are extremely similar and therefore it is hypothesised that a technique that can differentiate between these expressions can accurately classify subtle changes in appearance. During training 24 landmark points are manually placed at 24 specific locations on the mouth. We align the data to remove any unnecessary variation before applying the LLE algorithm. This is achieved by centering the data and then performing *Generalised Procrustes Alignment* (GPA) [40]. This technique aligns two shapes with respect to position, rotation and scale by minimising the weighted sum of the squared distances between the corresponding landmark points. More information on this technique can be found in [38]. We then calculate a shape model by performing LLE on the aligned training data. The LLE output using the aligned data as input is shown in Figure 1. We project new unseen data into the LLE space and

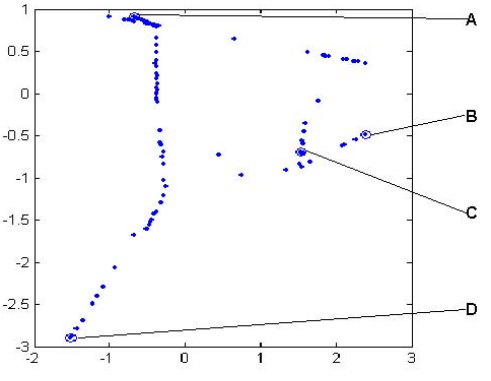


Fig. 1. Aligned LLE shape space. The Action Units portrayed, from top to bottom are; AU12, AU20+AU25, AU10+AU20+AU25, AU25+AU27.

use these outputs as inputs to the SVM classifier. As there exists four expressions to be separated, a one-against all approach yields four separate SVM classifiers. This approach requires, at most, four evaluations to acquire a result. The results from the one-against-all SVM approach in conjunction with LLE are detailed in Table 2.

Table 2. LLE SVM results. In the table above A = AU20+AU25, B = AU25+AU27, C = AU10+AU20+AU25 and D = AU12.

AU	*C*	σ	*Ts*
A-v-all	6.1	0.1	84.6
B-v-all	6	0.3	83.8
C-v-all	7	0.8	98.0
D-v-all	8	0.5	98.7
Average			91.3

In Table 2, C is the soft margin variable, σ is the kernel parameter, and Ts is the percentage of correctly classified test data. The test data consists of 10 subjects (60 images) portraying AU20+AU25, 10 subjects (60 images) portraying AU25+AU27, 3 subjects (18 images) portraying AU10+AU20+AU25 and, 3 subjects (18 images) portraying AU12. There are low, medium and high intensity examples of each expression in the training set. The soft margin variable and the kernel parameter are calculated using a cross validation approach [38].

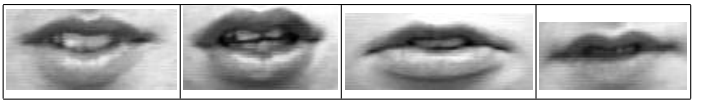
As can be seen from Table 2, this technique accurately classifies similar expressions with an average accuracy of 91.3%. This improves on our previous work as our PCA approach could only classify extreme appearance changes [38].

4.2 Examining the Intensity of Action Units

In this experiment information about the dynamics of facial expressions is extracted. The intensity over time of a composite AU lower facial expression and a single AU lower face expression are estimated according to the FACS intensity coding [11]. The FACS intensity coding ranges from A to E, with A representing a minor change in appearance, and E an extreme change in appearance. The input to these two experiments consisted of 2 datasets of mouth shapes taken from multiple subjects. These datasets consisted of multiple sequences each of a different subject. For each sequence we manually 6 frames from neutral to extreme. For AU20+AU25 a total of 10 subjects (60 frames) were landmarked, and 24 subjects (144 frames) were landmarked for the experiment on AU25.

Similar to the previous experiment, the mouth shapes were first aligned using GPA. As LLE has a well documented tendency to cluster data into identity if the data contains more than one identity, and our datasets contain multiple identities, a further preprocessing step was required. For this we use a technique called *Shape Differencing*, whereby the neutral mouth shapes of each subject was subtracted from the sample set for that subject. As we are only interested in the difference between these two shapes and not the actual shapes themselves, by using shape differencing, the variances occurring due to the identity will be removed. This is a valid step as to analyse the dynamics of facial expression it is necessary to have a sequence of images containing a neutral image.

Table 3. AU25 classified mouths



The images that were labeled as extreme in our datasets varied between each sequence, for example certain sequences only showed AU25 at an intensity of C. Table 3, shows a sample of images that were FACS coded AU25E. To deal with this variation, prior to running our experiments we manually FACS intensity coded the entire datasets.

LLE was then applied to the datasets, reducing the dimensionality to one dimension. We take the projections of the samples for intensities B, C, and D in the one dimensional LLE space and fit Gaussians to this data. Due to lack of samples for intensities A and E we just the calculate the mean. The resulting Gaussians and means are shown in Figure 2 and Figure 3. As can be seen from these figures our technique accurately characterises the intensity of the facial expressions over time and hence provides a means for modelling their dynamics.

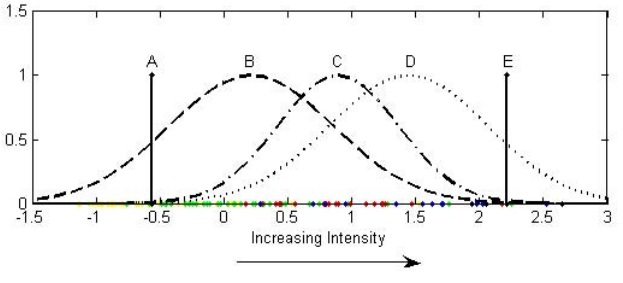


Fig. 2. Intensity over time for AU25 results

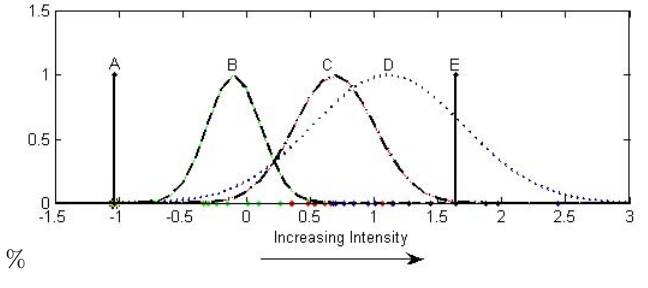


Fig. 3. Intensity over time for AU20 + AU25 results

5 Conclusion and Discussion

The accurate classification of facial expressions is a growing problem within several domains. The solution described in this paper takes a multidisciplinary approach drawing together psychological tools, statistical models and machine learning techniques. We first build a shape model that was based on an anatomical analysis of facial expression - FACS. The FACS provided us with a universal method of analyzing facial expression and allowed for the classification of facial expressions independent of identity.

In our first experiment, a shape model was calculated by using LLE to lower the dimensionality of the problem. One-against all SVMs were used to classify four lower face expressions (AU20+AU25, AU25+AU27, AU10+AU20+AU25 and AU12) resulting in an accuracy of 91.3% percent. While our previous work produced an accuracy of up to 99% percent with extreme expressions, it was unsuccessful at classifying subtle changes in expression [38] .

In our second experiment, multiple identities were used to build up intensity models for two lower face expressions. This intensity model demonstrates the changes in intensity over time, and can automatically FACS intensity code AUs.

The approach presented in this paper is an improvement on our previous work as it allows us to classify subtle changes in expression. We have shown that a low-dimensional non-linear representation of expression improves the accuracy of our classification results, enabling the classification of expression as it forms, and also providing an accurate means of classifying the intensity of AUs. This intensity information, in conjunction with timing information, provides the necessary basis for the automated analysis of facial expression dynamics.

This paper show that LLE provides an effective means of both classifying subtle changes in facial expression and estimating the intensity of facial. Future work will entail applying this technique to more comprehensive data sets incorporating a larger variety of AUs and individuals.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

References

1. Darwin, C., Ekman", P.: The expression of the emotions in man and animal. Chicago: The University of Chicago Press (1998) 1st edition in 1872, 2nd edition in 1889, 3rd edition with additional commentry by Paul Ekman in 1998.
2. Cohn, J., Kanade, T., Moriyama, T., Ambadar, Z., Xiao, J., Gao, J., Imamura, H.: A comparative study of alternative facs coding algorithms. Technical Report CMU-RI-TR-02-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2001)
3. Ekman, P., Friesen, W., Hager, J.: Facial action coding system. Consulting Psychologists Press (1978)

4. Cohn, J.F., Schmidt, K., Gross, R., Ekman, P.: Individual differences in facial expression: Stability over time, relation to self-reported emotion, and ability to inform person identification. Proceedings of Intel. Conf. On Multimedia and Expo, 2001. (2002)
5. Ambadar, Z., Schooler, J., Cohn, J.: Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions. Psychological Science (2005)
6. Bartlett, M.S., Movellan, J., Littlewort, G., Braathen, B., Frank, M.G., Sejnowski, T.J.: Towards automatic recognition of spontaneous facial actions. In Paul Ekman, editor, what the face reveals (2003) Oxford University Press.
7. I. Cohen, N. Sebe, L.C.A.G., Huang, T.S.: Facial expression recognition from video sequences: Temporal and static modeling. Computer Vision and Image Understanding, 91(1-2) (2003)
8. Ekman, P.: Facial expression of emotion: New findings, new questions. (1992)
9. Hadid, A., Pietikdinen, M.: An experimental investigation about the integration of facial dynamics in video-based face recognition. ELCVIA **5** (2005) 1–13
10. Littlewort, G., Bartlett, M.S., Fasel, I., Susskind, J., Movellan, J.: Dynamics of facial expression extracted automatically from video. Computer vision and Image understanding (2006)
11. Ekman, P., Friesen, W., Hager, J.: Facial Action Coding System Manual. (2002)
12. Ghent, J., McDonald, J., Harper, J.: Statistical model for expression generation using the facial action coding system. Technical Report NUIM-CS-TR-2003-01, NUI Maynooth (2003)
13. Saul, L.K., Roweis, S.T.: An introduction to locally linear embedding. http://www.cs.toronto.edu/ roweis/lle/publications.html (2001)
14. Schmidt, K., Cohn, J.: Dynamics of facial expression: Normative characteristics and individual differences. In: IEEE International Conference on Multimedia and Expo (ICME2001). (2001) 728 – 731
15. Cohn, J.: Automated analysis of the configuration and timing of facial expression. (2005) Afterword of What the face reveals (2nd edition): Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS) by P. Ekman and E. Rosenberg, ed., 2005.
16. Pantic, M., Patras, I.: Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. SMC-B **36** (2006) 433–449
17. Zhang, Y., Ji, Q.: Facial expression understanding in image sequences using dynamic and active visual information fusion. ICCV (2003) 113 to 118
18. Donato, G., Bartlett, M., Hager, J., Ekman, P., Sejnowski, T.: Classifying facial actions. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(10) p. 974-989. (1999)
19. Bartlett, M.S., Littlewort, G., Frank, M., lainscsek, C., Fasel, I., Movellan, J.: Recognising facial expression: Machine learning and application to spontaneous behaviour. IEEE International conference on computer vision and pattern recognition (2005)
20. Bartlett, M.S., Littlewort, G., Lainscsek, C., Fasel, I., Movellan, J.: Machine learning methods for fully automatic recognition of facial expressions and facial actions. IEEE International conference on systems, man and cybernetics (2004) 592–597
21. M. Stewart Bartlett, B. Braathen, G.L.T.J.S., Movellan, J.: Automatic analysis of spontaneous facial behavior: A final report. (2001) MPLAB-TR-2001-06, Institute for Neural Computation, University of California, San Diego.

22. M.S., Bartlett, G.L.I.F.J.M.: Real-time face detection and facial expression recognition: Development and applications to human computer interaction. (2003)
23. Bartlett, M.S., L.G.B.B.S.T..M.J.: A prototype for automatic recognition of spontaneous facial actions. (2003)
24. Bartlet, M., Littlewort, G., Lainscsek, C., Fasel, I., Frank, M., Movellan, J.: Fully automatic facial action recognition in spontaneous behavior bartlett. (2006)
25. Goneid, A., el Kaliouby, R.: Facial feature analysis of spontaneous facial expression. In Proceedings of the 10th International AI Applications Conference (2002)
26. Fasel, B., Luettin, J.: Automatic facial expression analysis: A survey. Pattern Recognition, Vol. 36(1) (2003) 259–275
27. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research **4** (2003)
28. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by local linear embedding. Science **290** (2000) 2323–2326
29. Chang, Y., Yeung, D.Y.: Robust local linear embedding. Technical report, Department of Computer Science, Hong Kong University of Science and Technology, HKUST-CS05-12 (2005)
30. Hadid, A., PietikAanien, M.: Efficient locally linear embeddings of imperfect manifolds. Proceedings of the Third International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany (2003) 188–201
31. Ridder, D.D., Kouropteva, O., Okun, O., PietikAanien, M., Duin, R.P.W.: Supervised locally linear embedding. ICANN 2003 (2003) 333–341
32. Kouropteva, O., Okun, O., Hadid, A., Soriano, M., Marcos, S., PietikAainen, M.: Beyond locally linear embedding algorithm. Technical report, Department of Electrical and Information Engerring, University of Oulu, Oulu, Finland, MVG-01-2002 (2002)
33. Ridder, D.D., Duin, R.P.W.: Locally linear embedding for classification. Technical report, Pattern Recognition Group, Department of Imaging Science and Technology, Delft University of Technology, Delft, The Netherlands, PH-2002-01 (2002)
34. Campbell, C.: kernel methods: A survey of current techniques. Neurocomputing **48** (2002) 63–84
35. Rogers, S., Williams, R.D., Campbell, C.: Class prediction with Microarray Datasets. In: BioInformatics with computational intelligence paradigms. Springer-Verlag (2004)
36. Rogers, S.: Machine Learning Techniques for Microarray Analysis. Faculty of engineering mathematics, University of Bristol (2004)
37. ScholKopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
38. Ghent, J.: A Computational Model of Facial Expression. PhD thesis, National University of Ireland Maynooth, Co. kildare, Ireland (2005)
39. Cohn, J., Kanade: Cohn-kanade au-coded facial expression database. Technical report, Pittsburgh University (1999)
40. Gower, J.C.: Generalised procrustes analysis. Psychometrika **40** (1975) 33–50

GLOBAL Topology Preservation in Isosurface Extraction of Volumetric Data*

Xingqiang Yang[1] and Caiming Zhang[1,2]

[1] School of Computer Science and Technology, Shandong University, Jingshi Road #73 Jinan, Shandong 250061, China
[2] School of Computer Science and Technology, Shandong economic University, Erhuandong Road #7366 Jinan, Shandong 250014, China
yxq@sdu.edu.cn, czhang@sdu.edu.cn

Abstract. We study the topology inconsistency between the interested object boundary and the isosurface extracted with Marching Cubes, and present a new method preserving the topology of object boundary. Different from the MC-like algorithms, the new method is not restricted to the Marching Cubes' partition on the grid vertexes: inner or outer vertexes. We find that a pixel curved out by Marching Cubes algorithm maybe include a part of the interested object and it is even a key part for preserving the boundary topology. So the boundary pixel is defined as a pixel that includes the object boundary and it's pixel value is between those of the inner and outer pixel. This paper shows how to find the boundary pixels and the boundary position in it. Thus we can find the object boundary in a pixel whose value is out of the isovalue, which is regarded as background by Marching Cubes algorithm. Experiments based on CT data also verify the effect of the new method.

Keywords: topology; isosurface; volumetric data; marching cubes.

1 Introduction

A lot of visualization of scalar fields can come down to the extraction and rendering of isosurface, such as human organ reconstruction from tomography. Isosurface extraction attracts much attention widely and many important research productions have been made. Marching cubes algorithm[1] plays an important role in isosurface extraction and is used most widely due to simpleness and efficiency.

There are topology ambiguities for MC algorithm. A lot of research papers have been published to preserve the topology consistency of isosurface extracted with MC algorithm. M .J. Dűrst[2] finds the ambiguity on the side face of the cube by analyzing the basic voxel model in MC algorithm. Nielson resolves the ambiguity in Marching Cubes with the asymptotic decider[3], and finds the critical points to approximate the isosurface in the cube with trilinear interpolant surfaces[4]. Natarajan

* Project supported by the National Nature Science Foundation of China (No. 60573180, No. 60533060).

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 344 – 353, 2006.
© Springer-Verlag Berlin Heidelberg 2006

also does researches on the isosurface in the cube, shows the ambiguity of isosurface in the cube, and presents an initial solution. The original Marching Cubes algorithm is unable to extract high quality triangle meshes with sharp features from the volumetric data. Papers[5][6] can reconstruct sharp features and reduce aliasing artifacts in the reconstructed model. Stander[7] and Boissonnat[8] provide guarantees on the topology of implicit surface polygonization based on critical point analysis. Their approaches assume that the implicit surface is smooth and require a priori knowledge of all the critical points. Gokul Varadhan, etc[9] presented a novel topology preserving subdivision algorithm to generate an adaptive volumetric grid. Their algorithm ensures that every grid cell satisfies two local geometric criteria: a complex cell criterion and a star-shaped criterion, and they show that these two criteria are sufficient to ensure that the surface extracted from the grid using Marching Cubes has the same genus and connectedness as that of the exact isosurface.

All these researches on the isosurface topology present various methods to reconstruct more accurate and reasonable topology grid, but they are limited to the MC's partition on the grid vertexes: inner or outer vertexes. Based on the MC's partition, the global topology of isosurface is determined. The most of improvements on the isosurface topology is limited in a cube, That is all of isopoints are determined by Marching Cubes, the MC-like algorithms try to link the isopoints in a better way, even insert some points. We don't think it is reasonable to sort the voxels into inside or outside of isosurface. Sometimes boundary can be included in the voxels out of isosurface.

The voxel value in scalar field is an expectation value of all geometric points in the voxel. Some values of points in a voxel are greater than voxel value, and some are less. It will produce an error in detail to determine whether a voxel is inside the isosurface or not only by its voxel value. If the error in detail is involved in the topology of boundary, it will result in serious topology inconsistency. For example, blood vessels may be very thin, even thinner than a voxel at some position due to pathological change, so that the value of interrelated voxel is less than the isovalue, and MC-like algorithms will get broken blood vessels. We try to find the voxels that are curved out of isosurface by MC algorithm but are pivotal to the topology of boundary surface, extract the isosurface in these voxels, and obtain a global topology consistency. Thus we can ensure the correct topology of isosurface among voxels, not only in the cube.

We study the scalar field based on CT data. Different from MC-like algorithms, This paper considers a voxel as a cube, and defines boundary voxels as a kind of voxels which are made up of interested object and background. Algorithms to find boundary voxels and the boundary in it are presented in this paper. We experimentalize on CT data to compare with the Marching Cubes, and the results show that our method can obtain a whole blood vessel, but Marching Cubes algorithm can only obtain two broken vessels.

Below we will analyze the topology inconsistency in MC-like algorithms, and present our methods, at last experiments will be shown.

2 Topology Inconsistency of MC-Like Algorithms

This paper will only analyze 2D data field for simplicity, the 3D data field can be treated in the same way. The data of this paper is from CT device.

Pixel values from image devices can be considered as expectation value of all geometric points in the voxel, which can be expressed as equation (1) where $E(x, y)$ denotes the density of the point (x, y) in pixel S. The pixel value of S can be defined as below

$$v = \iint_S E(x, y)dxdy \tag{1}$$

For example, the shadow in Fig.1 denotes the interested object, whose density is 700. The pixel values $P(i, j)$ (i=0,…4, j=0,…2) in Fig.1 are shown as equation (2). Assume the isovalue is T = 680.

$$\begin{matrix} 500 & 700 & 200 \\ 400 & 700 & 200 \\ 180 & 600 & 210 \\ 180 & 600 & 220 \\ 380 & 700 & 480 \end{matrix} \tag{2}$$

Because the interested object gets thinner, the pixel values $P(2, 1)$ and $P(3, 1)$ are less than T and are curved out of isoline according to Marching cubes. Marching Cubes can only obtain a broken interested object, just like the dashed lines in Fig.1. Such model not only brings misconception in visualization but also lead to wrong analysis in mathematics. such inconsistency is brought by sampling: the density of interested object is distributed over the whole pixel area, which makes the pixel value less than the isovalue so that the interested object is neglected by the isoline extracting algorithm.

Fig.2 shows CT image of a blood vessel. The pixel values becomes less at spot © due to the loss of blood stream caused by pathological change. The © spot may become a broken point of blood vessel model.

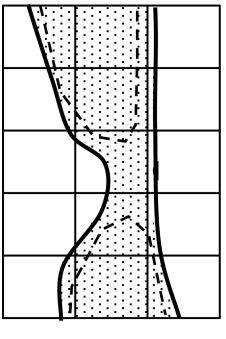


Fig. 1. Inconsistency topology

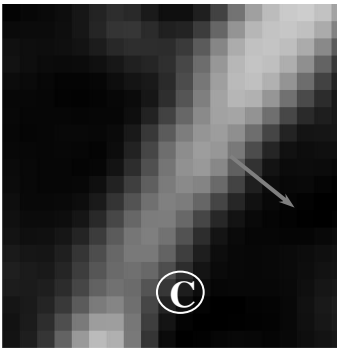


Fig. 2. CT image of a blood vessel

If we don't consider the error of image devices, pixels can be divided into three kinds: (1) the region of the pixel is full of interested object; (2) the region of the pixel is full of background; (3) the region of the pixel is composed of interested object and background. But actually the boundaries of interested object are usually fuzzy, such as what marked by the © in Fig. 2. These pixel values transit from high to low. For CT devices, this phenomena associates with the movement of human organs, precision of CT devices and reconstruction algorithms of CT images.

In a word, when isolines extracting algorithms are used, we must consider two facts: (1) the density of interested object is distributed over the whole region of the pixel; (2) Furthermore the density of interested object is even distributed across several adjacent pixels. So when human organs are small or thin, for example, one or several pixel size, the density feature of interested object disappears. It is impossible to obtain the correct model with an isolines extracting algorithm like Marching Cubes. The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

3 Reconstruction of the Geometric Model

3.1 Original Data Field and Pixel Data Field

Define 1: We refer to the accurate data in real world as **Original Data Field**, and the image data from devices as **Pixel Data Field**.

If we can have the accurate density of human body data field, then isoline algorithms will obtain accurate boundary of human organs according to different density values. However pixel data field is not accurate, its most basic unit of data is the pixel, which makes the edge of interested object unclear. This leads isoline algorithms may produce an inconsistent model. Marching Cubes algorithm regards a pixel value as a value at a vertex, and obtains an approximate original data field by interpolation. So we should note that the values at the vertexes of cube are not accurate.

Theoretically it is impossible to recover original data field from pixel data field. However we can obtain approximate original data field under some assumptions as what Marching Cubes does. We don't think it is a best way to define a value at a vertexes as a pixel value, so we try to propose other criteria by which original data field can be obtained from pixel data field.

Assumption 1: The densities at different positions of interested object is even relatively, and the density of interested object is markedly different from the density of background.

Assumption 2: The sum of the interested object density is invariable, even though the density of interested object is distributed across several pixels.

We try to make the density of interested object converge in the region of the interested object, since the edge of it become fuzzy by sampling.

Define 2: The pixels that are located at about the boundary and whose pixel values are between the density of interested object and the density of background, are called **transition pixels**.

Define 3: The pixels that includes the boundary of interested object are called **boundary pixels**.

Clearly, boundary pixels are special transition pixels. When resolution of the sampling device is higher that of the storage device (image), that is, the edge of object is thinner than a pixel and only one pixel is affected by the boundary, or the set of transition pixels only includes boundary pixels. Otherwise many pixels are affected by boundary so that there are many transition pixels along the normal vector of the boundary.

The original data field can be recovered by two steps: (1) The boundary in image is fuzzy. First it is necessary to find transition pixels, then to find the boundary pixels from the transition pixels. (2) After we have obtained the boundary pixels, or the boundary in image originally is clear. We will manage to find the position of boundary in the boundary pixels.

3.2 Find the Transition Pixels

The transition pixel has three features: (1) Its density is between the density of the interested object and that of background; (2) It lies between the object and background; (3) the pixel values of transition pixels transit gradually from the density of the interested object to that of the background. Denote the density of the object as T, the density of pixel p as $D(p)$, below is the algorithm to find transition pixels.

Algorithm 1. Find the transition pixels.

(1) Find all pixels whose values are greater than T, denoted as the set S_{in}. all of pixels that are adjacent to any pixel in S_{in} enclose the S_{in}, denoted as S_0. Let i=0.

(2) Suppose q is a element in S_i, $S_{i+1}=\{p \mid D(p)<D(q), q \in S_i, p \text{ is adjacent to } q\}$

(3) If $S_{i+1} \neq \Phi$ then
    let $i=i+1$;
    repeat step (2);
  else
    step (4).

(4) $S_t=S_1+S_2+\ldots+S_i$, S_t is the set of transition pixels.

Some restrictions can be applied to algorithm 1. For example, set the max loop number to avoid special cases. We can define the pixels enclosing the transition pixels as background pixels, denoted as S_b.

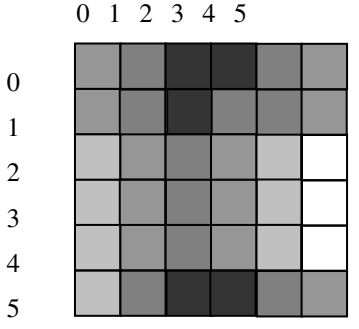


Fig. 3. Connecting pixels

Algorithm 1 finds transition pixels layer by layer from S_{in}, so it is necessary to find S_{in} first. That is, these transition pixels are adjacent to the S_{in} which is like a seed of algorithm 1. However sometimes we can't find such seed as S_{in} for some thin objects. For example, the pixels $P(2, 2)$, $P(3, 2)$ and $P(4, 2)$ in Fig.3 include respectively a part of interested object, but their pixel values are less than T because the object's density are distributed to the pixels around. $P(2, 2)$, $P(3, 2)$ and $P(4, 2)$ are transition pixels, but can't be found by algorithm 1.

So we should improve the method finding S_{in} in algorithm 1 as below:

(1) After finding S_{in} as algorithm 1, we extend the S_{in} with Region Growing Method, which relaxes the threshold T with a relaxation ΔT. Thus S_{in} will include a lot of transition pixels, such as $P(0, 1)$, $P(1, 1)$,$P(2, 2)$, $P(3, 2)$,$P(4, 2)$ and so on. The selection of ΔT depends on the estimation about the image.

(2) Shrink the S_{in} from outside to inside until meeting the condition below: reach to the pixel whose value is not less than T, or S_{in} will become unconnected.

The Pixels P_{02},P_{03},P_{12},P_{52},P_{53} in Fig.3 are in S_{in} at starting, then P_{22},P_{32},P_{42} are also included in S_{in} after two operations above. The pixels P_{22},P_{32},P_{42} are called **connecting pixels**, which connects two parts of object in image. Here the idea of Mathematical Morphology is adopted to obtain the topology of isolines.

Algorithm 1 can find the set of transition pixels, algorithm 2 will find boundary pixels from the set of transition pixels, which include the boundary of the interested object.

3.3 Determine Boundary Pixels

The transition pixels consist of three kinds of pixels, they lie respectively in region of the interested object, in region of the background, or on the boundary. We manage to classify the transition pixels into these three kinds. Assume that the background pixels are covered with the same substance, thus the expectation density of the background is $D_b=(\sum_{p\in S_b} D(p))/|S_b|$, $D_a=\sum_{p\in S_t}(D(p)-D_b)$ expresses the excessive density of transition pixels relative to the background pixels. The aim of algorithm 2 below is to gather the excessive density from the pixels near background to the pixels near the interested object. As shown in Fig.4, the excessive density is gathered near the interested object, and there are distinct jumps in the transition pixels. In Fig.4, the pixels marked with white points are boundary pixels.

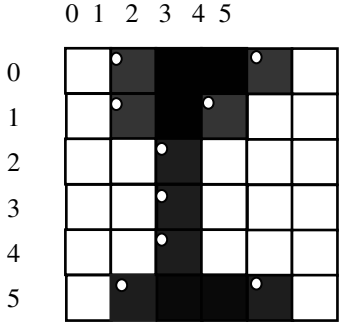


Fig. 4. Boundary Pixels

Algorithm 2. Find boundary pixels.

(1) Suppose $work = D_a$, which express the excessive density in the transition pixels, let j=1.

(2) compute $D_j=|S_j|\cdot(T-D_b)$, where T is the density of the interested object, the meaning of S_j is same as algorithm 1.

If $work > D_j$ Then

$work = work - D_j$

set the values of all pixels in S_j as T.

$j=j+1$

repeat the step (2).

Else

Step (3).

(3) The pixels in S_j exhaust all of density of *work*, but can't reach the density T. The pixels in S_j may be regarded as boundary pixels. For any pixel p in S_j, set the pixel value $D(p) = D(p)+(T-D(p))\cdot work/D_j$. The values of other pixel in S_j are set as the pixel value of background.

The pixels in S_j include the boundary of the interested object. The inside pixels to S_j overlay the interested object, The outside pixels overlay the background. Algorithm 2 classifies the transition pixels into such three kinds of pixels. Of course, if the boundary pixels in an original image are clear, then algorithm 1 and algorithm 2 are not necessary.

3.4 Find the Boundary with Sub-pixel Precision

We should manage to find the boundary in the boundary pixels although their pixel values are less than the isovalue. We can adopt the idea of sub-pixel to divide a pixel into many sub-pixels. With algorithm 2, the density can be collected together to the side of the interested object. In Fig.5, (1) is a pixel before subdivision, and (2) shows pixels after subdivision and the density is collected on the left. Then the boundary could be determined by iosline methods.

In practice the boundary can be found by approximate method below: (1) Assume the pixel value of boundary pixel is $D(D<T)$, set this pixel value as T instead of D; (2) Find boundary of the interested object with MC-like algorithm. The boundary is clearly extended outside because of increase $T–D$ of the pixel values; (3) The boundary extracted by MC-like algorithms should shrink at a ratio $(D/T)^{1/2}$. In Fig.6, the thin solid boundaries are found by MC-like algorithm, and the thick dashed boundaries are results shrunk.

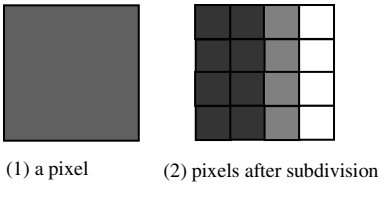


Fig. 5. Sub-pixel precision

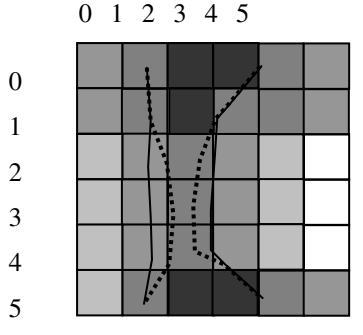


Fig. 6. Approximate method for sub-pixel

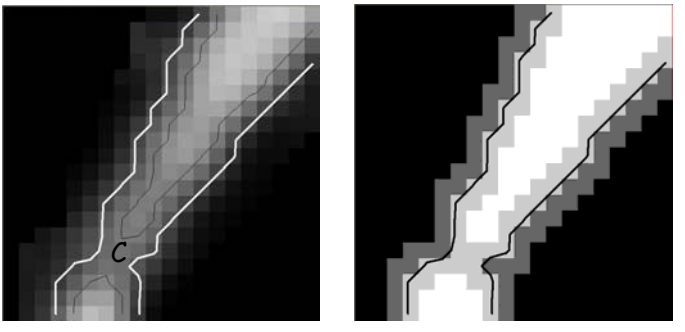
4 Experiments

The idea of this paper comes from reconstruction of blood vessels. We experimentalize on the blood vessels data from CT with new method of this paper. The results show that the new method can reconstruct the models of blood vessels and preserve the topology of blood vessels, which avoids the inconsistency of Marching Cubes.

Fig.7 (1) shows a segment of blood vessel with some pathological change at C spot, so that the pixels at C spot are dark. According to Marching Cubes, the pixels at C are curved out of the isoline. The black thin boundaries are produced by Marching

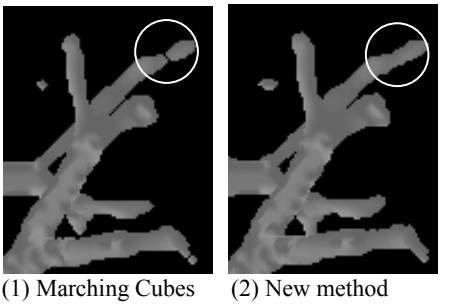
Cubes, which are inconsistent with the real blood vessel. We can obtain the consistent topology of the interested object with our new method, which is displayed with white thick boundary.

Fig.7 (2) shows several kinds of pixels in Fig.7 (1). The brigthest pixels are pixels which locate inside isoline, the darkest pixels are background pixels, the second darkest pixels are boundary pixels, the second brigthest pixels specify the difference between Marching cubes and this paper. The new method regards the second brigthest pixels as the inner pixels of interested object, but Marching Cubes doesn't.

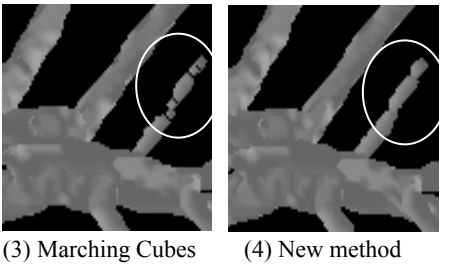


(1) image of blood vessel (2) various kinds of pixels

Fig. 7. Boundary of blood vessel from CT data



(1) Marching Cubes (2) New method



(3) Marching Cubes (4) New method

Fig. 8. 3D effect of the new method

The methods of this paper can be directly applied to 3D data field, detailed description is omitted. Based on the CT data of blood vessels, Fig.8(1) and Fig.8(2) is reconstructed respectively by Marching Cubes and the new method. Comparing the reconstructed models marked with circle, We can see that the models in Fig.8(1) are broken, but those in Fig.8(2) are connected. Fig.8(3) and Fig.8(4) show the same result.

5 Conclusion

Marching Cubes method regards the pixel value as a vertex value of its cube, which simplifies the meaning of the pixel value. We consider the voxel as a cube, which is not same as the Marching Cubes' cube. Under the assumptions in section 3, we try to recover the pixel data filed to original data field, and obtain topological consistent boundary of the interested object. Theoretic researches still remain to go deep, although experiments reach to better effects.

References

1. W. Lorensen and H. Cline. Marching cubes: a high resolution 3D surface construction algorithm, Computer Graphics, 1987,21(4):163-169.
2. M.J.Dűrst, "Additional reference to Marching Cubes". ACM Computer Graphics. 1988.22(2), P72-73.
3. Nielson G, et al. "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes". Visualization.1991.P83-91.
4. G M Nielson. "On Marching Cubes". IEEE transactions on visualization and computer graphics. 2003.9(3),P283-297.
5. KOBBELT L., BOTSCH M., SCHWANECKE U., SEIDEL H. P.: Feature-sensitive surface extraction from volume data. In Proc. of ACM SIGGRAPH (2001), pp. 57–66.
6. JU T., LOSASSO F., SCHAEFER S., WARREN J.: Dual contouring of hermite data. SIGGRAPH, (2002).
7. STANDER B. T., HART J. C.: Guaranteeing the topology of an implicit surface polygonization for interactive modeling. In Proc. of ACM SIGGRAPH (1997), pp. 279–286.
8. BOISSONNAT J.-D., COHEN-STEINER D., VEGTER G.: Isotopic implicit surface meshing. STOC (2004).
9. VARADHAN G., KRISHNAN S., SRIRAM T., MANOCHA D.: Topology preserving surface extraction using adaptive subdivision. In Proc. Symposium on Geometry Processing 2004 (2004).
10. http://www.bath.ac.uk/elec-eng/pages/sipg/research/morphology/morphology.htm
11. J.Serra, 'Image Analysis and Mathematical Morphology', Academic Press, ISBN: 0-12-637240-3.

Real-Time Model-Based SLAM Using Line Segments

Andrew P. Gee and Walterio Mayol-Cuevas

Department of Computer Science, University of Bristol, UK
{gee, mayol}@cs.bris.ac.uk

Abstract. Existing monocular vision-based SLAM systems favour interest point features as landmarks, but these are easily occluded and can only be reliably matched over a narrow range of viewpoints. Line segments offer an interesting alternative, as line matching is more stable with respect to viewpoint changes and lines are robust to partial occlusion. In this paper we present a model-based SLAM system that uses 3D line segments as landmarks. Unscented Kalman filters are used to initialise new line segments and generate a 3D wireframe model of the scene that can be tracked with a robust model-based tracking algorithm. Uncertainties in the camera position are fed into the initialisation of new model edges. Results show the system operating in real-time with resilience to partial occlusion. The maps of line segments generated during the SLAM process are physically meaningful and their structure is measured against the true 3D structure of the scene.

1 Introduction

Simultaneous Localisation and Mapping (SLAM) is the task of building a map of an unknown environment whilst simultaneously localising the position of the entity that is doing the mapping. Monocular vision-based SLAM considers the special case where the entity moving through the scene is a single camera. This case is interesting because it potentially offers a low-cost and real-time approach to SLAM in unprepared environments. The camera identifies natural features in the scene and uses these as landmarks in its map. Since a monocular camera cannot observe depth in one image, the measurements made in each frame consist simply of a bearing to each visible landmark.

The type of natural feature chosen for tracking will have a large effect on the design and performance of the SLAM system. Existing monocular vision-based SLAM systems have considered interest point features. These features are easy to detect and identify but are easily occluded and can only be reliably matched over a narrow range of viewpoints. Davison [1] presented the first real-time monocular SLAM system. The features used by this system are texture patches of interest regions detected with a Shi and Tomasi [2] saliency operator and matched using normalised cross-correlation. Molton et al. [3] modified this system to use 3D planar patches that warp their appearance to match the predicted camera viewpoint. This modification increases the range of viewpoints over which a feature

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 354–363, 2006.
© Springer-Verlag Berlin Heidelberg 2006

can be successfully matched but assumes that the underlying feature still has planar structure.

Furthermore, current visual SLAM approaches manage uncertainty by either maintaining a full covariance matrix [1,3] or by the use of particle filtering [4,5]. In the first case, this implies that map management demands an increase in processing that is quadratic to the number of features and in the case of particle filtering, a relatively large number of hypotheses have to be employed to represent the state of the system. Both cases have therefore limitations in terms of scalability.

In this paper an alternative approach to monocular vision-based SLAM is proposed that uses a combination of line segment features, robust estimation and model-based tracking. Section 2 discusses existing work using edge features for SLAM and outlines the approach that will be taken. Section 3 describes the model-based tracking system. Section 4 describes the automatic initialisation of new line segments and their integration into the model used for tracking. Finally, section 5 presents preliminary results illustrating the real-time operation of the SLAM system in some simple scenes.

2 Edge Features for SLAM

The use of edge features has been very popular in real-time tracking systems. Extraction and matching of edges is stable with respect to large viewpoint changes, robust to partial occlusion and efficient to perform in images. Edge features are typically less distinctive than interest point features and can suffer from the 'aperture problem', leading to ambiguity in localisation along the direction of the line. However, edge features with known positions can be combined to form a distinctive model that can be accurately tracked through an image sequence.

Despite the potential benefits of using edge features in a vision-based SLAM system, there has been very little work done in this area. Lemaire [6] has recently published results using 3D line segments as landmarks in an EKF SLAM framework for a mobile robot with odometry. Plücker coordinates are used to represent the 3D lines and new lines are initialised using a delayed Gaussian sum approximation algorithm. The matching of line segments between frames is aided by the association of Harris interest points in the image with detected line segments. This helps to increase their distinctiveness and reduces the possibility of mismatches. The system benefits from the viewpoint invariance of the line segments and should also be relatively robust to partial occlusion (although this is not demonstrated in the paper). However, the system relies on correctly matching line segments every frame to maintain tracking.

In this paper an alternative, model-based approach to monocular vision-based SLAM is proposed (figure 1). A combination of line segment features and model-based tracking is used to take advantage of existing robust algorithms developed for model-based tracking and reduce the problems associated with line segment matching and low level line segment extraction from images. New edge features generated during the SLAM mapping process are continually added to the

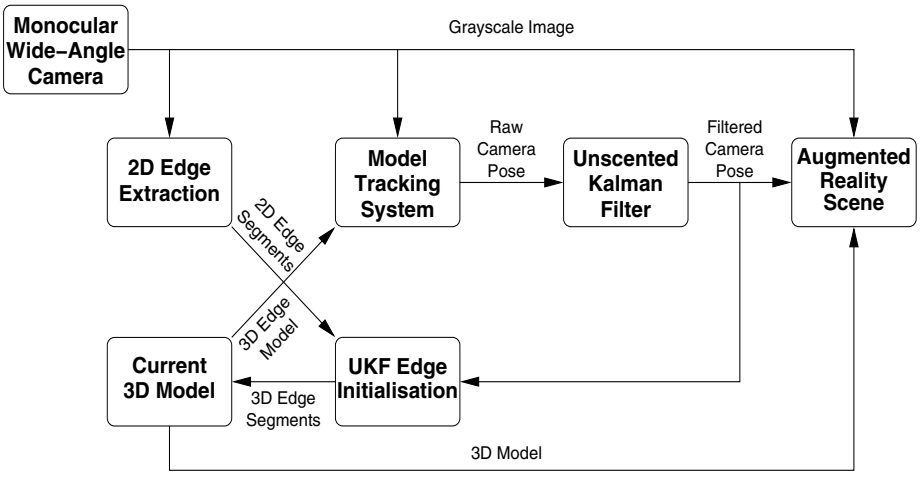


Fig. 1. Components of the model-based SLAM system. A camera provides grayscale images. The model tracking system estimates camera pose by fitting the current 3D model to strong image gradients. The optimised camera pose updates a UKF maintaining the estimated camera position. New 3D line segments are automatically initalised by a UKF, using 2D line segments extracted from the image, and added to the 3D model. The combination of image, camera pose and 3D model provides the information necessary to generate an augmented reality scene.

model used for tracking. This enables the model-based tracking system to maintain tracking in unknown areas, in contrast to previous systems that have been limited to tracking known CAD models. In addition to the benefits of stability over a wide range of viewpoints and robustness to partial occlusion, the map of line segments generated by this SLAM system provides information about the physical structure of the environment and can potentially be used to infer higher level structure in the scene.

3 Model-Based Tracking

Harris' RAPiD tracking system [7] introduced one of the first model-based tracking systems, where groups of 3D edge features are considered as a single 3D model that is fitted to strong gradients in an image in order to estimate camera pose. This approach has been popular because it avoids the problems associated with matching and extracting line segments, takes full advantage of the viewpoint invariance of edge features, and has low computational cost and complexity. Newer model-based tracking systems [8,9,10] have improved the robustness and performance of the original RAPiD tracking system but one significant drawback remains: the need for a wire-frame CAD model of the object to be tracked.

There are a number of model-based tracking systems that could be used as a basis for the model-based SLAM system discussed in this paper. Armstrong

and Zisserman's robust object tracker [8] addresses a number of problems in the original RAPiD system by employing RANSAC [11] and a 'leave-one-out' cross-validation scheme to remove outlying control points and model edges. Drummond and Cipolla [9] and Comport et al. [10] both use robust M-estimation and iteratively reweighted least squares (IRLS) to deal with outliers, but the virtual visual servoing (VVS) method presented by Comport et al. provides better computational efficiency and robustness [12].

In order to deal robustly with potential outliers in the model data, the model-based SLAM system adopts a combination of Comport et al.'s VVS system and a RANSAC estimation approach inspired by Armstrong and Zisserman's robust object tracker.

3.1 RANSAC Outlier Detection

The VVS system is designed for tracking a known CAD model with a reasonably large number of edges. Since the structure of the model is known, hidden line removal can be performed to handle occluded model edges. It is also known that every edge in the model is correctly positioned and corresponds to a real physical edge of the model. In contrast, the model-based SLAM system starts with a very simple model, a rectangle of known dimensions (see figure 3), to initialise tracking. New features will be added to this model as the camera moves through the scene, but it is quite likely that some of these features will be incorrectly initialised or correspond to false edges, such as the apparent contour of a curved surface or the edge of a shadow. Similarly, the model contains no information about occluding surfaces, so hidden line removal cannot be performed.

In order to deal with these problems, a RANSAC estimation approach is adopted at two different stages in the system. In the first case, RANSAC estimation is used to discard outlying edge points during the 1D search process, in the same way as [8]. In the second case, RANSAC estimation is used to discard outlying model edges during the camera pose estimation: putative camera poses are estimated from random samples of three edges using Comport et al.'s VVS method; for each putative pose, the remaining model edges are projected into the image and deemed to support that pose if the mean distance between the projected edge and the edge points detected in the image is less than a threshold; the putative camera pose with the most support is adopted and the final camera pose is re-estimated using the inlying model edges. Furthermore, each model edge keeps track of the number of times it has been detected as an inlier during the past 100 frames and uses this as a weighting factor in the VVS camera pose optimisation. Edges that are consistently detected as outliers are removed from the model permanently.

3.2 Kalman Filter Estimation

The modified VVS system computes the camera pose that minimizes the error between the projected 3D model and strong gradients detected in the image. It does not provide any information about the uncertainty of the computed

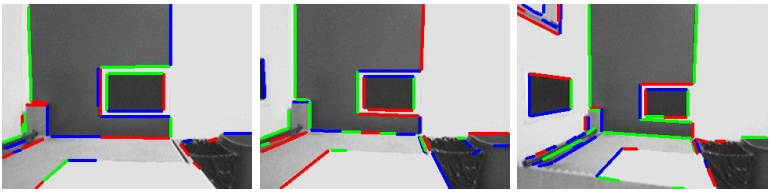


Fig. 2. A selection of images processed by the line segment extraction algorithm

camera pose. Incorporating a model of the camera uncertainty into the system is important because the camera pose is used to initialise new features in the SLAM map. In order to maintain an estimate of the camera pose and its associated covariance, the raw camera pose from the VVS tracking system is fed through an unscented Kalman filter (UKF) [13].

4 Mapping New Model Features

New 3D line segments are automatically initalised with a UKF and added to the 3D model. The approach uses 2D line segments extracted from the image to identify new line segments for initialisation and provide measurements for the UKF.

4.1 2D Line Segment Extraction

Line segments are detected and extracted from the camera images (figure 2) using a modified version of Gates' contour tracing algorithm [14]. This algorithm has been selected for its very fast performance. In our implementation of the algorithm a binary edge image generated using Canny edge detection is used as input and line segments are fitted to the extracted edge contours.

4.2 3D Line Segment Estimation

The depth of new lines is estimated with a UKF, by extending Pupilli's [4] approach for estimating the depth of point features to the case of line segments. A line segment initialised in the kth frame is represented by two endpoints with known unit vector directions $\{\mathbf{v}_k^0, \mathbf{v}_k^1\}$ from the camera centre of projection and unknown depths $\{d_k^0, d_k^1\}$. The state space model of the UKF contains the uncertain parameters for the line, namely the depths of the endpoints and the initial camera pose. The state model $\mathbf{x}_k$ for the filter initialised in frame k is $\{\mathbf{t}_k, \mathbf{q}_k, d_k^0, d_k^1\}$, where where $\mathbf{t}_k$ is the camera position in frame k and $\mathbf{q}_k$ is a quaternion representing the camera orientation. The 3D location of the endpoints $\mathbf{p}_m$ in the world co-ordinate frame is then calculated as:

$$\mathbf{p}_m(\mathbf{x}_k) = d_k^m \mathbf{R}_k \mathbf{v}_k^m + \mathbf{t}_k \tag{1}$$

where $\mathbf{R}_k$ is the rotation matrix generated from $\mathbf{q}_k$.

The UKF is initialised with the mean and covariance of the state parameters. The camera mean and covariance are taken from the camera's UKF (section 3.2) after updating the filter with the new camera pose measured this frame by the model-based tracking. However, it is impossible to say anything about the mean depth or variance of the line endpoints from a single observation. In [1,4] the depth and covariance of new point features are initialised using a 1D particle filter and only added to the Kalman filter once the depth distribution is approximately Gaussian. In contrast to this approach, we simply initialise the depth and variance of each line endpoint with a fixed value suitably chosen for the types of scenes we are observing. This avoids the delay in introducing features into the UKF state and avoids introducing another estimation mechanism (such as a 1D particle filter) into the system.

The state evolution model for the UKF is very simple. Since we assume that the scene structure is rigid, the state model parameters are all constants and process noise can be considered to be zero. Measurements to update the UKF are taken by projecting the predicted 3D line segment estimates from the UKF into the image and matching them to detected 2D line segments extracted from the image. The 2D line segments are parameterised by their centre point (x, y), orientation θ, and half-length l. The projected 3D line segment is matched to an extracted 2D line segment using Mahalanobis distance nearest neighbour matching within a 3σ search region defined by the projected uncertainty of the 3D line segment's centre point. The measurement equation is given by:

$$\mathbf{z}(\mathbf{x}_k) = \begin{pmatrix} \mathit{\Pi}(\mathbf{p}_0(\mathbf{x}_k), \mathbf{P}_k) \\ \mathit{\Pi}(\mathbf{p}_1(\mathbf{x}_k), \mathbf{P}_k) \end{pmatrix} + \mathbf{n}_k \tag{2}$$

where $\mathit{\Pi}$ is the pin-hole projection function for a calibrated camera, $\mathbf{P}_k$ is the mean camera projection matrix estimated in the kth frame, and $\mathbf{n}_k$ is the measurement noise from the image.

4.3 Automatic Feature Management

The addition of new features to the map is automated. Every N frames, a randomly selected 2D line segment that is not currently being initialised is added to a new UKF for initialisation. Each new line segment is assigned a confidence factor that increases for each successful observation in subsequent frames and decreases for each failed match. A line segment whose confidence factor falls to zero is removed from the UKF and discarded as unreliable. In this way, bad line segments are killed early in the initialisation process. The line segment UKFs are updated every frame with measurements from the image and added to the 3D model once the variance in their endpoint depths has decreased below a specified threshold (currently set at 0.5mm standard deviation). If a line segment has not converged within the first 150 frames after initialisation then it is eliminated. Lines that have been added to the 3D model are not used for tracking immediately. Instead, they are placed 'under watch' for 100 frames to test whether

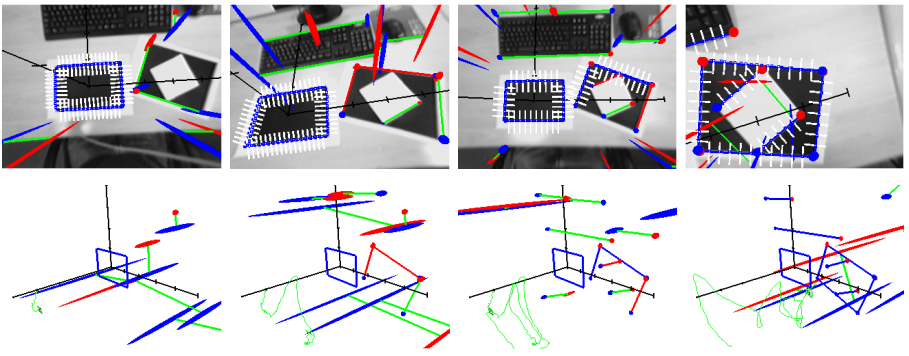


Fig. 3. Selected frames from a sequence demonstrating model-based SLAM in a desktop environment with well-defined, planar objects (please see `http://www.cs.bris.ac.uk/Research/Vision/Realtime` for the accompanying video). Each frame contains two subimages: (top) a camera image superimposed with the projected 3D model used for tracking; (bottom) a 3D model of the line segment map and a visualisation of the camera's track through the scene.

they agree with the estimated camera poses during this time. If a line segment that is 'under watch' is identified as an inlier in more than 80% of the frames, then it is accepted as a reliable line and incorporated into the model used for tracking. It is possible that this test will lead to bad lines being added to the model if the camera remains close to stationary during the test period, however the RANSAC estimation process (described in section 3.1) reduces the effect of these bad lines on the estimated camera pose and ensures that they are quickly removed from the model once the camera begins moving.

5 Results

Preliminary tests of the model-based SLAM system have been performed on a 3.0 GHz Pentium IV computer with 1 GB RAM and an NVidia GeForceFX 5200 AGP graphics card. The tracking system uses a calibrated Unibrain Fire-i handheld digital camera fitted with a wide-angle lens with 81° of horizontal FOV. We use camera images of size 320 x 240 pixels.

Figure 3 shows the system performing SLAM in a simple desktop environment using well-defined planar objects. A model of a known calibration pattern is used to initialise tracking. Each image in the sequence has two sections. The top section shows the camera image superimposed with the projected 3D model used for tracking (the blue wireframe). Around the edges of the projected model, white lines indicate the one-dimensional search region normal to each control point on the model. The blue squares indicate the positions of edge matches found along the search regions in the image. New edge features are drawn as green lines, with an ellipse at each endpoint to indicate 3σ uncertainty in the endpoint depths. Once a feature has converged, it is added to the 3D model and can be seen

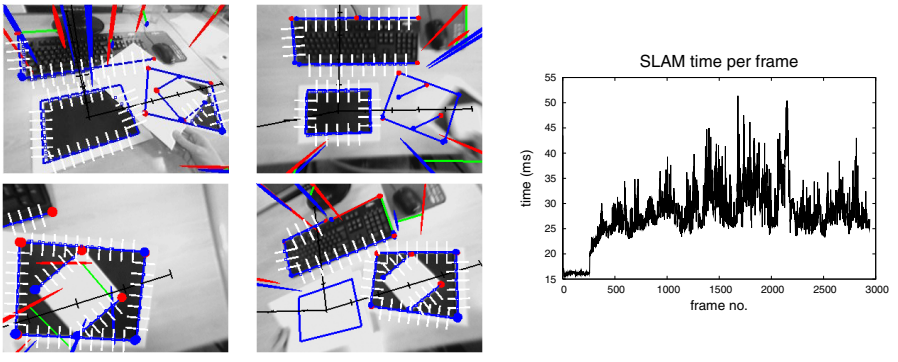
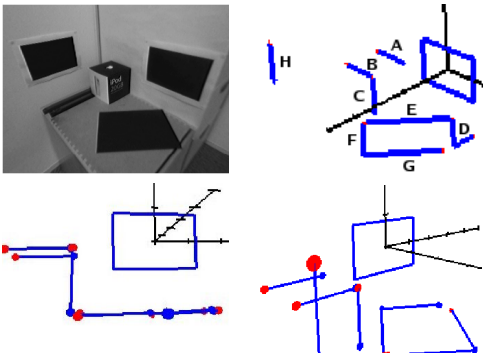


Fig. 4. (left) Robustness of the tracking system to various types of occlusion. From top-left to bottom-right: partial occlusion of the model; tracking when part of the model is removed; tracking when the original template is out of view; tracking when the original template is fully occluded. (right) Time spent on SLAM processing for each frame in the planar image sequence. Peaks in the processing time are caused by additional RANSAC sampling when the model is partially occluded.

projected in subsequent images as a normal model edge. Red lines represent new line segments that have converged but are currently being verified prior to being added to the model. The lower section of the images shows a 3D visualisation of the camera position and its track through the scene and the reconstructed 3D model of the SLAM map.

The SLAM algorithm typically runs in real-time through the planar sequence (figure 4 - right) with full RANSAC estimation of control points and model edges. Initial tracking using the known calibration pattern is smooth and, considering the pose of the augmentations, appears reasonably accurate. New features are gradually initialised on the second planar object and their estimated 3D locations can be seen to be approximately planar in the lower images (figure 3). Processing time does not increase significantly as new line segments are initialised, since the system does not maintain full covariance in the feature map. Once several line segments have been added to the map, it is possible to continue tracking even when the original template is occluded or out of frame (figure 4 - left). The system is robust to partial occlusion of the model, and this can include full occlusion of several model edges, but at least three partial model edges must be visible in each frame to maintain tracking.

The model-based SLAM system has also been tested with a sequence over a non-planar scene. The map of reconstructed line segments is compared to the measured dimensions of the scene, in order to estimate the accuracy of the reconstruction (figure 5). The scene is measured by hand using a ruler, with an estimated error of ± 0.5cm in the measurements. The results show that the reconstructed edges are typically accurate to within a few degrees of the true orientation and a couple of centimetres of the true position.



Edge	Error (°)	Error (±1.0cm)		
		x	y	z
A	1.9 ± 17.5	-0.4	0.4	0.2
B	1.7 ± 17.9	0.1	0.7	-0.5
C	2.3 ± 17.3	-0.1	-0.6	-0.3
D	5.1 ± 12.9	0.7	-0.2	2.1
E	2.2 ± 11.9	0.2	-0.4	0.9
F	2.2 ± 14.8	0.2	-1.9	0.3
G	2.4 ± 12.3	1.1	-1.5	-0.9
H	3.1 ± 16.4	-2.1	-2.6	1.6

Fig. 5. Selected frames from a sequence over a non-planar scene demonstrating the reconstruction of the physical structure of the scene. The error in the reconstructed model is expressed as the angular separation and mean distance between the end-points of the measured and reconstructed edges. Uncertainties in the estimated line segment position and errors in the measurement of the true scene are combined using standard error analysis to give an error range for the angular separation and distance calculations.

6 Conclusions

We have presented a real-time model-based SLAM system that combines line segment features, robust estimation and model-based tracking. The use of line segments provides stability over a wide range of viewpoints and robustness to partial occlusion, whilst the model-based tracking is fast and efficient and reduces the problems associated with line segment matching and extraction. The incorporation of RANSAC estimation enables the system to deal with errors in the reconstructed model and a UKF approach is presented for the initialisation of new line segments in the model.

Future work will consider a number of challenges. The current SLAM system maintains each model edge in an independent UKF and does not maintain full-covariance between the camera state and the line segments. This offers potential benefits in terms of scalability but means that the state estimates of the line segments in the model will be over-optimistic. Since the model-based SLAM tracking system does not use active search to match features, this optimism in the state estimate is less significant than it would be in other vision-based SLAM systems. The full implications of this approach remain to be explored, particularly when considering the extension of the model-based SLAM system to work over larger areas and with larger numbers of features.

Extension of the model-based SLAM system to work over larger areas will also require improved robustness. The current system is not robust to camera shake or rapid motion. If the camera has moved significantly between frames then the predicted position may not be close enough to the true camera position to ensure convergence of the model-based tracking, or the one-dimensional search region

may be too short to find correct edge matches for the model. However, edge features within the model form distinctive structures and it is hoped that these can be used as a mechanism for recovering from loss of tracking.

The use of VVS tracking and Gates' conic section extraction means that it should be relatively simple to extend the model-based SLAM system to deal with models containing conic sections as well as line segments, but the reliable initialisation of new edge features remains a significant challenge.

Acknowledgements. This work was funded by the UK EPSRC Equator IRC. Thanks to Mark Pupilli for useful discussions and use of his vision libraries.

References

1. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proc. of the Int. Conf. on Computer Vision. (2003)
2. Shi, J., Tomasi, C.: Good features to track. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. (1994) 593–600
3. Molton, N.D., Davison, A.J., Reid, I.D.: Locally planar patch features for real-time structure from motion. In: Proc. of the British Machine Vision Conference. (2004)
4. Pupilli, M., Calway, A.: Real-time visual slam with resilience to erratic motion. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. (2006)
5. Eade, E., Drummond, T.: Scalable monocular slam. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. (2006)
6. Lemaire, T., Lacroix, S.: Monocular-vision based SLAM using line segments. Technical report, LAAS-CNRS (2006) submitted to IROS 2006.
7. Harris, C.: Tracking with rigid models. Active Vision (1992) 59–73
8. Armstrong, M., Zisserman, A.: Robust object tracking. In: Proc. of the Asian Conf. on Computer Vision. (1995) 58–62
9. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Trans. on Pattern Analysis and Machine Intelligence **24** (2002) 932–946
10. Comport, A.I., Marchand, E., Pressigout, M., Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans. on Visualization and Computer Graphics **12** (2006)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
12. Comport, A.I., Kragic, D., Marchand, E., Chaumette, F.: Robust real-time visual tracking: Comparison, theoretical analysis and performance evaluation. In: Proc. of the IEEE Int. Conf. on Robotics and Automation. (2005) 2852–2857
13. Julier, S., Uhlmann, J.: A new extension of the kalman filter to nonlinear systems. In: Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls. (1997)
14. Gates, J., Haseyama, M., Kitajima, H.: A new conic section extraction approach and its applications. IEICE Trans. on Info. and Sys. **88-D** (2005) 239–251

Feature Correspondences from Multiple Views of Coplanar Ellipses

C. Barat, J.F. Menudet, H. Louhichi, and T. Fournel

Laboratoire Traitement du Signal et Instrumentation
UMR CNRS 5516 - Université Jean Monnet
18, rue Pr Benoît Lauras 42000 Saint-Etienne, France

Abstract. We address the problem of feature correspondences in images of coplanar ellipses with objective to benefit of robust ellipse fitting algorithm. The main difficulty is the lack of projective invariant points immediately available. Therefore, our key idea is to construct virtual line and point features using the property of tangent invariance under perspective projection. The proposed method requires first a robust detection of ellipse edge points to fit a parametric model on each ellipse. The feature lines are then obtained by computing the 4 bitangents to each couple of ellipses. The points are derived by considering the tangent points and the intersection points between bitangents. Results of experimental studies are presented to demonstrate the reliability and robustness of the feature extraction process. Subpixel accuracy is easily achieved. A real application to camera self-calibration is also described.

1 Introduction

Finding feature correspondences is an essential preliminary step in many Computer Vision tasks such as Structure From Motion or Stereovision. The most commonly used features are points and line segments. Their relations across multiple views have been extensively studied in the framework of projective geometry and led to the introduction of the fundamental matrix, the trifocal tensor or the homography matrix for the case of coplanar features [4]. It is well known that these entities can be estimated in a linear fashion from correspondences of points or lines [4] .

Nevertheless, line segments and especially points are rather instable features, not so easy to detect in noisy images. Several works have investigated the use of more complex primitives whose detection in images could be potentially more stable. Among possible primitives, conics have received a lot of attention because they play a central role in projective geometry and robust algorithms exist to compute them [10,2]. Estimation of fundamental matrix from at least 4 conics is addressed in [7]. In the case of two views of a conic with known epipolar geometry (i.e. known fundamental matrix), [5] proposes a solution to compute the homography associated with the conic plane. When the fundamental matrix is not available, two coplanar conics are necessary, as described in [8]. These results on conics are extended to general algebraic curves in [6] but at the expense of large systems of polynomial equations.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 364–373, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Unlike previous works, this article does not focus on multiple-view geometry of conics but rather gives a simple way to take advantage of conic fitting to construct virtual matching features with high accuracy. Essentially, the method we propose exploits bitangents to pairs of coplanar ellipses to obtain 4 lines and 14 points which are projective invariant, robust to noise and easy-to-compute.

Our method combines the strengths of point/line-based approach (simplicity, extensive background) and conic-based approach (stability). It could be used in numerous algorithms based on point correspondences such as camera self-calibration or motion estimation. It appears to be a solution to a problem arising when circular targets are used for point correspondences. Circular targets do not present any characteristic points. As a result, centroids of targets in images are classically used as feature points. And yet, they theoretically do not correspond to images of the target centers. This systematic error is not negligible when viewing distance is lower than 20 times the focal length. Moreover, the larger the circle, the greater the error. The projective invariance property of our method overcomes this limitation.

The next section details the computation of bitangents and associated points starting from images of two unknown ellipses. Then, an experimental study of features accuracy is presented in section 3. Finally, section 4 shows an application of the method to a real problem, namely camera self-calibration from multiple views of unknown coplanar ellipses.

2 Features Computation

The method we propose in this paper exploits the notion of bitangent to locate a set of reliable projective invariant features in images. A bitangent is defined as a line which is tangent to at least two objects. It has the property of being invariant to perspective projection.

When the two objects under consideration are two separated coplanar ellipses, there exist four bitangents, two internal (LR, RL) and two external (LL, RR). Letters L or R denote the relative position (left or right) of both ellipses relatively to bitangent (oriented from left to right on Figure 1(b)). These bitangent lines yield 14 projective invariant points which can be used as feature points for solving correspondences across multiple views of the two ellipses. (Figure 1(b)). Indeed, the 6 intersection points between bitangents and the 8 tangent points on ellipses remain unchanged under perspective projection.

Given an image of ellipses like the one of figure 1(a), the remaining question is how to compute the bitangents and the derived features with objectives of subpixel accuracy, robustness to noise, easiness and efficiency.

Description of the process. The process we suggest is composed of five steps:

- Ellipse edge detection
- Object labeling and ordering

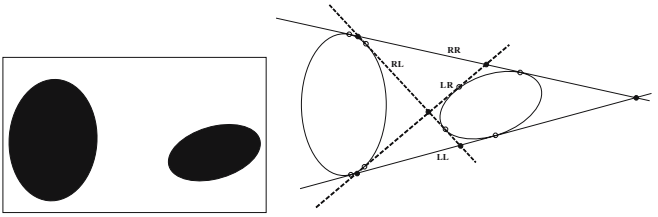


Fig. 1. Bitangents and feature points extraction (a) Original image (b) Common tangents(solid lines: external tangents, dashed lines : internal tangents) and associated feature points (filled circles : tangent points, rings : bitangent intersection points

- Ellipse model fitting
- Bitangents computation
- Feature points computation

The objective of the first three steps is to accurately estimate the parameters of the different ellipses and their relative positions with respect to a given pattern. Starting with ellipse edge points detection, a connected component labeling is then applied to identify each ellipse. Then, a sorting is done to match ellipses to target features.

Next, ellipses are processed by pair. Their parameter estimates serve as inputs to the algorithm for bitangent computation described later.

Finally, using bitangent equations, the different intersection points between bitangents and ellipses are calculated.

As the different steps are interlinked, the more robust the ellipse estimation, the more accurate the feature points. As a consequence, it is essential to perform ellipse edge detection with great care using a robust method. Let us take a closer look at the different steps and algorithms applied.

Ellipse edge detection. Images to process are images of an ellipses pattern from possibly oblique viewpoints. The amount of blur varies across the image plane in relation with the depths of the image points. Image blur and noise are 2 critical problems to address in order to guarantee accurate and stable features. A standard strategy is to adopt a multiscale method for edge detection. The algorithm chosen was proposed by Ducottet et al in [1]. It is a wavelet-based algorithm, which allows to detect and locate edges with an automatic adaptation of the scale to their smoothness. It is based on a modeling of contours as gaussian-smoothed singularities of three particular types (transitions, peaks and lines). The algorithm delivers a set of connected edge pixels per ellipse.

Object labeling and ordering. Once we have all edge points, we need to identify edge points of each ellipse. We apply a standard object labeling method to assign a number to each connected component. Then, we are concerned with

the problem of putting in correspondence the ellipses on the image with their corresponding features in space. This task starts by identifying the ellipse at the upper-left corner and consists then in making an ordered sorting of the ellipses from the reference one (from left to right and from top to bottom).

Ellipse model fitting. The point of this step is to determine the five parameters describing each ellipse. A least square ellipse fitting is applied to each labeled object as proposed in [2]. It is a robust method which provides coordinates of ellipse centers (u_0, v_0), axes lengths (a, b) and orientation θ.

Bitangents computation. The method described in [3] allows to compute the bitangents of two ellipses and the corresponding tangent points. It assumes that ellipses are defined by an analytic equation $aX^2 + bXY + cY^2 + dX + eY + f = 0$. Coefficients (a, b, c, d, e, f) can be trivially derived from parameters (u_0, v_0, a, b, θ) obtained previously. Bitangents are defined by equations of form $Y = UX + V$. All equations are combined together to give a polynomial in U of degree 4. In the general case of no parallel or vertical bitangents, it has four real single roots corresponding to the slopes of the four bitangents. Finally, the four corresponding values of V are computed and bitangents sorted according to their types (LL, LR, RL and RR).

Feature points computation. In [3], the author also gives equations to obtain the coordinates of the 8 tangent points. Furthermore, knowing bitangent equations, intersection points between bitangents can be easily calculated as line intersections. It leads to 6 feature points more. We get a total of 14 projective invariant points for a couple of ellipses.

3 Experimental Analysis

The question arising naturally is the accuracy of the computed features in function of data, i.e. ellipse edge points detected in images. An analytic derivation seems out of reach because of the equations involved (4th order polynomial). Therefore, Monte-Carlo simulations are performed in order to estimate mean error and standard deviation of feature points. The different experiments we have conducted allow to understand the influence of several parameters : the number of ellipse edge points, the relative position of 2 ellipses (distance, ellipse sizes, relative orientation) and the level of noise. We present here some results ascertaining the accuracy of our method. They concern only 13 feature points $\mathbf{p_k}$ (k=1..13) obtained from a setup of 2 ellipses, numbered as depicted in figure 2(a). The point resulting from intersection of bitangents LL and RR is discarded because of its instability. Indeed, this feature tends to infinity point in many natural configurations (such as 2 circles).

Protocol description. The study of the influence of the different parameters relies on the following steps:

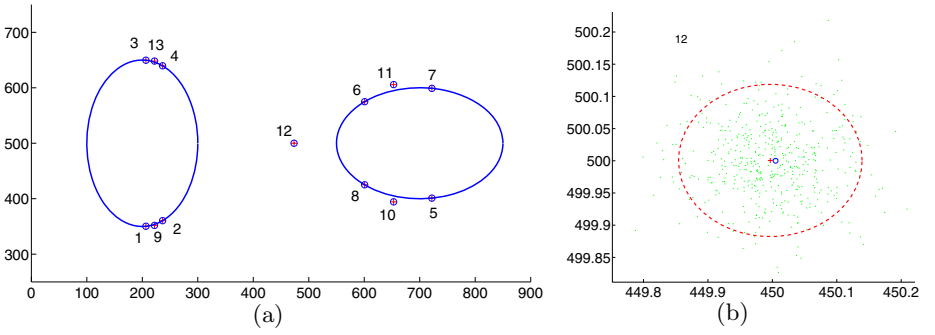


Fig. 2. (a)First setup: 2 ellipses with semi-axes 100 and 150 pixels and orthogonal orientation (b)Enlargement around feature number 12: circle (blue) gives the theoretical position, dots (green) are the 500 computed points with $N = 800$ and $\sigma_p = 1$ pixel. Cross and ellipse (red) are respectively mean and confidence ellipse (at two-sigma level) of point cloud.

- sampling of N points evenly spaced on ellipse boundaries
- corruption of points coordinates with gaussian noise with zero mean and standard deviation σ_p (in pixel)
- ellipses fitting on noisy points
- computation of the 13 feature points from fitted ellipses

For any specific set of parameters (N, σ_p), 500 trials are performed. The result is 13 point clouds of 500 points, associated with each of the 13 feature points. One of this point cloud (k=12) is displayed in figure 2(b). The circle gives the theoretical position of the feature point $\mathbf{p_k}$ and dots represent the computed feature obtained per trial.

The mean point $\bar{\mathbf{p}}_\mathbf{k}$(cross) and the covariance matrix of each point cloud are also computed. We denote $\Sigma_k^2 \geq \sigma_k^2$ the variances of a given point cloud along its principal directions. The confidence ellipse centered on mean point and with semi-axes $(2\Sigma_k, 2\sigma_k)$ is also drawn in figure 2(b).

Ellipse setup influence. The first experiment concerns the 2 ellipses displayed on figure 2, with orthogonal orientation.

Figure 3 presents results obtained with $N = 800$ (a point at every pixel) and $\sigma_p = 1$ pixel. The first graph shows the mean error, i.e. distance between the mean point $\bar{\mathbf{p}}_\mathbf{k}$ and the theoretical point $\mathbf{p_k}$. In the second graph, dark bar (blue) gives σ_k and lighter bar (green) Σ_k. The first conclusion is the negligible bias of the method, less than 0.02 pixels with this setup. The second conclusion is the different behaviors of features: clearly, features {5,6,7,8,10,11} are less stable than others. Especially, their uncertainty is more anisotropic, i.e. $\sigma_k \neq \Sigma_k$. The reason is the lower curvature of the ellipse at these feature points. Consequently, they are less constrained in the direction of tangent, resulting in an elongated point cloud.

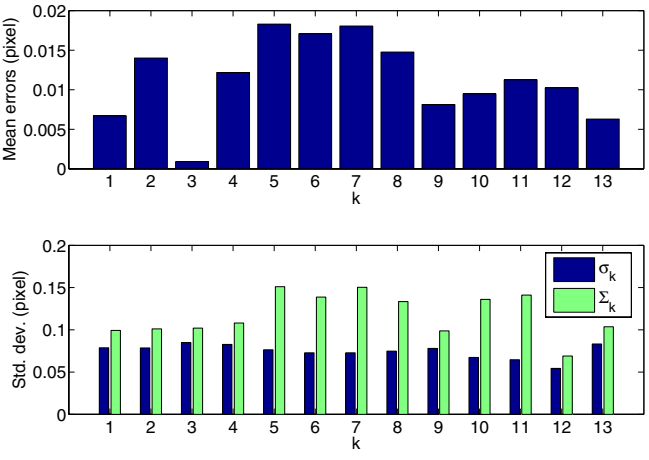


Fig. 3. Mean error (top graph) and standard deviations σ_k,Σ_k (bottom graph) of point clouds associated to the 13 features of figure 2(a) with $N = 800$ and $\sigma_p = 1$ pixel

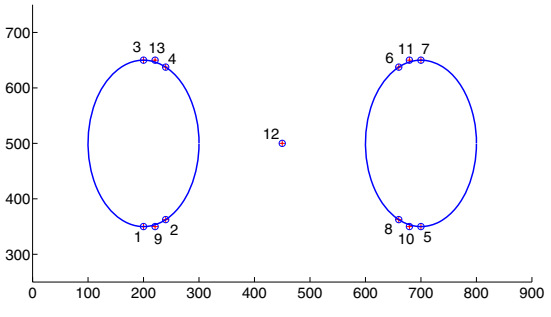


Fig. 4. Second setup: 2 ellipses with semi-axes 100 and 150 pixels and parallel orientation

We now consider 2 ellipses having the same orientation (Figure 4). The same experience as above is repeated and the results are displayed in figure 5.

We can see that all features are located in areas with highest curvature and behave identically, with approximatively $\sigma_k = 0.085$ pixel and $\Sigma_k = 0.1$ pixel. The mean error is still negligible. These values should be compared to the uncertainty in edge detection, i.e. $\sigma_p = 1$ pixel. This is to say that in this case, ellipse fitting brings a gain of 10 and allows to obtain sub-pixel measurement.

Parameter N influence. It seems obvious that the abovementionned gain is directly related to the number of points N used for ellipse fitting. This assumption is verified by studying $\bar{\Sigma}$ (the mean value of the Σ_k) as a function of N (with $\sigma_p = 1$). The resulting graph is shown in figure 6 (Left), with N varying from 5 (the minimum to fit an ellipse) to 1000. The hyperbolic decrease of $\bar{\Sigma}$ is manifest and demonstrates the efficiency of ellipse fitting in our application.

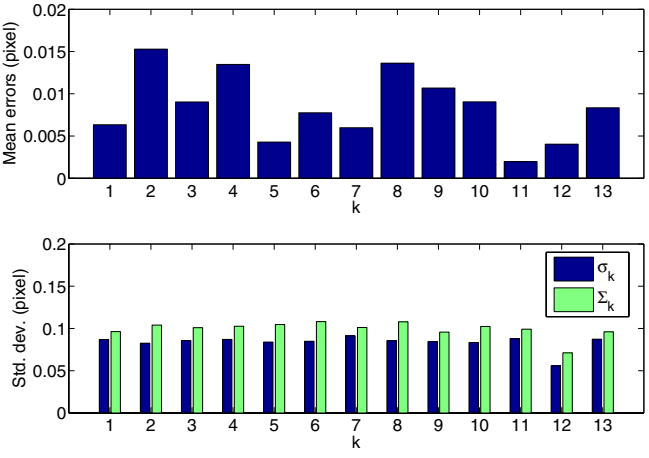


Fig. 5. Mean error (top graph) and standard deviations σ_k, Σ_k (bottom graph) of point clouds associated to the 13 features of figure 4 with $N = 800$ and $\sigma_p = 1$ pixel

In practice, it is quite easy to detect few hundreds of contour points at pixel precision in images. Hence, sub-pixel accuracy is guaranteed for the proposed 13 feature points.

Level of noise influence. Considering ellipses of the previous experiment and $N = 800$, figure 6 (Right) gives $\bar{\Sigma}$ for σ_p varying from 0 to 2 pixels. This last experiment shows the linear degradation of our algorithm with respect to the level of input noise σ_p.

4 Application to Camera Self-calibration

Camera calibration aims at the computation of camera intrinsic parameters from image(s) of a perfectly known object. This is a basic problem in computer vision when metric information are required from images. Intrinsic parameters allow to relate coordinates $[X\ Y\ Z]^{\mathrm{T}}$ of a 3D point (in camera frame) and coordinates $[u\ v]^{\mathrm{T}}$(in pixel) of its projection in image:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \tau f & 0 & v_0 \\ 0 & f & u_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}$$

where f is the focal length, τ the aspect ratio of pixels and $[u_0\ v_0]^{\mathrm{T}}$ coordinates in image of principal point. This is the standard pinhole model which made assumptions of negligible optical distortions and zero skew camera. In practice, aspect ratio τ is very close to 1 in most of modern CCD.

Camera self-calibration is an (harder) variant of the camera calibration problem occurring when the scene is unknown. In this case, the basic data is

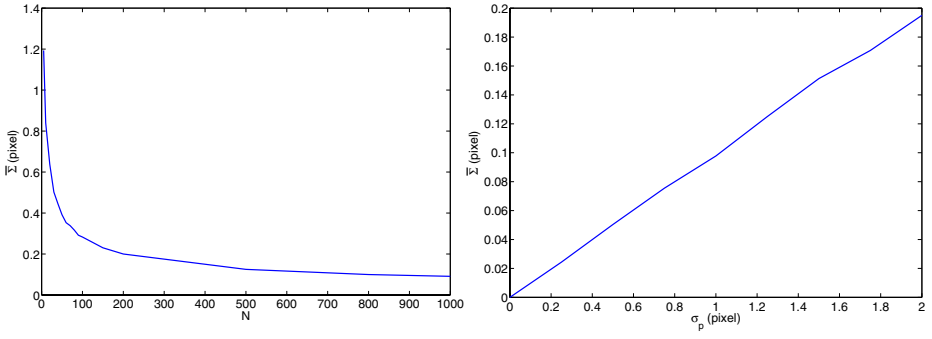


Fig. 6. Mean value of the Σ_k in function of: (Left) the number of points N used for ellipse fitting and with input noise level $\sigma_p = 1$ pixel; (Right) the input noise level σ_p and with $N = 800$

correspondences of points in images. Therefore, camera-self-calibration could benefit of our method. Nevertheless, it is well-known that basic self-calibration algorithms suffered from degeneracy when correspondences come from coplanar 3D points, which is precisely the case in our approach. Specific self-calibration algorithms have been proposed to handle this case [11]. This technique, known as plane-based self-calibration, only required the homographies relating the different views of an unknown planar scene.

We propose to find such homographies from a planar scene made of 5 circles as shown in figure 7. Note that the fact we have circles in practice will not be used by our algorithm unlike circle-based calibration methods [12]. 6 images (resolution 1600×1200) of this scene are taken with a digital still camera from different viewpoints. 104 feature points per image are computed using adjacent ellipses. Then, inter-view homographies are estimated using these features and the normalized DLT algorithm [4]. The residual RMS error is about 0.15 pixel which is in agreement with simulation results.

Without changing settings of camera (zoom, focus), 10 images of planar calibration pattern are acquired. Thus, Zhang's calibration method [13] can be applied. The result will be used as ground truth values of parameters. They are compared with the output of our plane-based self-calibration algorithm (initialization with Triggs's method followed by bundle adjustment). Results are presented in table 1. Relative errors are computed for focal length ($|\Delta f/f|$), principal point ($|\Delta u_0/f|$, $|\Delta v_0/f|$) and aspect ratio ($|\Delta\tau/\tau|$). Principal point and aspect ratio are rather well estimated whereas focal length result seems less accurate. This is surprising because usually principal point is the most difficult parameter to estimate. Therefore, simulations with known camera parameters have been performed. They have not shown specific instability of focal length estimation. We concluded that the problem is not inherent to the method but probably due to the images themselves. More experiments are necessary but we are convinced that it could be a valuable approach for camera calibration.

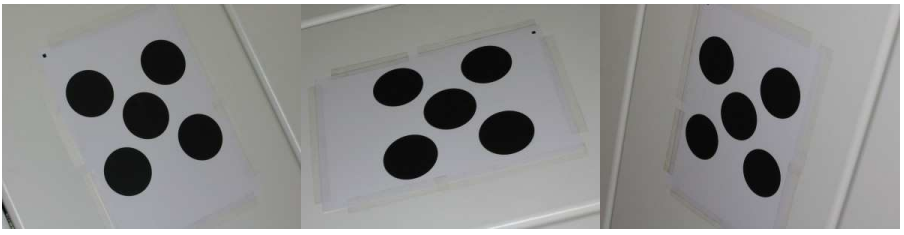


Fig. 7. 3 views (among 6) of 5 coplanar ellipses (circles in practice)

Table 1. Comparison between plane-based calibration and self-calibration methods. Relative errors are computed with calibration method as reference.

Param.	Calib.	Self-Calib.	Relative Errors	
f	4140.13	4065.76	$\lvert\Delta f/f\rvert$	1.81%
u_0	789.91	787.74	$\lvert\Delta u_0/f\rvert$	0.05%
v_0	609.22	636.15	$\lvert\Delta v_0/f\rvert$	0.65%
τ	0.9988	0.9971	$\lvert\Delta\tau/\tau\rvert$	0.17%

5 Conclusion

This article proposes an ellipse-based method for computation of invariant features in images. The use of bitangent lines to a couple of coplanar ellipses is the key idea that yields projective invariance. Hence, invariant points can be derived from intersections of bitangents with ellipses and other bitangents. These points and lines finally provide feature correspondences across multiple views. Advantages of this approach are two fold. First, ellipse fitting brings accurate and stable features (typically 0.1-0.2 pixel) from simple edge detection at pixel precision. Several tasks could benefit of the method, such as fiducial-based tracking for Augmented Reality or wide-baseline matching. In these contexts, the most critical step is clearly the automatic detection of ellipses in images. Second advantage is the extensive background of feature-based methods that can be subsequently used. For example, homography relating images of 2 or more ellipses can be directly estimated in this way. Note that the former solution of [8] is more complicated although it also involves roots of a polynomial of degree 4. Indeed, it needs the computation of intersection points between ellipses which are (in the general case) points with complex coordinates, possibly at infinity.

A practical application of our method was demonstrated with camera calibration from images of several unknown ellipses. Inter-images homographies are estimated from invariant feature points and plane-based self-calibration algorithm is applied. Results are quite satisfying although they surely can be improved, for example with subpixel edge detection of ellipses. Comparison with circle-based calibration methods should be done to evaluate the potential of our approach. Another important point is the integration of features uncertainty.

Indeed, experiments have shown that the covariance matrix of each feature is related to ellipse curvature or angle between bitangents. In future work, we plan to predict such covariance matrices in order to incorporate them in feature-based methods allowing it (such as bundle adjustment).

References

1. C. Ducottet, T. Fournel and C. Barat, Scale-adaptive detection and local characterization of edges based on wavelet transform, *Signal Processing*, 84: 2115-2137, 2004.
2. A. Fitzgibbon, M. Pilu, and R. B. Fisher, Direct Least Square Fitting of Ellipses, In *IEEE Trans. PAMI*, 21(5), pp. 476-480, 1999.
3. L. Habert, Computing bitangents for ellipses, In Proc. 17th Canadian Conference on Computational Geometry (CCCG'05), 2005, 294-297.
4. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, *Cambridge University Press,* 2002.
5. C.Schmid and A.Zisserman, The geometry and matching of lines and curves over multiple views, In *International Journal on Computer Vision*, 40(3), pp. 199-234, 2000.
6. J.Y. Kaminski and Amnon Shashua, Multiple View Geometry of General Algebraic Curves, In *International Journal on Computer Vision*, 56(3), pp. 195-219, 2004.
7. F. Kahl and A. Heyden. Using Conic Correspondence in Two Images to Estimate the Epipolar Geometry, *Proceedings of the International Conference on Computer Vision*, 1998.
8. P. K. Mudigonda C. V. Jawahar P. J. Narayanan, Geometric Structure Computation from Conics, In *Proceedings of Indian Conference on Computer Vision, Graphics & Image Processing*, 2004
9. C. A. Rothwell, A. Zisserman, D. A. Forsyth, J. L. Mundy, Planar Object Recognition using Projective Shape Representation, *International Journal on Computer Vision*, 16(1), pp. 57-99, 1995.
10. S.P. Sampson, Fitting conic sections to very scattered data: an iterative refinement of the Bookstein algorithm, In *Computer Graphics and Image Processing*, vol. 18, pp. 97-108, 1982.
11. B. Triggs, Autocalibration from planar sequences, In *Proc. of the European Conference on Computer Vision,* Freiburg, Germany, June 1998.
12. Y. Wu, X. Li, F. Wu, Z. Hu, Coplanar circles, quasi-affine invariance and calibration, *Image and Vision Computing,* 24(4), pp. 319-326, April 2006.
13. Z. Zhang, A flexible new technique for camera calibration, *IEEE Trans. PAMI,* vol. 22, pp. 1330-1334, 2000.

Evaluation of Subpixel Tracking Algorithms

Johan Skoglund and Michael Felsberg

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden
{skoglund, mfe}@isy.liu.se

Abstract. Evaluation of tracking algorithms can be done in several different ways, either using real or synthetic images. The main benefit with the second alternative is that the environment is completely controlled, there is no problem to get the ground truth and the noise is well known. This paper contains the results from an evaluation of subpixel tracking algorithms. The main focus of the evaluation is to compare the performance of subpixel methods with different computation complexity, in order to see whether the tracking performance justifies more complex algorithms.

1 Introduction

Motion information from images is useful for many applications like pose estimation, object tracking and video compression. In this paper we focus on one type of method for doing this, called block matching. Common for these methods is that a patch is given and the algorithm tries to find the best match in an image. Choosing the best tracking algorithm is a complex problem because the evaluation of the different methods can be done with different focus.

In its most basic form, the matching algorithm generates the best matching pixel-position. By adding subpixel accuracy to this matching it is possible to increase the precision for applications that mainly relies on the precision of the tracking. The subpixel accuracy is however usually not for free, it requires more calculations. In this paper we presents the result from an evaluation of the subpixel accuracy for algorithms with different complexities.

2 Block Matching

Block matching is a common name for all algorithms that try to find the position of an patch p, in an image b. This can be done in a number of different ways but the goal is to find the minimum of an error function

$$\min_{\gamma} E(p, T(b, \gamma)) \tag{1}$$

The error function can be separated into two important functions, T and E. The function T transforms the image, e.g. rotates, scales and cuts out a region from

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 374–382, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the image. E measures the difference between the patch and the region generated by T. The parameters γ which steers the transformation T and minimizes the error function are estimated.

2.1 L1 vs L2 Norm

Although any function could be used for measuring the difference between the patch and image in (1), it is very common to use a version of the L_1 or the L_2 norm, i.e. *sum of absolut difference* (SAD) or *sum of squared difference* (SSD), respectively.

$$E_1 = \sum_{x,y} |p(x,y) - b(x,y)| \tag{2}$$

$$E_2 = \sum_{x,y} (p(x,y) - b(x,y))^2 \tag{3}$$

From a theoretical point of view the L_2 norm is most attractive since the theory for least square problems is well known and the measurement is the energy of the distance. For practical situations however, the L_1 norm might be superior [1].

- L_1 norm is more robust to outliers, outliers influence the result for the L_2 norm more than of the L_1 norm.
- A brute force L_1 norm algorithm executes faster on a modern PC [2] since it contains special instructions for this operation. L_1 is better suitable for special hardware since the absolute value is easier to compute than the square.
- The L_2 norm is however often used in iterative methods, methods which step by step minimize the error function, since minimization of least square problems are well studied, for example the well known KLT algorrithm [3].

It is also common to normalize the patch and the image before matching to reduce the influence of an illumination change. The normalized cross correlation (NCC) is an example which normalizes both the energy and the dc-level and measures the distance using the L2 norm [4]. First the dc-levels $\overline{p}$, $\overline{b}$ are subtracted from the patch and the image, and afterwards the measured distance is divided by the norm of the patch and the image.

$$E_{\mathrm{NCC}} = 1 - \frac{\sum_{x,y} (p(x,y) - \overline{p})(b(x,y) - \overline{b})}{\sqrt{\sum_{x,y} (p(x,y) - \overline{p})^2} \sqrt{\sum_{x,y} (b(x,y) - \overline{b})^2}} \tag{4}$$

Both methods based on the L_1 and the L_2 norm are compared in this paper.

3 Subpixel Accuracy

The result from the block matching in its most basic version, is the best matching pixel result. It is however often desirable to reduce the average error in the matching and one way to do this is to add subpixel accuracy to the tracking. This improvement can be done in two different ways:

- Interpolation of the image; the dimension of the image/patch is usually quite large and the interpolation is therefore time consuming.
- Interpolation of the error function; faster since fewer values has to be interpolated but will probably also give less accuracy.

3.1 Interpolation of Error Function

The output from the matching is for a given position a measurement of the error, the difference between the patch and the image. One way to get subpixel accuracy is to interpolate this error function. The interpolation is done by fitting a suitable function to the measured errors within a region around the pixelbased minima. Which function that should be used depends mainly on two factors:

- The chosen function should correspond to the function used for measuring the error.
- Function fitting might generate a nonlinear problem which takes long time to solve. For practical problems it might be better to simplify the problem to faster get a solution with less accuracy.

Taylor expansion of (1) in $\lceil\gamma_0\rceil$ and $\lfloor\gamma_0\rfloor$ gives

$$E_1(\gamma) \approx E_1(\gamma') + (\gamma - \gamma')\frac{dE_1}{d\gamma}(\gamma') \quad \gamma' = \lceil\gamma_0\rceil, \lfloor\gamma_0\rfloor \tag{5}$$

We assume a linear symmetric model of the error function. $\frac{dE_1}{d\gamma}$ is estimated from an interval not containing the minimum γ_0 of the error function. In this derivation we assume, without loss of generality that the value to the left of the minimum is larger than the value to the right, i.e. the minimum γ_0 of the continuous error function is to the right of the pixel-based minimum $\lfloor\gamma_0\rfloor$. The solution to the opposite case follows analogous. We estimates the derivative as

$$\frac{dE_1}{d\gamma} \approx [E_1(\lfloor\gamma_0\rfloor) - E_1(\lfloor\gamma_0\rfloor - 1)] \tag{6}$$

Using this estimation in the two version of (5) gives

$$E_1(\gamma) \approx E_1(\lfloor\gamma_0\rfloor) + (\gamma - \lfloor\gamma_0\rfloor)\frac{dE_1}{d\gamma}(\lfloor\gamma_0\rfloor) \tag{7}$$

$$E_1(\gamma) \approx E_1(\lceil\gamma_0\rceil) - (\gamma - \lceil\gamma_0\rceil)\frac{dE_1}{d\gamma}(\lceil\gamma_0\rceil) \tag{8}$$

Combining (7) and (8) gives that the displacement d

$$d = \gamma_0 - \lfloor\gamma_0\rfloor = 1 - \lceil\gamma_0\rceil + \gamma_0 \tag{9}$$

between minimum and the center point is estimated as

$$d = \frac{E_1(\lfloor\gamma\rfloor - 1) - E_1(\lfloor\gamma\rfloor + 1)}{2(E_1(\lfloor\gamma\rfloor - 1) - E_1(\lfloor\gamma\rfloor))} \tag{10}$$

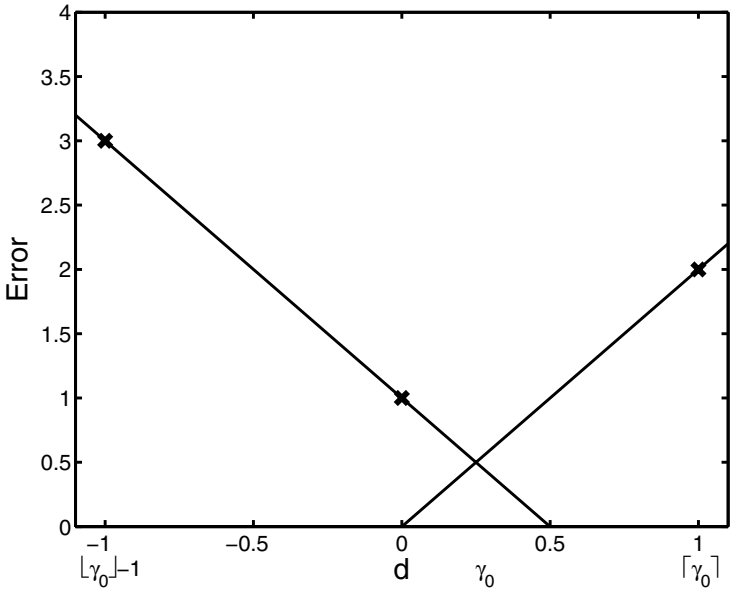


Fig. 1. Example of cone fitted to 3 points

Figure 1 shows a geometric view of the basic principle. First the steepest possible line is drawn drawn through the centerpoint and one of the other points. Then a line is drawn through the remaining point with the same slope. The minimum is found found as the intersection between the two lines. This corresponds to fitting a 1D cone to the measured errors. For 2D error functions one can apply this method for each coordinate separately assuming that the function is separable. A proper 2D model of cone fitting is parameterized in this way:

$$cone(P) = \frac{(P-C)\mathbf{S}(P-C)^t}{|P-C|} + h \tag{11}$$

where $P = [x, y]$ is a point, C is the center of the cone, S is a symmetric 2x2 matrix which gives the shape of the cone and h gives the minimum. The least squares fitting of (11) to the data is a nonlinear problem which can be solved by the Gauss Newton method. This minimization is both time consuming and gives more advanced code compared to the simpler model.

4 Evaluation Setup

The evaluation of tracking algorithms has been done within the MATRIS project[1]. The goal of this project is to develop a real time system for pose estimation of

[1] Homepage www.ist-matris.org



Fig. 2. Synthetic test image

cameras. One central part of this system are efficient algorithms for patch tracking in video sequences. The creation of these patches has been done in a initialization step and the illumination might therefore be different between the image and the patches. It is therefore desirable if the tracking algorithm works reasonable well even if the patch and image differ in lighting conditions. To achieve this all tracking algorithms in the evaluation are implemented invariant to dc-changes.

There is a number of problems with an evaluation of tracking algorithms. The most important problem is probably to generate the ground truth without bias. In this evaluation, we solve this problem by using a synthetic image sequence generated from real textures. See figure 2 for an example of an image. With this method it is possible to generate the ground truth and to simulate illumination changes and motion blur.

4.1 Experimental Setup

To generate the test images a number of tools used or developed in the MATRIS project was used. The test data was created using this procedure:

1. Create a textured 3d-model consisting of planar patches from a number of real images.
2. Render a sequence of images showing the model from different poses. Save the 2D center position for the planar patches together with the image.
3. Start to create patches and predict the position for patches in all images. This is done with software developed in the project.

4. Estimate the start pose for the camera, the position of the camera for the first image using the image content.
5. Warp all patches that are visible from the estimated pose using a homography and save these patches.
6. Track the position for all visible patches and use this to improve the pose, combining the 3D-model and patch positions.
7. Iterate step 5-6 for all images.

This gives a number of images together with almost correctly transformed patches and their correct position. These images are then modified to simulate noise. The results from the different tracking algorithms are compared below.

5 Evaluation

In the evaluation, three different methods for generating subpixel accuracy have been compared.

1. KLT. The Kanade-Lucas-Tomasi gradient descent method with linear interpolation of the image.
2. L1. Brute force matching using a DC-invariant L1 norm. Simplified cone model.
3. L1. Full 2d cone model.

The two most interesting measurements for comparing different tracking algorithms are the number of outliers e.g [5], and the empirical error for inliers. These two measurements are however more or less independent, there is probably no reason to believe that a low number of outliers should influence the accuracy for inliers or vice versa. These two measurements can therefore be evaluated independently of each other. In this evaluation we look at the average error for inliers, the RMS error measured in pixels between ground thruth and the estimated match.

5.1 Noise Model

To simulate the noise in a camera a number of different models for the noise are possible. Each model have different pros and cons regarding plausibility, simplicity and correctness, and no noise model is better in all ways. For this evaluation we are using additive Gaussian and Salt and Pepper noise. One can argue that these models might not be the most realistic but we decided nevertheless to use them for two reasons:

1. Simplicity, multiplicative or additive noise is the easiest model to use.
2. Gaussian noise is commonly used when no other model is especially motivated.

Another source of noise that is important is the error introduced by geometric distortions like rotations and wrong viewing angle. Locally, the effect of geometric distortions might be large. In this evaluation the pose of the camera is estimated with a small error. Therefore the warping of the patches will be slightly wrong. Except of this, the effect is neglected in this evaluation.

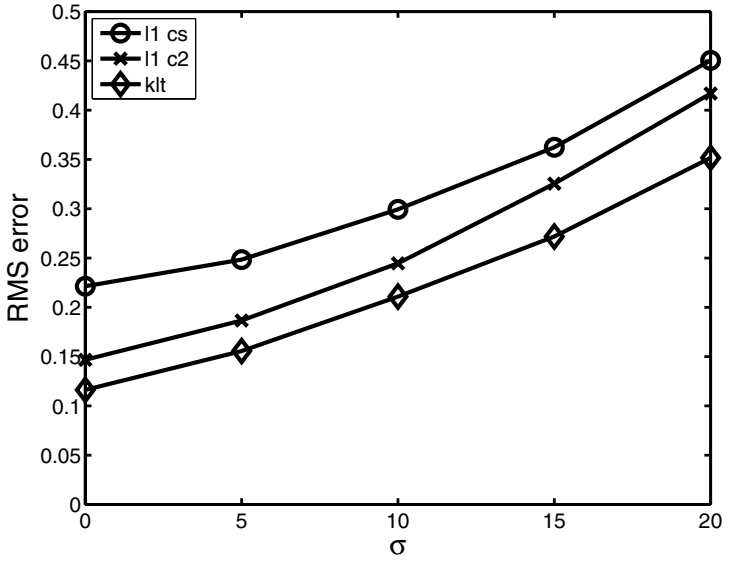


Fig. 3. RMS pixel error for different methods, Gaussian noise

6 Results

Evaluation of a tracking algorithm can be done with focus on different parameters. For this evaluation we focus on the subpixel-accuracy for different methods and add Gaussian or Salt and Pepper noise to see how robust the results are.

Figure 3 shows the RMS error for the different algorithms for subpixel accuracy and with different amount of Gaussian noise in a RGB image with values between 0 and 255.

The subpixel RMS error for all these methods is significant better than the RMS error we would get with the nearest pixel. For a pixelbased method, in the best case where we always find the nearest pixel, the error would have an uniform distribution between -0.5 and 0.5 and this would yield an RMS error of $\sqrt{\int_{-0.5}^{0.5}\int_{-0.5}^{0.5}(x^2+y^2)dxdy} \approx 0.4$.

Figure 4 shows the RMS error with Salt and Pepper noise. Both L_1 based methods are also here able to reduce the RMS error. The accuracy for the L_2 method decreases however rapidly when the amount of noise is increased.

As we could expect, the average error is largest for the fastest method and is lower for the more complex algorithms. The method using linear interpolation of the image gives the smallest error with Gaussian noise but is also the most time consuning method and does not work especially well with higher amount of Salt and Pepper noise. All the tested methods significantly decreases the error compared to a pixelbased method, possibly except KLT and high amount of Salt and Pepper noise. The difference in the average error between the best and the worst method is fairly small. For many applications with time constraints, it is

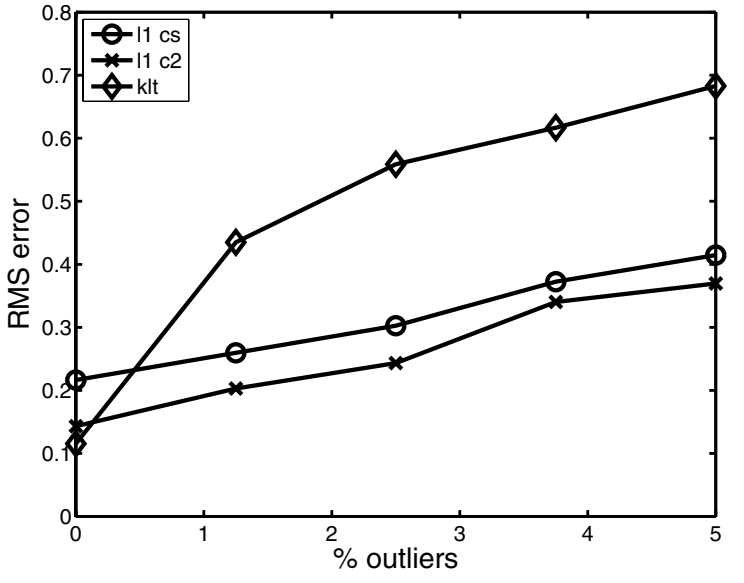


Fig. 4. RMS pixel error for different methods, Salt and Pepper noise

probably better to use a faster method which is able to handle more patches than a slower method with higher accuracy. More patches increases the probability that we always have sufficient with inliers and it might also be possible to improve the accuracy by averaging several results.

7 Conclusions

We have compared three different methods for subpixel accuracy in tracking. For Gaussian noise, interpolation of the image gave a slightly smaller error than the other methods. This method is however, the most time consuming method and did not work especially well with Salt and Pepper noise. Our conclusion is therefore that the accuracy in many applications is probably higher if we are able to use a faster method and instead increase the number of patches. Compared to a "nearest pixel" method the gain by introducing subpixel-estimates was significant even for the most basic method and should therefore always be used. If the noise characteristics for the images is unknown or we can expect Slat and Pepper noise we suggest that the more robust L_1 norm should be preferred.

Acknowledgment

This work has been supported by EC Grant IST-2002-002013 MATRIS.

References

1. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision **19** (1996) 57–92
2. Skoglund, J., Felsberg, M.: Fast image processing using sse2. In: Proceedings of the SSBA Symposium on Image Analysis, Malmö (2005)
3. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81). (1981) 674–679
4. Crowley, J., Berard, F., Coutaz, J.: Finger tracking as an input device for augmented reality (1995)
5. Banks, J., Corke, P.: Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision. The International Journal of Robotics Research **20** (2001) 512–532

Adaptive Real-Time Rendering for Large-Scale Molecular Models

Jun Lee[1], Sungjun Park [2], and Jee-In Kim[3,*]

[1] Computer Science & Engineering, Konkuk University, Seoul, Korea
junlee@konkuk.ac.kr
[2] Game Engineering, Hoseo University, Cheonan, Korea
sjpark@office.hoseo.ac.kr
[3] CAESIT, Konkuk University, Seoul, Korea
jnkm@konkuk.ac.kr

Abstract. Real-time surface rendering of large-scale molecular models such as a colon bacillus requires a great number of polygons to be displayed on a display device. Since a long latency of display and manipulation is fatal in maintaining presence in a virtual environment, high performance computing power and high quality graphical components are required to exercise real-time rendering of such a large-scale molecular model. We propose an algorithm which enables a PC level computer to render and display large-scale molecular models in real-time. The proposed algorithm adaptively visualizes large-scale molecular models. We tested our algorithm with molecular models of which polygons range from 533,774 polygons to 2,656,246 polygons. Our experiments showed that frame rates of displaying and manipulating the models ranged from 17.85 to 55.64 frames-per-second. The frames rates are 4.3 to 6.9 times higher than those of the models which are obtained using a conventional system. Our system enables biologists to display and manipulate large-scale molecular models in real-time which could not be done fast enough to be used in a virtual environment using the conventional systems.

1 Introduction

Surface rendering of molecular models [1] is a process of generating three dimensional shapes of molecular structures on a display device. The three dimensional structures play important roles during analysis of molecular structures, interactions between proteins, and design of new drugs.

As a molecular weight of a molecular model increases, the number of polygons, which become surfaces of the molecular model, rapidly increases. Therefore, high performance computing power and high quality graphical capability are required to display three dimensional models of large molecules in real-time. If such a high performance computer system is not available, biologists must find alternatives. For example, the number of polygons could be reduced, if points (instead of polygons)

*Corresponding author.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 383 – 392, 2006.
© Springer-Verlag Berlin Heidelberg 2006

could be used in modeling large molecules. However, such alternatives could result in inaccurate images of molecular models.

In this paper, we propose an algorithm of real-time rendering based on an adaptive level of detail (LOD) technique which is widely used in representing complex geometrical objects for different requirements and applications [10-14]. Our algorithm can reduce a number of polygons for representing a large-scale molecular model. The polygon reduction process is adaptively applied so that users can get a certain level of frame rates. It includes an adaptive LOD process where large-scale molecular models are displayed at different levels depending on distances from users at required frame rates. The visual and geometrical features of the molecular models should be maintained during the adaptive LOD process. Even if the algorithm was executed on a practical PC class computer which had a general graphics card with a GPU (Graphic Processing Unit), our tests showed that the models could be displayed and manipulated in real-time. The algorithm was implemented and executed on our visualization system for molecular models, VRMMS (Virtual Reality Molecular Management System) [2, 3]. The tests were exercised and compared with a conventional molecular visualization system, VMD (Visual Molecular Dynamics) [6]. The results showed that our system offered 4.3 to 6.9 times higher frame rates than those offered by VMD.

2 Related Works

With current surface rendering algorithms [4, 5], it is difficult to execute real-time rendering of large molecular models. It is because the numbers of polygons of the molecular models rapidly increase as their molecular weights increase. Therefore, VMD [6], which is the most popular tool for molecular modeling, now accelerates the rendering process using a GPU on a specialized graphics card. Since VMD is running on a high performance computer at the level of a graphical workstation equipped with a specialized graphics card, we cannot achieve satisfactory performance of real-time rendering using a PC with an ordinary graphics card. [1]

LOD (Level of Detail) [8,9] can be used in reducing the numbers of polygons of the molecular models. The method simplifies polygon meshes of the molecular models and applies rendering algorithms to the models. The method dynamically provides with multiple levels of detailed views of the molecular models. The main goal of the method is to maintain a certain level of frame rates so that the simulation results can be naturally viewed and manipulated[8,9]. In this case, however, we must support a mechanism for guaranteeing image quality which is sufficient for biologists. For instance, the geometrical features of the models must be maintained so that biologists can recognize the details of the critical parts of the molecular models.

Adaptive LOD[12,16] approach is take advantage the balance of rendering performance and quality of rendering images. It calculates best case in rendering fps per rendering objects. In molecular models, the number of polygon per atom is important factor. So we developed adaptive cost function for display molecular models.

---

[1] It is commonly regarded that any real-time rendering algorithm must provide more than 15 FPS (frames-per-second) [7].

3 Adaptive Level of Detail Rendering of Molecular Models

We propose a real-time adaptive LOD method for molecular modeling in this paper. The proposed method aims to simplify polygon meshes of molecular models while the geometrical details of their critical parts for molecular modeling are maintained. One of such critical parts is called as an Active Site, where two molecules can be combined and the energy level of the compound becomes stable. At the same time, we guarantee a certain level of frame rates by utilizing the adaptive LOD technique. Our proposed method can provide with a real-time interaction for a virtual environment of molecular modeling.

3.1 Process Overview

The adaptive LOD rendering consists of two steps. The first step is pre-processing as shown in Fig. 1-(a). It is for preparing three dimensional models of molecules for real-time rendering. The second step, real-time rendering as shown in Fig.1-(b), follows the first step. In the first step, three dimensional structures of molecules stored in a PDB (Protein Data Base) file are read. A set of polygon meshes is generated for each molecular model. Then, a molecular surface generation algorithm is applied [4]. The generated meshes are to be simplified using a simplification algorithm. The results are classified into a set of simplified models of LOD. The rates of simplification are 100%, 80%, 60%, 40%, and 20%, respectively. The LOD models are classified and stored into their corresponding classified LOD files (CLF) as shown in Fig. 1-(a).

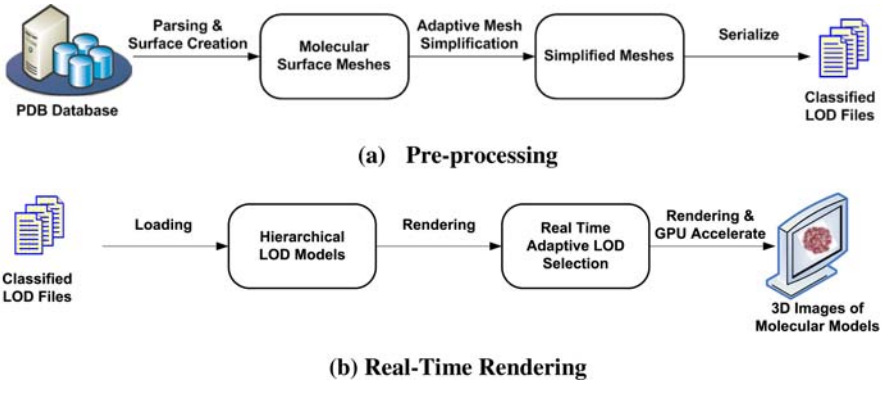


Fig. 1. System Overview

Adaptive real-time rendering is the main operation of the proposed method and utilizes the adaptive LOD technique. First, the proper complexities of molecular models for real-time rendering are determined depending on the specification of the system where the algorithm is implemented and executed. That is, the performance of the real-time rendering is calibrated for the system. The process reads classified LOD files and constructs hierarchical LOD models from the files. Therefore, properly simplified images of molecular models can be provided with a certain level of frame

rates. Also, the simplification algorithm is not applied to all parts of molecular models so that some critical parts of the models can be more precisely viewed. We propose evaluation functions to determine such critical parts. When the LOD value changes, the proper model is adaptively selected. So the natural transition of LOD models can be exercised as the distances between the models and the users vary.

3.2 Pre-processing

3.2.1 Mesh Simplification

We use an adaptive mesh simplification method and based on an edge contraction algorithm. The edge contraction algorithm is fast and provides with high quality images. As shown in Fig. 2, we utilized Garland and Heckert's algorithm [10].

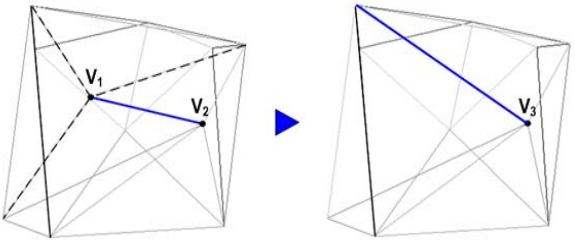


Fig. 2. An Example of Edge Construction

The algorithm works as follows: A list of vertices, which compose triangles, is constructed. Each element of the list corresponds to a vertex. It has information of segments, which are connected by the vertex, and triangles, which contain the vertex. Then, distances between the vertices and their adjacent vertices are calculated. The distances are used to find candidate vertices for mesh simplification. As for an evaluation function for mesh simplification, we define a quadratic function Q(v) as presented in the formula (1). A vector, v = [x, y, z, 1]T denotes a vertex, and a plane, where the vertex is included, is defined as ni = [a, b, c, d]t, ntix = 0.

$$Q(v) = \sum_{t=1}^{k} (n_i^t v)^2 \tag{1}$$

Suppose an edge between two vertices, v1 and v2, is reduced and a new vertex v3 is generated. The evaluation function of the new vertex can be defined as follows:

$$(\mathcal{V}_1, \mathcal{V}_2) \rightarrow \mathcal{V}_3, \tag{2}$$

$$Q(\mathcal{V}_3) = Q(\mathcal{V}_1) + Q(\mathcal{V}_2)$$

The new function value is utilized in ordering vertices which can be candidates for the mesh reduction. The values of the evaluation functions are compared. A pair of vertices, of which evaluation function is the lowest, is selected as a candidate for the edge contraction.

3.2.2 Critical Parts for Molecular Modeling

In molecular modeling, especially in docking simulation, detailed three dimensional shapes of critical parts such as active sites are very important. Since molecular models consist of iso-surfaces, the mesh simplification algorithm may change the shapes of the models and the shape of the critical parts may be transformed. Therefore, the mesh simplification process should maintain shapes of boundaries of active sites. We propose an algorithm to resolve this problem.

First, the algorithm recognizes boundaries of a critical part. We can find boundaries of the critical parts using Q(v) in the formula (1). The boundaries are detected when a line segment (v_1, v_2) belongs to a single triangle. Then, we protect the boundaries so that their shapes cannot be changed during mesh simplification. We build a virtual plane which perpendicularly meets the triangle of the boundary. A new vertex, v, is defined on the virtual plane. It is close to the line segment, (v_1, v_2). Therefore, the line segment remains after the mesh simplification so that the shapes of the boundaries are maintained.

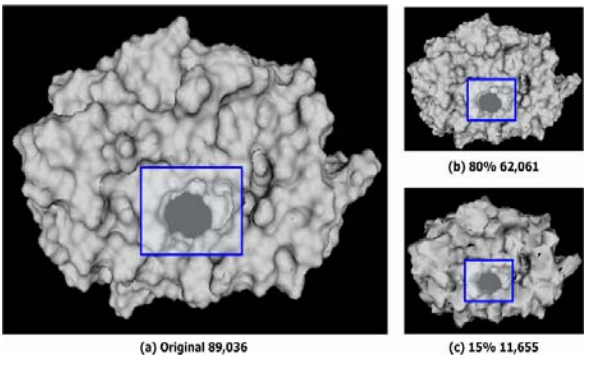


Fig. 3. Mesh simplification with maintaining geometrical features of critical parts for different LOD models: (a) The original model consists of 89,036 polygons, (b) The 80% simplified version has 62,061 polygons, (c) The 15% simplified version is composed of 11,655 polygons

The new triangle, n*, is defined as shown in the formula (3). It is based on the line segment, (v_1, v_2), and a normal vector, n.

$$n^* = \frac{n \times (v_2 - v_1)}{\left\| n \times (v_2 - v_1) \right\|} \tag{3}$$

The results are presented in Fig. 3. It shows that the critical parts, active site in this example, of simplified models can be viewed as those of precise models, even if other parts of models are simplified. The geometrical features of the active sites are maintained. The original shape of the active site in Fig. 3-(a) can be sustained and viewed, though other parts are simplified into 80% as shown in Fig. 3-(b) and 15% as shown in Fig. 3-(c).

3.3 Real-Time Rendering

3.3.1 Cost Function

First, the cost function of the real-time rendering is used to maintain the rendering speed more than a certain level. In general, real-time rendering of three dimensional computer graphics requires faster than 15 FPS. The cost function is used to determine whether the three dimensional geometrical models of molecules are to be simplified or expressed their precise geometrical information by displaying the more detailed shapes of polygon meshes. When the rendering speed is estimated to be faster than 15 FPS, the mesh simplification process is not required so that more detailed images can be displayed. Otherwise, the mesh reduction is applied.

Secondly, we use the cost function to maintain a certain level of image quality. The polygons of molecular structure correspond to atoms. Each atom has its own color determined by its atomic type. While the mesh simplification process progresses, the colors of the atom could be mixed or disappeared. Thus, the simplification process must maintain color information of each atom using the cost function. The cost function controls number of polygons to be rendered as presented in the formula (4).

$$P_{cnt} = \begin{cases} P_{cnt} - Z_{eye} \bullet \dfrac{P_{cnt}}{A_{cnt}}, \cdots\cdots\cdots if\ C_{fps} \le T_{fps} \\ P_{cnt} + Z_{eye} \bullet \dfrac{P_{cnt}}{A_{cnt}}, \cdots\cdots\cdots if\ C_{fps} > T_{fps} \end{cases} \tag{4}$$

If the current frame rate of display (C_{fps}) is slower than the target frame rate ($T_{fps,}$ Its default value is T_{fps} = 15fps), it reduces number of polygons (P_{cnt}) in order to improve the rendering speed. Since LOD is supported, P_{cnt} depends on the Z value of the position of a user (Z_{eye}) and the number of atoms in the model (A_{cnt}). On the other hand, the C_{fps} is could be faster than T_{fps}. In such a case, P_{cnt} should be increased in order to provide high quality images of molecular models. The proposed system can always provide with a fast rendering speed which is faster than the target frame rate, namely, 15 FPS. Also, it supports the adaptive level of image quality based on LOD values.

4 Experiments

The experiment was conducted on a 3.2 GHz Pentium 4 PC with 2GB RAM, nVdia Quadro 3400 graphic card. Our application was developed using OpenGL on Windows XP. Four very large molecular models selected from PDB Bank [15] for our experiments. Table 1 shows PDB codes, names, number of atoms and number of polygons of the molecular models.

Our first experiment is for measuring rendering performance. We tested rendering speeds of each model for 100 seconds using the FPS values. Each model has 5

Table 1. Selected molecular models

PDB CODE	Name	Atom Num	Polygons
1AOD	Bacillus	14,165	533,774
1CX2	Musculus	22,228	801,601
1YCE	Na+ ATPase	27,147	970,187
1MT5	Hydrolyase	64,155	2,656,246

different LOD models. They are the original model, the 80% simplified version, the 60% simplified version, the 40% simplified version, and the 20% simplified version. For each model, the rendering speeds of the 5 different LOD models were measured. Fig 4 shows the FPS values of the models. The rendering speeds progressively increases as the level of simplification increases. For an instance, the rendering speed of the original model of Hydrolyase (represented as D in the graph) is around 17 FPS. The speed increases to 20 FPS with the 80% model, 23 FPS with the 60% model, 27 FPS with the 40% model, and 40 FPS with the 20% model.

Secondly, we compared average FPS values of our system (VRMMS) with the VMD, which is the most popular in this field [6]. We used the 60% simplified model. Table 2 shows the FPS values of VMD and VRMMS. The FPS values of VMD suggest that large molecular models, of which numbers of polygons are more than 800,000, cannot be smoothly rendered and manipulated in VMD. On the other hand, VRMMS could show faster frame rates even though large models are rendered.

The rendered images with different LOD models of the selected molecular models are shown in Fig 5, 6, 7, and 8, respectively.

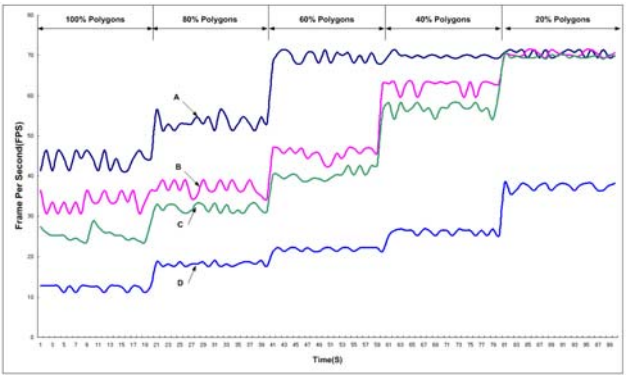


Fig. 4. Rendering speeds (in FPS values) of three dimensional interactions using molecular models, when the proposed algorithm is applied. A=Bacillus, B=Musculus, C=Na+ATPase, D=Hydrolyase.

Table 2. Comparing the average FPS values of real-time rendering the three dimensional molecular models using VMD and VRMMS

PDB CODE	VMD	VRMMS
Bacillus	12.8	55.64
Musculus	7.5	38.05
Na+ ATPase	6.4	31.94
Hydrolyase	2.6	23.21

(FPS)

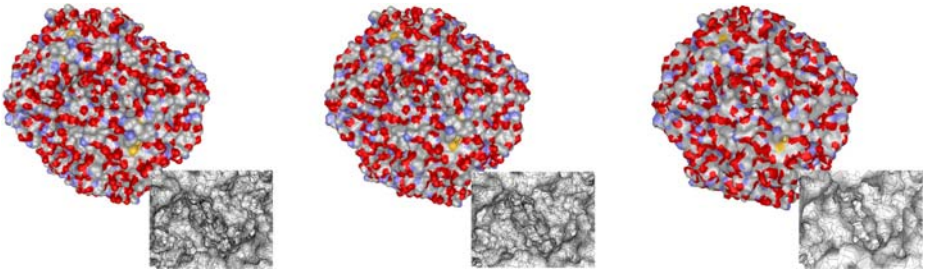


Fig. 5. Bacillus: The 100% version with 533,774 polygons, the 60% simplified version with 314,696 polygons, the 20% version with 100,104 polygons

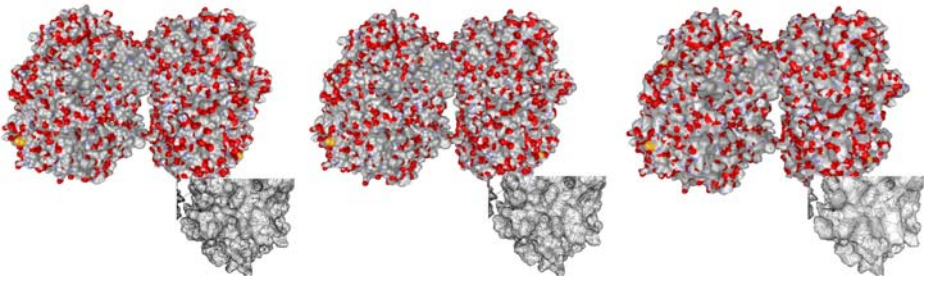


Fig. 6. Musculus: The 100% version with 801,601 polygons, the 60% simplified version with 473,407 polygons, the 20% version with 149,711 polygons

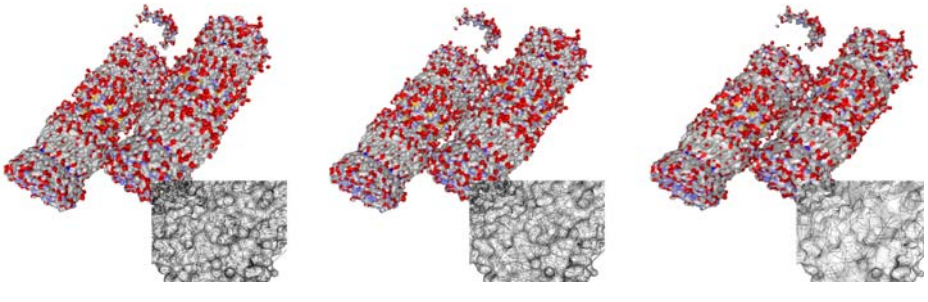


Fig. 7. Na+ATPase: The 100% version with 970,187 polygons, the 60% simplified version with 571,680 polygons, the 20% version with 181,978 polygons

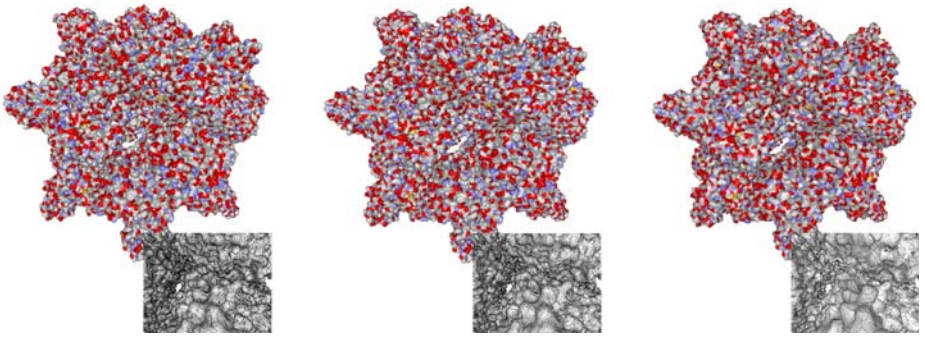


Fig. 8. Hydrolyase : The 100% version with 2,656,246 polygons, the 60% simplified version with 1,565,592 polygons, the 20% version with 496,340 polygons

5 Concluding Remarks

In this paper, we propose an algorithm to enhance rendering speed using adaptive LOD rendering. While an ordinary LOD method cannot maintain detailed geometrical information of critical parts of three dimensional models, the proposed algorithm recognizes the critical parts and prevents the polygon meshes of the parts from being simplified during the LOD procedure. Therefore, the shapes of the critical parts can be maintained.

The performance evaluation of the proposed algorithm showed that the system can be operated with better performance comparing with the other algorithms. We measured the FPS values of rendering three dimensional large-scale molecular models as we applied the LOD technique. Then, we compared our system with VMD, which is the most popular. The experiments were exercised using a PC class computer system with a general graphics card. The results demonstrated that the system can be interactively used with high quality images.

The proposed system can be used in the docking simulations for large-scale molecular models which have been processed by a high performance computing system. It can also be applied to increase productivity of users using three dimensional graphics and virtual reality in the field of a new drug design.

Acknowledgements

This work was supported by the Ministry of Information & Communications, Korea, under the Information Technology Research Center(ITRC) Support Program.

References

1. F.M Richards, “Areas, volumes, packing and protein structures,” in Annual Review of Biophysics and Bioengineering, 6, pp. 151-176, 1977.
2. Jee-In Kim, Sungjun Park, Jun Lee, Youngjin Choi and Seunho Jung, “Development of a Gesture-Based Molecular Visualization Tool Based on Virtual Reality for Molecular Docking”, Bulletin of the Korean Chemical Society, Vol.25, No.10, pp 1571-1574, 2004.

3. Sungjun Park, Jun Lee and Jee-In Kim, "A Molecular Modeling System Based on Dynamic Gestures", LNCS 3480, pp.886–895, 2005.
4. Amitabh Varshney, Frederick P. Brooks and Jr. William V. Wright, "Linearly Scable Computation of Smooth Molecular Surfaces", IEEE Computer Graphics and Applications, Vol. 14, No.5, pp 19 – 25 Sept 1994.
5. C.H. Lee and A. Varshney, "Representing Thermal Vibrations and Uncertainty in Molecular Surfaces", SPIE Conference on Visualization and Data Analysis 2002, San Jose, CA, Jan, 2002.
6. VMD, http://www.ks.uiuc.edu/Research/vmd/
7. Tomas Akenine-Moller and Eric Haines, REAL-TIME RENDERING, The 2nd Edition, AK Perters.
8. Stefan Birmanns, Maik Bolyes, Hwrwig Zilken and Willy Wriggers, "Adaptive Visuo-Haptic Rendering for Hybrid Modeling of Macromolecular Assemblies", In: Proceedings Mechatronics and Robotics, Vol.4, pp. 1351-1356, 2004.
9. Chandrajit Bajaj, Peter Djeu, Vinay Siddavanahalli and Anthony Thane, "Interactive visual exploration of large flexible multi-component molecular complex", Proc. of the Annual IEEE Visualization Conference, pp. 243-250, 2004.
10. Michel Garland and Paul S.Heckbert, "Surface Simplification using quadric error metrices", In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp 209-216, ACM Press/Addision-Wesly Publishing Co. 1997.
11. J. Xia and A. Varshney, "Dynamic View-Dependent Simplification for Polygonal Models", IEEE Visualization'96 pp327-334, 1996
12. David P.Luebke, and Benjamin Hallen, "Perceptually Driven Simplification for Interactive Rendering", Proceedings of the 12th Eurographics Workshop on Rendering Techniques, pp. 223-234, 2001
13. H. Hoppe, "View-Dependent Refinement of Progressive Meshes", ACM Siggraph97, pp189-198, 1997
14. Polygonal Models, Paolo Cignoni, Fabio Ganovelli, Enrico Gobbetti, Fabio Marton, Federico Ponchio, and Reberto Scopigno, "Adaptive TetraPuzzles : Efficient Out-of-Core Construction and Visualization of Gigantic Multiresolution**",** ACM Transaction on Graphics 23.3 pp796-803, 2004
15. PDB Bank, http://www.rcsb.org/
16. Thomas A. Funkhouser, Carlo H. Séquin, "Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments", Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp 247 – 254, 1993

An Efficient Algorithm for Connected Attribute Thinnings and Thickenings

David Lesage[1,4], Jérôme Darbon[2,5], and Ceyhun Burak Akgül[1,3]

[1] École Nationale Supérieure des Télécommunications (ENST)
46 rue Barrault, F-75013 Paris, France
[2] EPITA Research and Development Laboratory (LRDE),
14-16 rue Voltaire F-94276 Le Kremlin-Bicêtre, France
[3] Electrical and Electronics Engineering Department, Bogazici University
34342 Bebek, Istanbul, Turkey
[4] in collaboration with Siemens Corporate Research
755 College Road East, Princeton NJ 08540, USA
[5] UCLA Mathematics Department, Box 901555, Los Angeles, USA
{lesage, akgul}@enst.fr, jerome@math.ucla.edu

Abstract. Connected attribute filters are morphological operators widely used for their ability of simplifying the image without moving its contours. In this paper, we present a fast, versatile and easy-to-implement algorithm for grayscale connected attribute thinnings and thickennings, a subclass of connected filters for the wide range of non-increasing attributes. We show that our algorithm consumes less memory and is computationally more efficient than other available methods on natural images, for strictly identical results.

1 Introduction

Connected attribute filters are morphological operators enjoying the property of simplifying an image while preserving the contour information, as they work on the connected components of the image level-sets [9,15,20]. Such filters are widely used in practice for image classification [1], filtering [18], segmentation [5,21], feature analysis [23] and shape filtering [26].

Connected attribute filters are first classified according to their theoretical properties. Filters considering image upper level-sets, such as attribute openings and thinnings, are anti-extensive [18]. Intuitively, they are well suited for applications where objects are brighter than their background, as higher level components are seen as included in lower level ones. The dual vision takes lower level-sets into account and leads to extensive filters such as attribute closings and thickenings. A second distinction among connected filters is made on the type of attribute used. Attribute openings and closings correspond to the case where the attribute associated with a level set is a non-decreasing function w.r.t. set inclusion. A popular example can be found in area opening and closing [24,20]. A larger class consists of filters using non-increasing criteria, namely attribute thinnings and thickenings. Numerous practically interesting criteria fall in that

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 393–404, 2006.
© Springer-Verlag Berlin Heidelberg 2006

category, such as elongation criterion from [26], complexity/simplicity, motion estimation and entropy factors from [18].

Efficient algorithms have been described in [6,14,18,24] for connected attribute openings and closings. To our knowledge, few algorithms have been proposed for attribute thinnings and thickenings [3,18]. In this paper, we present a new algorithm for such filters. It requires *less* memory and exhibits *better* runtime performance compared to other available methods, while yielding strictly identical results.

Defining connected attribute filters for grayscale images with non-increasing criteria is not a straightforward task. Indeed, the threshold decomposition property [8,12] only holds for increasing criteria, i.e. for attribute openings and closings, allowing for a direction definition using lower- or upper-sets of the image. For thinnings and thickenings, various definitions become suitable. Salembier's *Min/Max-Trees* offer convenient representations to express them [18].

A Max-Tree is an inclusion tree that considers connected components of upper-level sets of an image. A node of the Max-Tree corresponds to a connected component C_0 of an upper-level set. Its parent is defined as the connected component C_1 of the higher upper-level set such that $C_0 \neq C_1$ and $C_0 \subset C_1$. The leaves of a Max-Tree correspond to the regional maxima of the image. An illustration is given in Figure 1-a). The dual representation by a Min-Tree considers lower-level sets instead of upper-level sets. As reported in [18], connected attribute filters for grayscale images can be defined as a *pruning* of its Min- or Max-Tree, in such a way that the threshold decomposition property [8,12] holds. This enables the straightforward reconstruction of the filtered image from the pruned tree. Given a non-increasing criterion, many different pruning rules can be considered, each leading to a different connected-set filter [18,26].

Following strictly this definition, a possible approach to implement a connected attribute filter consists in first building the Min/Max-Tree and then applying a pruning process. The original Min/Max-Tree construction is presented by Salembier *et al.* in [18]. It relies on a recursive flooding procedure using a hierarchical queue data structure. An alternative scheme for Min/Max-Tree construction based on Tarjan's union-find algorithm has been originally proposed by Hesselink in [11] and by Najman *et al.* in [16].

Direct implementations which by-pass tree construction are also possible. The latter require to choose the pruning rule before performing the filtering process. A popular choice for the pruning, known as the *Max rule* [18], consists in removing a node if its associated criteria do not satisfy a prescribed threshold and if all its descendants were removed. In [3], Breen *et al.* propose a queue-based algorithm to perform direct filtering using the *Max rule*. In [25], Wilkinson *et al.* present an union-find based algorithm for performing connected attribute openings and closings. It should be noted that the same algorithm also performs a *Max rule*-pruning for non-increasing criteria.

In this paper, we follow the approach of Darbon and Akgül [6] for connected attribute closings and openings and extend it to non-increasing criteria. The main contributions of this paper are the following. We propose a fast and

memory wise algorithm to perform the filtering for the main pruning rules proposed in the literature. We compare our algorithm with other available methods and show that our algorithm is *more efficient* in terms of both memory requirements and computational load. In Section 2, we present our algorithm along with its variations to cope with different pruning rules. Section 3 is devoted to the presentation of numerical experiments and comparisons. Finally in Section 4, we draw some conclusions and sketch future prospects.

2 Algorithm

The proposed algorithm relies on a tree pruning scheme, as in Salembier *et al.*'s original work for the Max-Tree construction [18]. A *flooding* process builds the inclusion tree of connected components and a successive *resolution* stage applies a pruning strategy to recover filtered pixel values. Our work also benefits from optimizations introduced in [6], by avoiding the external hierarchical queue and the sorting procedure used in [18]. We emphasize our algorithm is straightforward to implement given the pseudo-code of this Section and is versatile enough to be suitable for numerous filtering tasks without modification.

In the following, we limit ourselves to attributes that can be computed recursively during the flooding pass, by adding new points or by merging components. For the sake of simplicity, our algorithm is given for anti-extensive, attribute thinning filters. The dual representation for attribute thickenings only requires few tests to be inverted in the flooding procedure. We denote by N the size of the image, L the number of gray levels and A the number of auxiliary values needed to compute the attribute. We first describe the flooding process which builds the tree and then the resolution pass which finalizes the filtering.

2.1 Flooding

Our flooding step is directly inspired by Salembier's Max-Tree one and is generic for any resolution strategy. Following ideas introduced in [6], we use an array `status` to store pointers to points encoded as positive integers (lexicographic rank). In our algorithm, `status` serves two purposes. It first stores the hierarchical queue and eventually turns into the Max-Tree itself, encoding child-to-parent links. It is initialized to a special value `NONE` which signifies that no link was set and that therefore points have not been processed.

We use three arrays of length L, namely `last`, `representative` and `attribute`. The value `last[h]` stores the last point introduced in the `h`-level queue. The value `representative[h]` stores the first encountered point for the current `h`-level component C_h^k, which serves as an unique representative of C_h^k. The value `attribute[h]` holds the A values needed to compute C_h^k attribute.

Since we seek after a generic thinning procedure, we build the tree explicitly and keep its structure intact until resolution. We cannot afford altering it as done in [6]. To keep time and memory requirements as low as possible, we instead use a boolean array `tag` which points out components not satisfying the thinning criterion λ.

Our flooding process is detailed in Algorithm 1. At level h, the procedure `flood(h)` iterates the queue at the same level, using `last[h]` (line 2), to retrieve current component C_h^k. Point `p` is dequeued (lines 3-4) and is set to point to its representative `r` (line 6). Non-representative points are tagged so they always take their representative's value, without loss of generality w.r.t. the resolution rule. Then non-processed neighbors `q` of `p` are visited (line 7). They are pushed in the queue (lines 10-11) and registered as representative if we reached a new level component (line 14). If `q`'s gray-level `i` is higher than `h`, the flooding procedure is relaunched recursively to extract the subtree (lines 17-19). Back from the neighbors, the attribute of the current `h`-level component is updated (line 21).

Once the propagation at level `h` is over, C_h^k is completely retrieved and its final attribute value can be tested (line 21). If it does not fulfill λ, C_h^k representative is tagged. Parent relationships are then set in a similar way to [18] and [6] (lines 23-29). Auxiliary arrays at level `h` are finally reset (lines 31-32).

The flooding is initially launched at the lowest level in the image. Eventually, the entire tree structure is encoded in `status`. By construction, non-representative points are directly linked to their representative and representatives point to a lower level representative until root is reached. Additionally, we can directly identify a component that does not fulfill the attribute criterion by checking its `tag`.

2.2 Resolution

Once the tree is built and tagged, a resolution phase eventually retrieves filtered values and stores them in the output image `out`. Depending on the type of non-increasing attribute [18] and the application, several resolution schemes have been proposed in the literature. We implemented classical rules such as *Direct*, *Min* and *Max* rules [10,19,20] as well as the *Subtractive* rule from [22]. Due to space restrictions, we only present Min and Max rules. Others can be easily deduced from the mechanisms presented here.

In contrast to Salembier's original Max-Tree approach [18], we do not dispose of a sorted structure allowing tree traversals from parents to children. Our algorithm remains as efficient as possible by altering the tree to avoid redundant operations.

Min Rule. A node is removed if it does not satisfy λ or if an ancestor was removed. A loop on all image points launches the recursive procedure of Algorithm 2. The tree is progressively encoded with the final filtered values as negative integers (`-val-1`) similarly to [6]. A point is set to its parent's value if it is tagged or if its parent's value changed. An illustration of this strategy is given in Figure 1-b).

Max Rule. A node is removed if it does not satisfy λ and if all of its children are removed. This rule requires two passes since the tree is encoded with only parent relationships. The first loop launches Algorithm 3 procedure on all image points. It uses a boolean `child_kept` to propagate the information that a child node has kept its value. If it can be resolved during the first pass, the final filtered value

is encoded in `status[p]` as a negative integer. If not, the point does not fulfill λ and has all of its children removed. Thus, a second pass (Algorithm 4) simply tracks down non-negative `status` values to set them to their parent's value. An illustration of this strategy is given in Figure 1-c).

Algorithm 1. flood(level h)

```
1: r ← representative[h]
2: while last[h] ≠ NONE do // propagation at level h
3:    p ← last[h] // pop p from the queue
4:    last[h] ← status[p]
5:    if p ≠ r then // set to the representative
6:       status[p] ← r; tag[p] = true
7:    for all neighbors q of p with status[q] = NONE do
8:       i ← ima[q]
9:       status[q] = last[i]; last[i] = q // push q in the queue
10:      if representative[i] = NONE then
11:         representative[i] ← q
12:      m ← i
13:      if i > h then // flood higher level components
14:         repeat
15:            m ← flood(m)
16:         until m > h
17:   attribute[h].update(p) // update the attribute with p
18: if attribute[h].val < λ then // tag as not satisfying λ
19:   tag[r] = true
20: // set parentship
21: m ← h-1
22: while m > 0 and representative[h] = NONE do
23:   m ← m-1
24: if m > 0 then
25:   attribute[m].merge(attribute[h]) // merge attributes
26:   status[r] ← representative[m] // link representatives
27: // reset level h auxiliary values
28: attribute[h].reinit(); last[h] ← NONE;
29: representative[h] ← NONE
30: return m
```

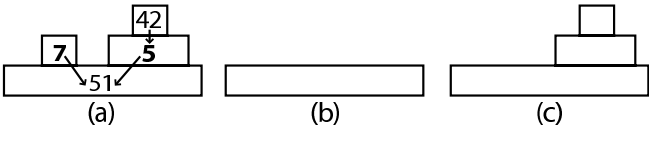


Fig. 1. Max-Tree and attribute values of a 1D signal (a), Min-rule (b) and Max-rule thinning (c) for $\lambda = 10$

Algorithm 2. resolve_min_rec(point p)

```
if status[p] ≥ 0 then // not processed
  parent_val ← resolve_min_rec(status[p])
  if tag[p] = true or parent_val ≠ ima[status[p]] then
    status[p] ← -parent_val-1; out[p] ← parent_val
  else // p keeps its value
    status[p] ← -ima[p]-1; out[p] ← ima[p]
return out[p]
```

Algorithm 3. resolve_max_rec1(point p, boolean child_kept)

```
if status[p] ≥ 0 then // not processed
  if tag[p] = false or child_kept then // p keeps its value
    status[p] ← -ima[p]-1; out[p] ← ima[p]
    resolve_max_rec1(status[p], true) // propagate info
return
```

Algorithm 4. resolve_max_rec2(point p)

```
if status[p] ≥ 0 then // not processed, assign parent's value
  out[p] ← resolve_max_rec2(status[p])
  status[p] ← -out[p]-1
return out[p]
```

3 Analysis and Experiments

In this Section, we compare our approach to Salembier *et al.*'s Max-Tree algorithm [18] and Wilkinson *et al.*'s method based on Tarjan's union-find method for attribute openings [25]. The latter is not a thinning algorithm, but is strictly equivalent to a thinning with a Max rule for non-increasing attributes. We voluntarily omit priority queue-based algorithms [24,3] which exhibit a high dependance w.r.t the threshold value λ [14]. Experiments show that our algorithm is faster for natural images and requires less memory than other other available algorithms. We first compare memory requirements and then present time results along with theoretical time complexity.

3.1 Memory Requirements

Since $N \gg L$ in general, we neglect memory used by arrays of size L for all three algorithms. Salembier *et al.*'s Max-Tree algorithm is the most demanding one in terms of memory. It uses a queue of N integers and a tree of size N. Each tree node contains at least a pointer to the parent, the final attribute value and the filtered gray level. Min and Substractive rules also require the original gray level to be stored, totalizing $5N$ values plus a mandatory output image.

Wilkinson *et al.*'s union-find algorithm uses two arrays, `PixelSort` and `Parent`, of N integers each. The array `Parent` eventually stores the output image. Unlike both other algorithms, attributes cannot be computed in an array of size L. It

requires an array `auxdata` of N pointers and a dynamic allocation/deallocation scheme to limit the memory consumption to $N + (A * N)/2$ values in the worst case. This totals to at least $3N$, at worst $3N + (A * N)/2$ [14].

Naïve implementation of our approach requires an integer array `status`, a boolean array `tag` and an output image of size N. Similarly to Meijster *et al.*'s method [14], a straightforward modification allows for encoding filtered values in `status`. The array `tag` can also be avoided by using a bit of each `status[p]` value with bit-masking operations, for an unsignificant computational cost. Memory requirements can thus easily be lowered to only N integers.

Memory load can become critical for applications such as medical imaging. Dealing with volumes of nearly $N = 512^3$ voxels, each integer array represents 512MB in memory. Our algorithm allows for processing such volumes with reasonable requirements using computers of the present era.

3.2 Computational Load

From a theoretical point of view, the computational cost of both Salembier *et al.*'s Max-Tree and our algorithm is dominated by the linear flooding. The search of parent nodes is linear w.r.t to L and the resolution is also linear w.r.t N, leading to a complexity of $\Theta(N \cdot L)$. The complexity of Wilkinson *et al.*'s union-find method is $\Theta(N \cdot \log(N))$ as shown in [14]. Salembier *et al.*'s Max-Tree and our approach are unattractive in theory because their complexity is pseudo-polynomial [2]. For their complexity to be polynomial, it should be polynomial w.r.t $\log_2(L)$ since it requires only $\lceil \log_2(L) \rceil$ bits to encode the integer L. They can be made polynomial by using a binary heap, but it seems to be practically uninteresting [14]. Even if pseudo-polynomial algorithms are theoretically worse than polynomial ones, they may be more attractive in practice. A well known example is Dial's pseudo-polynomial algorithm [7] for solving the shortest path problem. For many practical problems it outperforms other polynomial algorithms such as Dijkstra's one, as reported in [2,7].

Algorithms were tested on a variety of natural images, using various criteria. Due to space restriction, we only present performance results obtained on four images, the well-known *Lena*, a highly textured image of a *carpet*, a large *satellite* image with few gray levels and a *cardiac Computed Tomography* (CT) medical volume. Original images are given in Figure 2. Their properties are summed up in Table 1 where *Complexity* stands for the ratio between the number of iso-level connected components and image size. Note that the medical volume was cropped (the original counts 400 slices) in order to allow Salembier *et al.*'s Max-Tree and Wilkinson *et al.*'s union-find algorithms to run properly under our memory constraints (1GB RAM).

Tests presented in Figure 3 were performed using the non-increasing, 2D and 3D *elongation* criteria proposed in [26]. Resolutions for Salembier *et al.*'s Max-Tree and our algorithm used Max rule to be strictly equivalent to Wilkinson *et al.*'s union-find method. This way, results obtained by either of these schemes are identical, as they match the exact same theoretical definition and only differ by the mechanisms involved. Note that applying such a criterion and such a

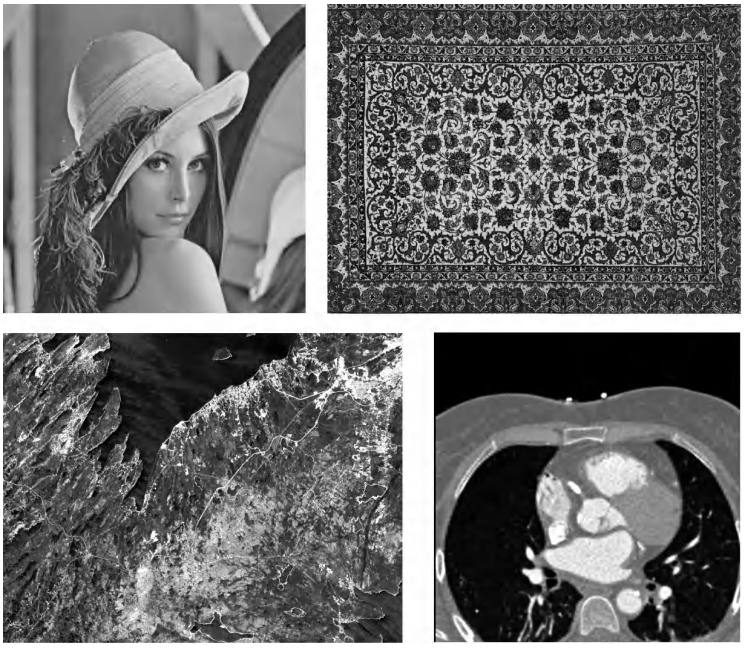


Fig. 2. Test images. From left to right, top to bottom: Lena, carpet, satellite and cardiac CT (2D slice detail).

Table 1. Test images

Image	Size	Levels	Complexity
lena	512×512	216	0.38
carpet	1024×715	255	0.84
satellite	1380×1780	30	0.24
cardiac CT	$512 \times 512 \times 100$	5000	0.11

resolution rule may not be meaningful for all the images presented here. These tests are presented for the sake of fair performance comparison only. However, we emphasize they are representative of the behaviors observed for different criteria and different resolution rules. A result sample is depicted in Figure 4, where the elongation criterion is applied on the carpet image. Another result is given in Figure 5, with the same criterion and on the satellite image. Such a filtering could be used to help further extraction of road networks. Please refer to the works cited in introduction for additional illustrations of the numerous applicative possibilities offered by connected attribute filters.

All three algorithms have been implemented in C++ and compiled with full optimization. Timings were obtained by averaging 100 runs on a Pentium 4 2.4GHz (1MB cache and 1GB RAM). As depicted in Figure 3, our method outperforms other algorithms for natural images. Execution time of Salembier *et al.*'s Max-Tree and our algorithm is independent of λ. A slight dependence to λ is

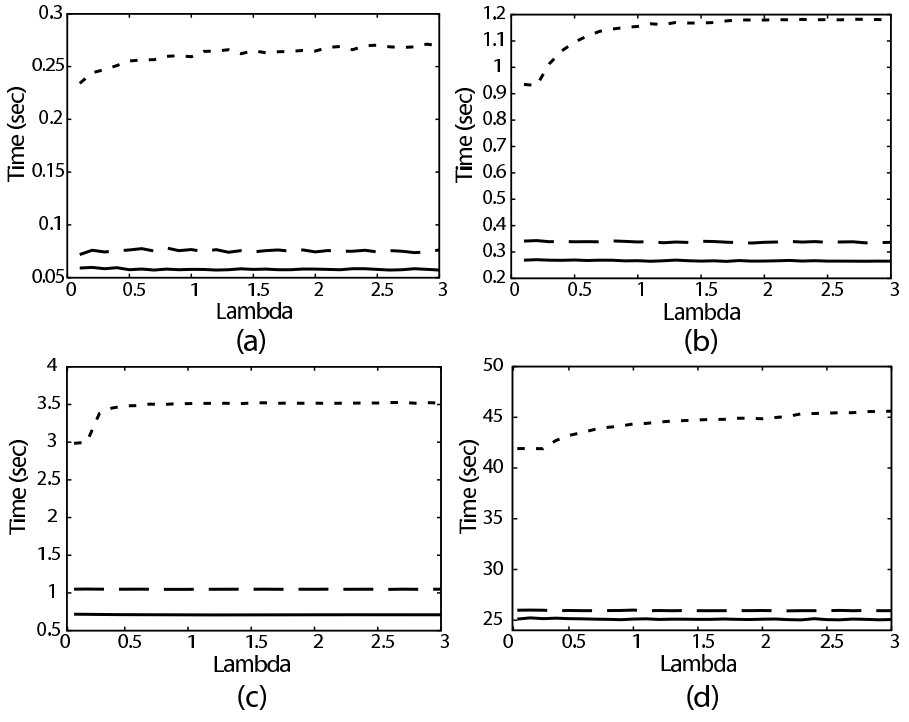


Fig. 3. Timing results for our algorithm (solid), Max-Tree (dashed) and Union-Find attribute opening (dotted) in function of λ. Elongation attribute thinning [26] on (a) lena, (b) carpet, (c) satellite and (d) cardiac CT.

observed for the Wilkinson *et al.*'s union-find approach, confirming conclusions of [14].

Since it is based on a similar core flooding scheme, our algorithm exhibits a limited performance gain (5 to 30%) compared to Salembier *et al.*'s Max-Tree algorithm, mainly due to a faster tree construction (no pixel sorting, no external queue and less memory used). Our resolution step is slower than Salembier *et al.*'s Max-Tree's one because of our non-ordered tree traversal. It seems that the bigger the image, the less our advantage on the original algorithm becomes significant, as can be seen for the medical volume. However, memory consumption for Salembier *et al.*'s Max-Tree algorithm can meanwhile become prohibitive for such applications. We emphasize the effect of memory load and 'cache-friendliness' on performance. The less memory an algorithm needs and the more spatially coherent memory accesses are, the less often cache faults will occur. In this regard, our algorithm combines all these advantages.

Wilkinson *et al.*'s union-find approach, on the other hand, seems to suffer from a construction process more prone to important memory jumps. It may also be penalized by its dynamic allocation/deallocation scheme. The medical volume example suggests that the performance gap narrows when the number

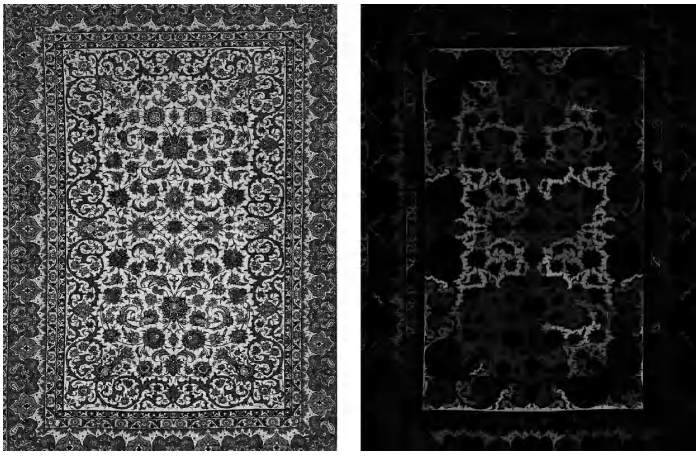


Fig. 4. Elongation attribute thinning [26] ($\lambda = 1$) on the carpet image. Left: original image. Right: filtered result.

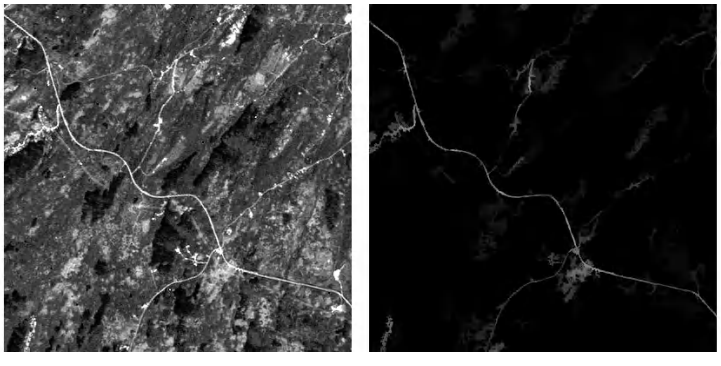


Fig. 5. Elongation attribute thinning [26] ($\lambda = 1$) on the satellite image (detail). Left: original image. Right: filtered result.

of gray levels increases, supposedly because of the dependance on L affecting both other algorithms [14,17]. To draw a definitive conclusion, one should adapt Wilkinson *et al.*'s union-find attribute opening to thinning operations. To our knowledge, this would unfortunately require additional equipment, following for instance ideas from [16]. Finally, the union-find approach holds a clear advantage in terms of parallelization possibilities [14]. This may turn it into a candidate of choice for highly-parallelized environments.

4 Conclusion

In this paper, we have presented a new algorithm for attribute thinnings and thickenings. It is easy to implement and enjoys the versatility and generality

of Salembier's original algorithm. Experiments on natural images show that our algorithm exhibits *better* performance than others while consuming considerably *less* memory. In a forthcoming paper, we will present new algorithms for efficient interactive filtering and visualization, as originally proposed in [13] with the Max-Tree algorithm. Extension of this approach to granulometries [14,22] and grain filters [4] will also be described.

Acknowledgements

The authors thank Isabelle Bloch and Elsa Angelini from ENST Paris, Gareth Funka-Lea from Siemens Corporate Research and Bülent Sankur from Bogazici University for their valuable suggestions. The cardiac CT volume is courtesy of Siemens Medical Solutions.

References

1. S.T. Acton and D.P. Mukherjee. Scale space classification using area morphology. *IEEE Transactions on Image Processing*, 9(4):623–635, April 2000.
2. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows.* Prentice Hall, 1993.
3. E.J. Breen and R. Jones. Attribute openings, thinnings and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, 1996.
4. V. Caselles and P. Monasse. Grain filters. *J. of Math. Imaging and Vision*, 17(3):249–270, 2002.
5. J. Crespo, R. Schafer, J. Serra, C. Gratin, and F. Meyer. The flat zone approach: A general low-level region merging segmentation method. *Signal Processing*, 62(1):37–60, 1998.
6. J. Darbon and C.B. Akgül. An efficient algorithm for attribute openings and closings. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*, Electronic proceedings, 2005.
7. R. Dial, F. Glover, D. Karney, and D. Klingman. A computational analysis of alternative algorithms and labeling techniques for finding shortest path trees. *Networks*, 9:215–248, 1979.
8. F. Guichard and J.M. Morel. Mathematical morphology "almost everywhere". In *Proceedings of ISMM*, pages 293–303. Csiro Publishing, April 2002.
9. H. Heijmans. Connected morphological operators for binary images. *Computer Vision and Image Understanding*, 73(1):99–120, 1999.
10. Henk J. A. M Heijmans. Theoretical aspects of gray-level morphology. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):568–582, 1991.
11. W.H. Hesselink. Salembier's min-tree algorithm turned into breadth first search. *Information Processing Letters*, 88(5):225–229, December 2003.
12. P. Maragos and R.D. Ziff. Threshold superposition in morphological image analysis systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(5):498–504, 1990.
13. A. Meijster, M.A. Westenberg, and M.H.F. Wilkinson. Interactive shape preserving filtering and visualization of volumetric data. In *Proc. of the Conf Comp. Signal Image Proc. (SIP)*, pages 640–643, Kauai, Hawaii, USA, August 2002.
14. A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. on PAMI*, 24(4):484–494, April 2002.

15. F. Meyer. Levelings, image simplification filters for segmentation. *Journal of Mathematical Imaging and Vision*, 20:59–72, 2004.
16. L. Najman and M. Couprie. Quasi-linear algorithm for the component tree. In *SPIE Symposium on Electronic Imaging*, pages 18–22, 2004.
17. L. Najman and M. Couprie. Building the component tree in quasi-linear time. *Accepted to IEEE Transactions on Image Processing*, 2006.
18. P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Trans. on Image Processing*, 7(4):555–570, 1998.
19. P. Salembier and J. Serra. Flat zones filtering, connected operators and filters by reconstruction. *IEEE Transactions on Image Processing*, 4:1153–1160, 1995.
20. J. Serra and P. Salembier. Connected operators and pyramids. In *Proc. SPIE Image Algebra Math. Morphology*, volume 2030, pages 65–76, 1993.
21. P. Soille. *Morphological Image Analysis Principles and Applications*. Springer-Verlag, 1999.
22. E. R. Urbach and M. H. F.Wilkinson. Shape-only granulometries and grey-scale shape filters. In *Proceedings of the International on Mathematical Morphology*, pages 305–314, 2002.
23. C. Vachier. Morphological scale-space analysis and feature extraction. In *Proceedings of the ICIP'01*, pages 676–679, October 2001.
24. L. Vincent. Grayscale area openings and closings, their efficient implementation and applications. In *Proc. EURASIP Mathematical Morphology and Its Application to Signal Processing*, pages 22–27, 1993.
25. M.H.F. Wilkinson and J.B.T.M. Roerdink. Fast morphological attribute operations using tarjan's union-find algorithm. In *Proceedings of the ISMM*, pages 311–320, Palo Alto, CA, June 2000.
26. M.H.F. Wilkinson and M.A. Westenberg. Shape preserving filament enhancement filtering. In *Proc. of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 770–777, 2001.

Creating Multi-layered 3D Images Using Reversible Jump MCMC Algorithms

Sergio Hernandez-Marin[1], Andrew M. Wallace[1], and Gavin J. Gibson[2]

[1] Joint Research Institute of Signal and Image Processing, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK
[2] Dept. of Actuarial Mathematics and Statistics, School of Mathematical and Computer Sciences. Heriot-Watt University, Edinburgh EH14 4AS, UK
{snh3, a.m.wallace, g.j.gibson}@hw.ac.uk

Abstract. Standard 3D ranging and imaging systems process only a single return from an assumed single opaque surface. However, there are situations when the laser return consists of multiple peaks due to the footprint of the beam impinging on a target with surfaces distributed in depth or with semi-transparent surfaces. If all these returns are processed, a more informative multi-layered 3D image is created. We propose a unified theory of pixel processing for ladar data using a Bayesian approach that incorporates spatial constraints through a Markov Random Field. The different parameters of the several returns are estimated using reversible jump Markov chain Monte Carlo (RJMCMC) techniques in combination with an adaptive strategy of delayed rejection to improve the estimates of the parameters.

Keywords: 3D ranging and imaging, reversible jump Markov chain Monte Carlo, Markov random fields, multi-layered image, time-correlated single photon counting, burst illumination laser.

1 Introduction

The goal of our work is a complete characterisation of the 3D surfaces viewed by a laser 3D imaging system. In general this includes measuring multiple returns viewed by a single pixel. There are many reasons why this occurs, for example the field of view may cover more than one surface due to divergence of the optical system, we may image through transparent objects [1], or objects (e.g. branches) may move in or out of the field of view during a finite collection time (e.g. when imaging a vehicle behind a tree on a windy day).

1.1 Background

We employ two principal approaches to range measurement and depth imaging, by Time-Correlated Single Photon Counting (TCSPC) [2] and by range gating, or repeated Burst Illumination Laser (BIL) [3]. With reference to Fig. 1(a), a pulsed laser is directed towards the target; the time of flight (TOF) of the laser to and from the target gives a measurement of distance since the speed of light in the

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 405–416, 2006.
© Springer-Verlag Berlin Heidelberg 2006

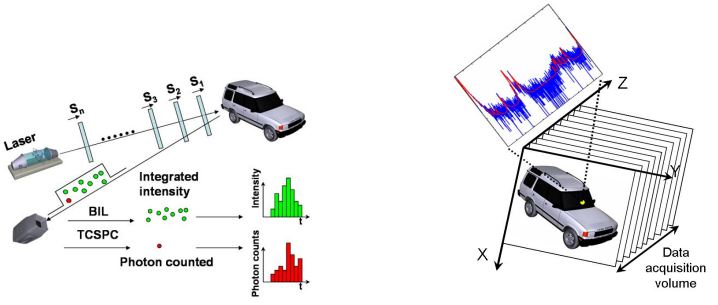
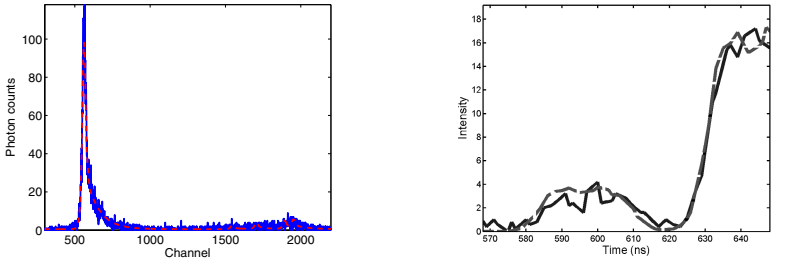


Fig. 1. (a) TOF principle (b) Spatial structure of an object

medium is known. In the TCSPC method, the detector records the arrival of the first photon; repeated laser pulses result in a histogram of photon arrival times (see Fig. 1(b)) that can be processed to determine the target range and signature. In the BIL method the detector integrates the received laser light over a time period (typically a few ns). Thus, the pulsed laser-CCD detector pair records the laser return from a distant slice of space. The range data is created from the multiple flashes by variable range gating. A histogram of integrated intensities is created for each pixel with the time axis determined by the range gate delay (see Fig. 2(b)). Thus, we can consider the observed photon (resp. intensity) histogram as a sample of a non-normalized statistical mixture distribution. To create a full 3D image in the TCSPC method, the laser is normally scanned. In a BIL system, the CCD records a full 2D slice in 3D space at each range gate so the 3D image is built from the repeated laser pulses as the range gate changes. The relative merits of each approach are discussed in [2,3] and the references therein.

Classical processing methods of 3D ranging and imaging only consider a single return. In general, this single return corresponds to either the first pulse or the last pulse received. The peak detector is a simple threshold comparator or a matched filter. The system used therefore requires a reasonably high signal-to noise ratio to be able to discriminate the return from the background noise. In practice, this fails because the pulse of interest is not obvious and more complex processing of the returns is needed. Schilling et al. [4] showed that the use of the entire multiple returns signal rather than a subset thereof provides more information about the scene of interest. Stochastic algorithms are computationally more intensive than the previous methods and can take significantly longer to reach a solution. Nevertheless, they may give estimates of the range to higher resolution than the classical algorithms. Typical processing methods calculate the Maximum Likelihood Estimates (MLE) of the parameters in the model using the Expectation-Maximization (EM) algorithm. The main problem is that this algorithm can converge to a local maxima or a saddle point of the likelihood and is sensitive to the initial estimates. Statistical approaches to deal with the problem of multiple returns include that of Wax and Kailath [5] who proposed detecting



(a) (b)

Fig. 2. (a) Histogram of photon counts (vertical axis) against time (measured in 4.96*ps* channels, horizontal axis) corresponding to one pixel of Fig. 5(a) and final fit from RJMCMC estimation (dotted line). (b) Variation of pixel intensity on board as function of time from a BIL system and final fit from RJMCMC estimation (dotted line).

the number of signals in a multichannel time-series using information theoretic criteria for model selection such as Akaike (AIC). In practice, they calculated for every possible number of signals the MLE estimates of the parameters of the signals and then they calculated the AIC scores for each model. The number of signals was determined by minimizing the AIC criterion. Since the maximum number of signals must be fixed a priori, there is a risk of underestimation of the actual number of peaks.

Spatial structures arise naturally in 3D ranging and imaging and allow us to incorporate inter-pixel constraints between photon/intensity signatures that image different parts of the scene. Recently, there have been various approaches in introducing spatial dependencies in a mixture distribution to take into account spatial heterogeneity using a Bayesian approach [6,7]. However, all of them consider that the different observations come from a unique mixture distribution in which the number of components and some of the parameters (maybe all) are variable. In these approaches, the spatial dependencies were introduced either through the weights of the mixture distribution [6] or through the different allocation variables [7].

1.2 Overview of Our Approach

We make a number of contributions. First, we present a unified theory of pixel processing that is applicable to both TCSPC and BIL. To that aim, we adopt a Bayesian statistical approach based on reversible jump Markov chain Monte Carlo (RJMCMC) techniques to assess the number, positions and amplitudes of the returned signals from target surfaces. Second, we incorporate spatial constraints in a multi-layered image (i.e. an image in which each pixel has several surface returns) for the first time. In our case, in each pixel there is a multivariate record (see Fig. 2(a)) so we use a mixture distribution per each pixel instead of a "global mixture" that includes all the possible returns. Moreover, the spatial dependencies are introduced through the prior distribution of the number of

elements of the mixture. Third, we show a delayed rejection step [8] in a RJMCMC algorithm. The delayed rejection step is an adaptive strategy considered to improve the performance of the RJMCMC methods in cases where the proposal distribution is badly calibrated to the target.

2 Model

Consider a rectangular grid of pixels, labeled $m = 1, 2, \ldots, N$ at each of which a multivariate measurement $\boldsymbol{y}^m = (y_1, y_2, \ldots, y_t)$ is available. Each multivariate measurement is a histogram of photon counts or intensities in which y_i^m is the value recorded in channel i, $i = \{1, \ldots, t\}$. The exact functional form of the photon count and the intensity histograms are unknown. To interpret the TCSPC data, we have used the parametric form of the expected temporal variation of the photon count distribution, employing four piecewise exponential functions [9]. In this study, we assume that the shape parameters of the returned pulses are fixed and known from an instrumental response. In the case of the BIL system, we use a look-up table of an instrumental function to interpret the intensities histogram. This is acquired from the response of the BIL system to a Lambertian reflecting surface at a similar range to the object of interest.

Since the methodology developed to analyse the different histograms is the same, we use indistinctly the notation f_{op} when referring to the TCSPC operating model or the BIL instrumental response. Furthermore, assuming that, in the same histogram, several signals can be present and that these signals will be observed against a finite background level, B_m, whose expected value is considered as a constant across all the channels, the observed histogram in pixel m, $\mathcal{F}_m(i, k_m, \boldsymbol{\beta_m}, \boldsymbol{t_{0_m}}, B_m)$, can be considered as a sample of a statistical mixture distribution with density

$$\mathcal{F}_m(i, k_m, \boldsymbol{\beta_m}, \boldsymbol{t_{0_m}}, B_m) = \sum_{j=1}^{k_m} \beta_{mj} \cdot f_{op}(i, t_{0_{mj}}) + B_m \tag{1}$$

where k_m is the number of peaks, $\boldsymbol{\beta_m} = (\beta_{m1}, \beta_{m2}, \ldots, \beta_{mk_m})$ is a vector of amplitude factors and $\boldsymbol{t_{0_m}} = (t_{0_{m1}}, t_{0_{m2}}, \ldots, t_{0_{mk_m}})$ is a vector of the time of the peak maxima.

If the time resolution is sufficiently fine, the value y_i^m recorded can be considered to be a random sample of a Poisson distribution with intensity equal to $\mathcal{F}_m(i, k_m, \boldsymbol{\beta_m}, \boldsymbol{t_{0_m}}, B_m)$ which depends on the different model parameters.

$$y_i^m \sim Poisson(\mathcal{F}_m(i, k_m, \boldsymbol{\beta_m}, \boldsymbol{t_{0_m}}, B_m)) \tag{2}$$

In order to build up a likelihood function, we make two further assumptions. First, the observations recorded in each channel i of the histogram are conditionally independent given the value of the parameters. Second, the spatial dependencies are included in Bayes' equation through the different parameters, the different $\boldsymbol{Y} = (\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^N)$, are therefore spatially independent given the

parameters. The likelihood for the total array is expressed in Eq. 3 and represents the physics of sensor operation.

$$L(\boldsymbol{Y}|\boldsymbol{k},\boldsymbol{\Omega},\boldsymbol{T},\boldsymbol{B}) = \prod_{m=1}^{N}\prod_{i=1}^{i_{\max}} e^{-\mathcal{F}_m(i,k_m,\boldsymbol{\beta_m},\boldsymbol{t_{0_m}},B_m)}\frac{\mathcal{F}_m(i,k_m,\boldsymbol{\beta_m},\boldsymbol{t_{0_m}},B_m)^{y_i^m}}{y_i^m!} \tag{3}$$

where $\boldsymbol{k} = (k_1, k_2, \ldots, k_N)$, $\boldsymbol{\Omega} = (\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \ldots, \boldsymbol{\beta_N})$, $\boldsymbol{T} = (\boldsymbol{t_{0_1}}, \boldsymbol{t_{0_2}}, \ldots, \boldsymbol{t_{0_N}})$ and $\boldsymbol{B} = (B_1, B_2, \ldots, B_N)$.

3 Bayesian Inference

The objective is inference about the unknowns of Eq. 3 in order to obtain accurate estimates of the number of peaks, position, amplitude and background of the returned signals. These unknowns are regarded as drawn from appropriate independent prior distributions. In this case, the position parameter, $t_{0_{mj}}$, is drawn from a flat uniform distribution with support in the interval $[1, i_{max}]$, to reflect in some way our complete prior ignorance of the true position of the peaks. For the prior distribution of scale parameters β_{mj} and B_m we assume a gamma distribution since the support of our parameters is on $\Re^+$. In this paper, these parameters are assumed spatially independent.

At this point, it only remains to define the prior for the number of peaks, k_m. We consider that observations that correspond to nearby locations are more likely to have similar numbers of peaks than observations from locations that are far apart. In the model investigated in this paper we use the product of two different probability distributions as the joint probability distribution of the number of peaks. For the first term, we take a uniform distribution on $[0, ..., k_{max}]$ with k_{max} chosen to be suitably large and for the second term, we take a Potts model. The first term represents the number of peaks as unstructured variables allowing the RJMCMC algorithm to propose changes in the dimensionality of the state space and the second term is considered as a penalty function if adjacent pixels have different numbers of peaks.

The Potts model has been used previously in image processing applications or in disease mapping applications to model allocation variables. We follow the formulation used in [7] in which $\boldsymbol{k}$ is modeled jointly

$$p_1(\boldsymbol{k}|\psi) = \frac{e^{\psi \cdot U(\boldsymbol{k})}}{\sum_{k \in \{1,2,\cdots,k_{max}\}^N} e^{\psi \cdot U(\boldsymbol{k})}} \tag{4}$$

where $U(k) = \sum_{m \sim m'} I[k_m = k_{m'}]$ are the number of like-labeled neighbor pairs in the configuration $\boldsymbol{k} = (k_1, k_2, \ldots, k_N)$. The parameter ψ is nonnegative and controls the amount of "smoothing", that is, $\psi = 0$ corresponds to spatially independent number of peaks and as $\psi \to \infty$ we favor patterns where like-labeled number of peaks are neighbours.

The full joint prior distribution can be modeled as

$$\boldsymbol{f}(\boldsymbol{k},\boldsymbol{\Omega},\boldsymbol{T},\boldsymbol{B}) = p_1(\boldsymbol{k}|\psi)\left(\frac{1}{k_{max}}\right)^N\left(\frac{1}{i_{max}}\right)^{kN}\prod_{m=1}^{N} f_G(B_m|c,d)\prod_{j=1}^{k} f_G(\beta_{mj}|a,b) \tag{5}$$

where f_G is the probability density function (pdf) of a gamma distribution with mean $a \cdot b$ ($c \cdot d$ respectively) and variance equal to $a \cdot b^2$ ($c \cdot d^2$ respectively).

Using equations 3, 5 and Bayes' theorem the target distribution can be expressed as

$$\pi(\boldsymbol{k},\boldsymbol{\Omega},\boldsymbol{T},\boldsymbol{B}|\boldsymbol{Y}) \propto L(\boldsymbol{Y}|\boldsymbol{k},\boldsymbol{\Omega},\boldsymbol{T},\boldsymbol{B})\boldsymbol{f}(\boldsymbol{k},\boldsymbol{\Omega},\boldsymbol{T},\boldsymbol{B}) \tag{6}$$

since we just have to know the posterior distribution up to a normalizing constant.

However, as simulating from the joint probability distribution is difficult we propose to update the different parameters using their respective full conditional distributions. We define a Markov random field and use a second-order neighborhood. The Hammersley-Clifford theorem [10] ensures that we can use the full conditional distribution of Eq. 4.

$$p_m(k_m|\ldots) \propto \exp(\psi \sum_{m' \in\ \delta_m} I[k_m = k_{m'}]) \tag{7}$$

where δ_m denote the neighbors of m. In this way, a rather complex multivariate probability distribution of a MRF can be obtained by successive simulations from the full conditional distribution. Since these simulations can be performed in parallel, considerable gains in computing time are possible.

4 Data Analysis

We use reversible RJMCMC techniques [11] to assess the number, positions and amplitudes of the returned signals from target surfaces. This allows moves between state spaces with different dimensionality, i.e. different numbers of viewed surfaces, and moves when the dimension is fixed to improve the estimation of the parameters of such subspace, i.e. position (depth) and amplitude (reflectance) of surfaces. RJMCMC provides a natural method for updating beliefs in response to new information and incorporates prior knowledge into the analysis. Indeed, RJMCMC techniques allow us to explore the full posterior distribution of the parameters of the mixture distribution of Eq. 1, given the data values Y supplied by the different histograms. The parameters we seek are estimated from the values of a Markov chain whose limiting distribution is a target distribution π. In our case, π arises from the posterior distribution defined by Eq. 6, nevertheless, we use simpler versions of it involving only full conditional distributions.

The Markov chain constructed involves moves of various types. These are: (a) updates to parameters $\boldsymbol{\Omega},\boldsymbol{T},\boldsymbol{B}$, (b) the random birth of a peak, (c) death of a peak, (d) the random splitting of a peak into two peaks and (e) merging of two peaks into a single peak. Movements (b), (c), (d) and (e) are governed by a

probability explained in [11]. Assume we are at pixel m[1] and the current state of the parameter variable set is ϕ, with $\phi = (\boldsymbol{k}, \boldsymbol{\Omega}, \boldsymbol{T}, \boldsymbol{B})$, we propose a move n to a state $\phi' = (\boldsymbol{k}', \boldsymbol{\Omega}', \boldsymbol{T}', \boldsymbol{B}')$ drawn from the proposal probability density, $q_n(\phi, \phi')$. The proposed move is accepted with probability:

$$\alpha_m^1(\phi, \phi') = min \left\{ 1, \overbrace{\frac{\pi(\phi'|y)}{\pi(\phi|y)}}^{Target\ ratio} \times \overbrace{\frac{q_n(\phi', \phi)}{q_n(\phi, \phi')}}^{Proposal\ ratio} \right\} \tag{8}$$

If the proposed move is accepted, the new value of the Markov chain becomes ϕ' otherwise we keep the current value ϕ. Note, that if the number of peaks does not change, i.e. a fixed dimension of the space of solutions, the acceptance probability of Eq. 8 reduces to the Metropolis-Hastings acceptance probability and we perform a parameter updating using the full conditional distributions as a target distribution. Whereas the target distribution is given by the shape parameters of the returns, the proposal distribution is chosen to be representative of the solution, yet to allow rapid convergence to a limiting distribution. In our case we draw the proposal of the amplitude, β'_{mj}, and the background, B'_m, from the gamma distributions $\Gamma(1.5, \beta_{mj})$ and $\Gamma(1.5, B_m)$ respectively whereas the proposed position, $t'_{0_{mj}}$, is drawn from the normal distribution $\mathcal{N}(t_{0_{mj}}, \sigma_1)$.

The birth, death, split and merge moves imply changes in the parameter-space dimensionality (k changes). In these cases, the acceptance probability of Eq. 8 can be reduced to:

$$\alpha_m^1(\phi, \phi') = min \left\{ 1, \overbrace{\frac{\pi(\phi'|y)}{\pi(\phi|y)}}^{Target\ ratio} \times \overbrace{\frac{r_n(\phi')}{r_n(\phi)q(u)}}^{Proposal\ ratio} \times \overbrace{\left|\frac{\delta(\phi')}{\delta(\phi, u)}\right|}^{Jacobian} \right\} \tag{9}$$

where $r_n(\cdot)$ is the probability of choosing move type n when in state ϕ, u is a vector of continuous random variables which ensures the reversibility of the deterministic function $\phi' = h(\phi, u)$, which allows the move to a higher-dimensional space, and *q(u)* is the pdf of *u*. The Jacobian term joint to this deterministic function ensures the "dimension-balancing" condition which allows reversible jumps between different state spaces. The Jacobian arises from the change of variable from (ϕ, u) to ϕ'.

For a birth we propose values for the parameters of the new peak. In practice, we propose values drawn from a gamma and a uniform distribution for the amplitude and the position of the peak respectively. For a death, one of the current peaks is chosen at random to be removed. The split and merge moves are more complicated. For a split, again a peak is chosen at random and divided into two peaks. The split is performed so that the new two peaks equate to the total mass of the peak split. For a merge, the initial peaks to be amalgamated must be within a temporal separation below a certain threshold, and mass balancing is again considered, that is, the total mass of the new peak must be equal to the addition of the total mass of the initial peaks.

[1] The superscript m will be dropped for notational convenience.

Our approach also includes a delayed rejection stage in the RJMCMC algorithm as proposed in [8] in order to improve the mixing of the Markov chain which entails an improvement of the estimates of the different parameters. The main advantage of this stage is that is allowed to depend on the current position of the chain and also on the value we have just proposed and rejected and therefore we learn from the information obtained during the run of the chain. We applied this stage to the position and amplitude parameters. Since we do not have any a priori knowledge of the state space, the proposal distributions are very vague as their spread depends on the current position of the chain. As a consequence, the parameter update is not very accurate and slow. Using delayed rejection, we are able to explore values that are further away or very close to the current estimates since the spread of the proposals can be tuned through the different stages.

The basic idea of this stage is that whenever a candidate move is rejected, we make another attempt to move using a second proposal instead of turning to the next transition. In a fixed-dimension, this can be summarised as follows: when at ϕ (k fixed) we propose a new state ϕ' from a density $q_1(\phi, \phi')$. If the move is rejected, a second proposal, ϕ'', is made from a density $q_2(\phi, \phi', \phi'')$ and the new acceptance probability is $\alpha_m^2(\phi, \phi'') = \min\{1, A\}$, where A is

$$A = \frac{\pi(\phi'')q_1(\phi'', \phi')q_2(\phi'', \phi', \phi)\{1 - \alpha_1(\phi'', \phi')\}}{\pi(\phi)q_1(\phi, \phi')q_2(\phi, \phi', \phi'')\{1 - \alpha_1(\phi, \phi')\}} \tag{10}$$

In our case, the new values of our position and amplitude parameters for the delayed rejection stage were drawn from $\mathcal{N}(t_{0_{mj}}, \sigma_2)$ and $\mathcal{N}(\beta_{mj}, \sigma_3)$ respectively.

5 Experimental Results

Initially, we consider analysing a small window of TCSPC data with spatial constraints. The data corresponds to a pixel array of 50 by 50 histograms of photon counts from the structure shown in Fig. 3(a). The area analysed corresponds to that shown in the reflectance image of Fig. 3(b). The data set has been analysed with the model defined in previous sections with the followings settings for the previously unspecified constants: $\sigma_1 = 1000$, $\sigma_2 = 10$, $\sigma_3 = 50$, $a = 6$, $b = max(y)/2$, $c = 1.03$ and $d = 1000$. The values of a and b are chosen to penalise very small returns whereas the values of c and d have been chosen to make the prior on the background as vague as possible. For the first term of the prior on k we took a uniform distribution on the integers $0, 1, \ldots, k_{max} = 8$. For the second term, we took a Potts model defined by Eq. 7 with parameter $\psi = 0.5$.

The results correspond to 1000 sweeps of the algorithm described in section 4. Fig. 4 shows the marginal posterior distribution of the number of peaks for a second order neighbourhood of the data array analised without (Fig. 4(a)) and with spatial constraints (Fig. 4(b)). As can be seen from the different figures, the Potts model forces the different pixels to have a similar spatial structure in the number of peaks whereas when using non spatial constraints the number

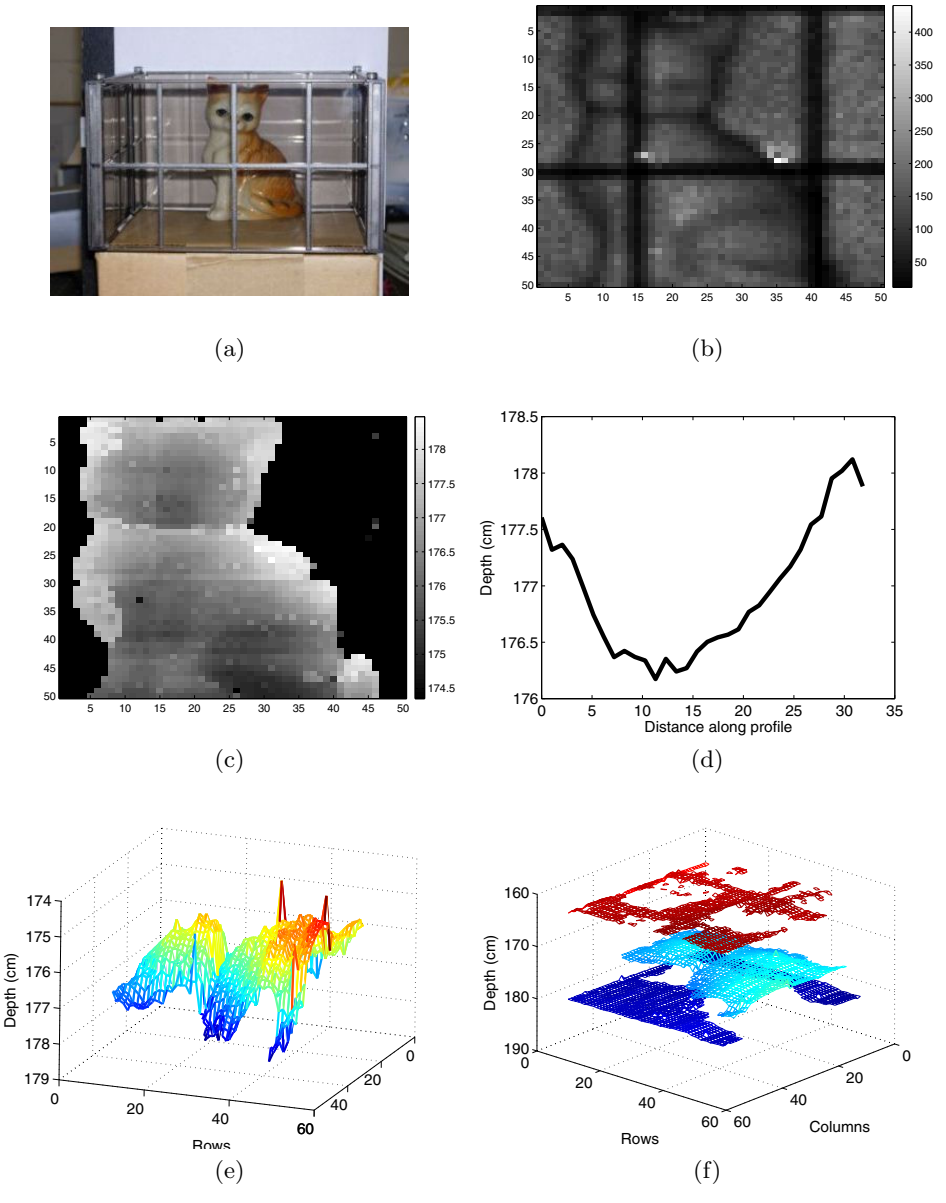
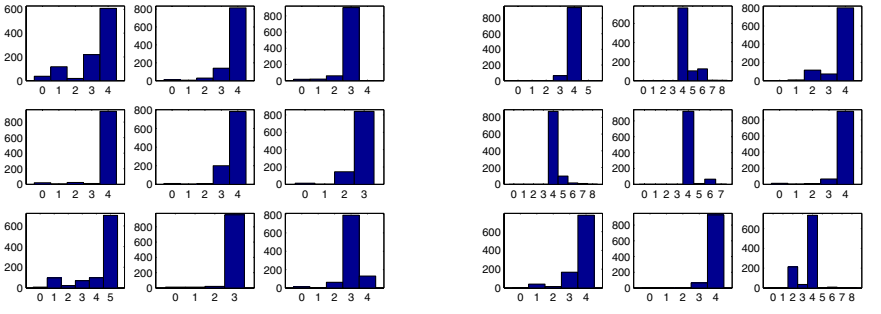


Fig. 3. (a) Image of complex multiple return structure composed of a toy cat and a semi-transparent toy greenhouse (b) Reflectance image obtained with TCSPC system, the unit is photon counts (c) False color depth image of the toy cat (d) Depth profile of the toy cat corresponding to row 30 (e) Mesh of the toy cat (f) Multiple layer structure of the target

of peaks from pixel to pixel is variable. Moreover, the smoothing parameter forces the different neighbours to have the same number of peaks and therefore "smoothing" the number of different possible surfaces. The "extra" number of peaks is possible due to the footprint of the beam impinging on a target with surfaces distributed in depth.

Fig. 3(c) shows the toy cat and its depth variation using a false color image whereas Fig. 3(d) shows a depth profile of the previous image corresponding to row 30 where can be seen that the algorithm captures the curvature of the object. Fig. 3(e) and 3(f) show a mesh reconstruction of the toy cat and a three layer representation of the complete target. The first layer (closer to the detector) represents the first glass of the toy greenhouse. Note, that all the values are not available. This is due to three main factor: first the reflectance of the surface is very poor (semitransparent surface). Second the returns are very below the background noise and on the other hand the incidence of the laser was not normal and therefore reducing the number of photons detected. Third, the prior on k is "oversmoothing" the number of peaks. The second layer corresponds to the toy cat whereas the third layer corresponds to the second glass of the toy greenhouse. The distance between the pair of glass surfaces is estimated to be about 18.0cm which is consistent with the dimensions of the toy greenhouse.



(a) (b)

Fig. 4. Marginal posterior distribution of the number of peaks for a determined neighbourhood of 9 pixels of the scene (toy greenhouse and toy cat) (a) Non spatial constraints considered (b) Spatial constraints considered

We have acquired 3D images of a number of test objects of varying size at different distances and we have applied the RJMCMC method using a delayed rejection step to both TCSPC LaDAR data and range gated BIL data. Fig. 5 shows a series of examples from the TCSPC and BIL data with similar settings to those described previously, corresponding to 1000 sweeps of the algorithm and in which only one return is shown. Fig. 5(a) and 5(c) show respectively a toy fighter and its corresponding depth image. The toy fighter is mounted on a clamp that is also visible in the depth image. The data was obtained using a TCSPC scanned system. Fig. 2(a) shows an example of a histogram return corresponding

to one pixel of the image and its respective fitting using RJMCMC. Fig. 5(d) shows a meshed reconstruction of the BIL image of the three planar targets (Fig. 5(b)) that gives some idea of the depth variation across the targets. The peaks in the depth data adjacent to the planes are caused by ground reflections before or beyond the planar targets. The planar targets were placed at depth intervals of approximately $12m$. The central planar target was placed at distance of about $204m$ from the receiver. Fig. 2(b) shows the temporal intensity response of a single pixel. The intensity values are plotted on a log scale for illustrative purposes.

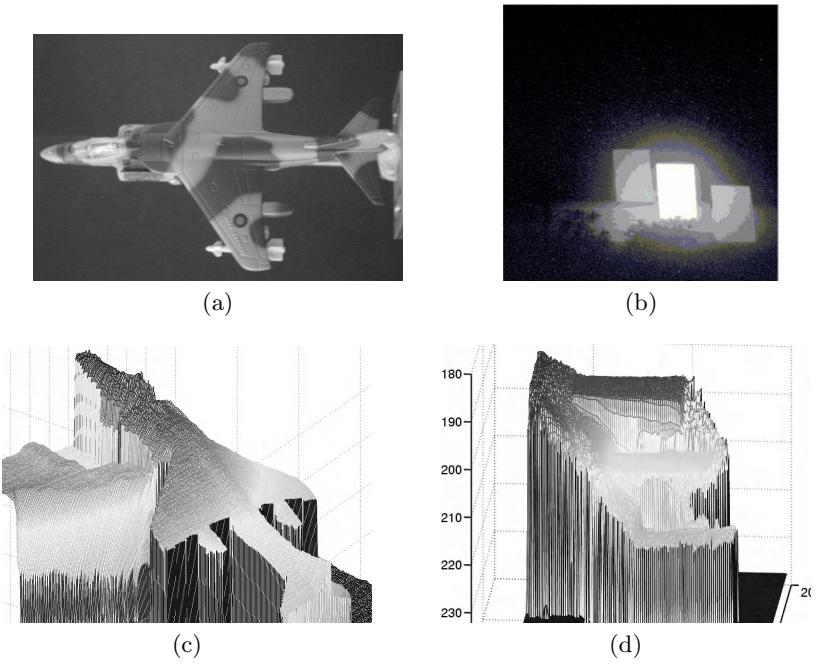


Fig. 5. (a) Image of a toy fighter (b) Log-intensity, single frame BIL image with stand-off distance centred on the middle of three uniform, matt planar targets, situated approximately 12m apart. (c) Mesh (3D image) of the toy fighter acquired from a TCSPC system. (d) Mesh of three planar targets from a BIL system.

6 Conclusion

We have described the development and application of RJMCMC techniques using a delayed rejection stage and prior information on the number of peaks to process time-of-flight data using both single photon counting and variable temporal gating to extract range measurements. Our initial results show that it is possible to resolve multiple returns and hence characterise objects distributed in 3D space and to extract an underlying spatial structure in the data processed.

We anticipate that the results obtained from processing spatial dependencies on the number of peaks will be improved by consideration of constraints on other parameters such as position and amplitude of the peaks, but that remains a goal for future work.

Acknowledgements. The authors would like to acknowledge the support of BAE SYSTEMS, SELEX S&AS and the Royal Society.

References

1. Wallace, A.M., Csakany, P., Buller, G., Walker, A.: 3D imaging of transparent objects. In: Proc. of British Machine Vision Conference. (2000)
2. Wallace, A.M., Buller, G.S., Walker, A.C.: 3D imaging and ranging by time-correlated single photon counting. Computing & Control Eng. Journal **12** (2001) 157–168
3. Busck, J., Heiselberg, H.: Gated viewing and high-accuracy three-dimensional laser radar. Appl. Opt. **43** (2004) 4705–4710
4. Schilling, B.W., Dallas, B.N., Templeton, G.C., et al.: Multiple-return laser radar for three-dimensional imaging through obscurations. Applied Optics **41** (2002) 2791–2799
5. Wax, M., Kailath, T.: Detection of signals by information theoretic criteria. IEEE Trans. on acoustics, speech and signal processing **ASSP-33** (1985) 387–392
6. Fernandez, C., Green, P.: Modelling spatially correlated data via mixtures: a Bayesian approach. J. Roy. Statist. Soc. Ser. B **64** (2002) 805–826
7. Green, P., Richardson, S.: Hidden Markov models and disease mapping. J. American Statistical Association **97** (2002) 1055–1070
8. Tierney, L., Mira, A.: Some adaptive Monte Carlo methods for Bayesian inference. Statistics in Medicine **18** (1999) 2507–2515
9. Pellegrini, S., Buller, G., Smith, J., Wallace, A., Cova, S.: Laser-based distance measurement using picosecond resolution TCSPC. Meas. Sci. Technol. **11** (2000) 712–716
10. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. J. Roy. Statist. Soc. Ser. B **36** (1974) 192–236
11. Richardson, S., Green, P.: On Bayesian analysis of mixtures with an unknown number of components. J. Roy. Statist. Soc. Ser. B **59** (1997) 731–792

A Multi-Modal Interface for Road Planning Tasks Using Vision, Haptics and Sound

Matt Newcomb and Chris Harding

Human-Computer Interaction Program, Virtual Reality Applications Center (VRAC),
Iowa State University, Ames IA, USA
talenos@iastate.edu, charding@iastate.edu

Abstract. Planning of transportation infrastructure requires analyzing combinations of many different types of geo-spatial information (maps). Displaying all of these maps together in a tradition Geographic Information System (GIS) limits its effectiveness with visual clutter and information overload. Multi-modal interfaces (MMIs) aim to improve the efficiency of human-computer interaction by combining several types of sensory modalities. We are presenting a prototype virtual environment using vision, haptics and sonification for multi-modal GIS scenarios such as road planning. We use a point-haptic device (Phantom) for various haptic effects and sonification to present additional non-visual data while drawing on a virtual canvas. We conducted a user study to gather experience with this multi-modal system and to learn more about how these users interact with geospatial data via various combinations of sensory modalities. The results indicate that certain forms of haptics and audio were preferentially used to present certain types of spatial data.

Keywords: Multi-modal interfaces, road planning, GIS, haptics, sonification.

1 Introduction

When civil engineers plan new infrastructure, such as roads, railroads, pipelines, or power lines, they commonly use a Geographic Information System (GIS) to work with the relevant geospatial data. In a GIS, spatial data is displayed traditionally in many overlapping "layers", a road planning project (on which we focus here) may look at the combination of 1) already existing roads, 2) urban structures (such as cities), 3) the terrain's elevation (digital elevation model or DEM), 4) the soil's engineering properties and 5) environmentally sensitive areas (like wetlands). An (admittedly visually cluttered) example for planning a highway connection in a rural area is shown in Fig. 1, combining only six "overlapping" layers.

1.1 An Overview of the Process of Planning Roads in Rural Areas

Planning roads in rural areas is a complex process with many stages - from a broad look at all the contributing factors (spatial and non-spatial), to finding a corridor or strip of land for the general route and the precise placement of the actual road. Although a detailed description for the creation of the optimal road is beyond this

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 417 – 426, 2006.
© Springer-Verlag Berlin Heidelberg 2006

paper's scope, examples of some of the more intuitive rules for road placement are: a shorter road that requires a tunnel through a mountain may be more expensive than a longer road around the mountain; traversing farmland is less expensive and therefore more desirable than traversing dense forest. In addition, the impact on environmentally sensitive areas, potential expandability in the future and even political issues such as social justice also need to be considered.

In summary, the process of planning the placement of a new road is by no means straightforward. It involves multiple, overlapping sets of spatial data and multiple steps. Only a few of the planning problems can be tackled with automated (algorithmic), computer-optimized solutions – most decisions are still made by a human, based on expert knowledge and experience. Our goal is to provide a novel multimodal framework for planning new roads, in which practitioners are able to transform their expert knowledge and past experiences into an initial road layout more efficiently than when using a traditional GIS.

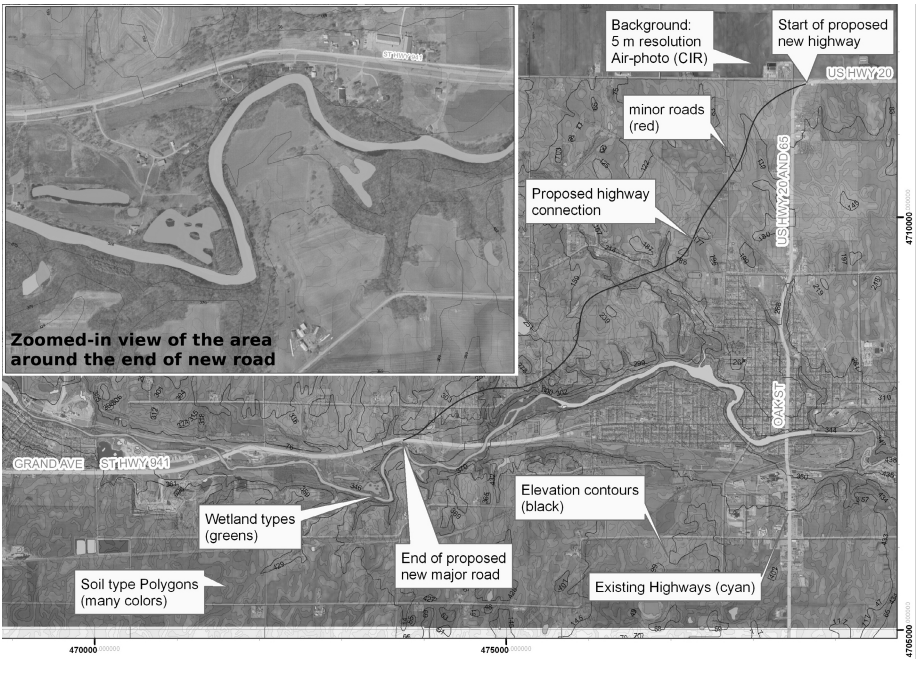


Fig. 1. Example of combined geospatial layers used in a traditional GIS. The exact placement (digitizing) of the new road (labeled "proposed highway connection") based on information in these layers, is the focus of our project.

1.2 Problems with the Traditional Systems

At the road placement stage, on which we focus, it is vital for the user to directly analyze several layers of geo-spatial information simultaneously, to digitize line segments (the proposed road) based on this analysis and to be confident in this placement. However, displaying large amounts of spatial information purely through the

visual channel may cause information overload and visual clutter. Planners using today's GIS, are forced to a) display many overlapping layers at once, either combining semi-transparent layers of raster data and to overlay these with discrete vector data (polygons, lines or points) or b) constantly switch between different layers.

Both techniques are problematic: combining semi-transparent layers can introduce color ambiguities; switching between layers forces the users to keep the switched-off spatial data in working memory and combine the full picture in their own heads. As it is desirable to make spatial decisions in a more holistic way by perceiving all the important data at once, we propose that it could be possible to use a combination of the three major senses, i.e., a multi-modal system, for some parts of road planning tasks. The following section will give a brief background on multi-modal interfaces and why we think they can solve the problems of visual overload by providing a more holistic way of perceiving spatial data.

2 Background on Multi-Modal Interfaces (MMI)

2.1 Theoretical Background of MMI

Multi-modal interfaces present information via multiple sensory modalities. The field of MMIs is receiving a lot of attention and seems to be motivated by the expectancy that MMIs can be used to create novel and effective user interfaces. Multi-modal theories suggest that by channeling information through multiple senses, more information can be processed simultaneously and potentially more effectively. According to Miller's seminal 1956 paper "The magic number seven", there is only a finite amount of discrete information "chunks" that we can keep in our immediate memory (for less than a second). He postulated that it could be possible to increase that amount by organizing data into different "dimensions". Since then many have applied Miller's "dimensions" to multimodal channels. For example, Samman (2005) showed that working memory capacity in multi-modal tasks surpassed the one-dimensional working memory limits by a factor of up to three. Oviatt (2004) showed that MMIs can be helpful in reducing cognitive load and improving task performance. These results support the multiple resource theory (Wickens, 2002), which postulates potential interference between tasks that share sensory modalities and therefore suggests that using different modalities for different tasks could be helpful and more efficient. Based on these theories, our goal is to investigate if multi-modal interactions with overlapping, multi-dimensional, geo-spatial data could also would decrease cognitive load and increase efficiency.

2.2 Previous Work on Relevant MMIs

The following selection of recent MMI was relevant for designing our multi-modal system for road planning. A study reported by Emery (2003) on user performance of 2D mouse-based drag-and-drop task showed that the auditory cues improved performance when they were present and haptic cues showed the best improvement when combined with other modes, especially auditory. McGee (2001) tested the perception of surface roughness when using combined haptic/audio textures. Certain combinations of multi-modal texture were helpful while other combinations produced

interference. Wall (2003) reports on using a Phantom to map data to different friction values. Work by Griffin (2002) and Jacobson (2002) indicate that haptic and/or audio cues are effective at working with spatial information.

Krygier (1994) presents a system of representing spatial data via realistic of abstract sounds that uses analogies to visualization. Fisher (1993) uses the duration of sound to communicate the reliability of pixels on a remotely sensed image. A predecessor to our current system was used for hapto-visual suitability analysis (Harding & Newcomb, 2004)

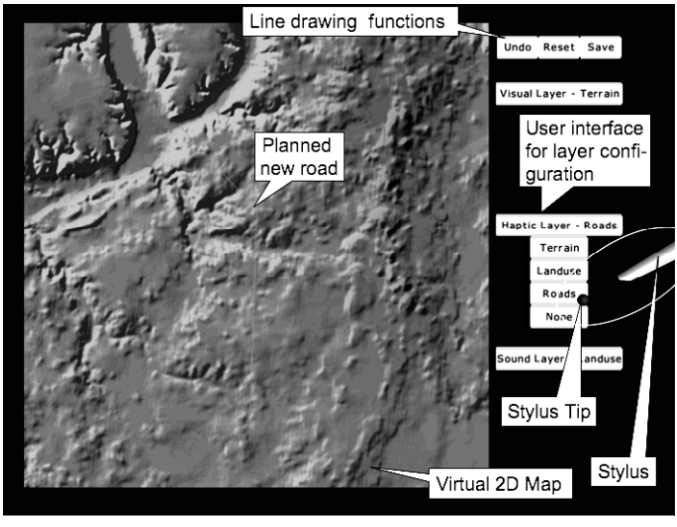


Fig. 2. A screenshot of the system showing the terrain surface (left) and the haptic user interface (right). The circle shows the visual model of the user's stylus ending in a tip. This tip is used to press the interface buttons when selecting a data layer for a modality; it is also used to draw directly on the virtual map.

3 Methods: Technology Used in Our System

Our system deals with interactive, multi-modal, data exploration of overlapping spatial data (via visual, haptic and audio) and the creation of new line segments (new roads). We use a stylus-like interface for digitizing lines (drawing roads) onto a "paper" plane (map). Fig. 2 shows the user's view of this virtual map, a intuitive metaphor for drawing of lines on paper maps. The user interface (Fig. 2, right) is operated by touching haptic buttons and sliders with the stylus' tip - it lets the user configure which type of spatial data is mapped to which sense (e.g., terrain to visual, land use to audio and roads to touch) as well as direct drawing of new roads on the virtual map.

We use a Desktop Phantom, a grounded haptic point force-feedback device (Fig. 3) to set up an impenetrable 2D plane over which we drape the "visual" map. The stylus-tip (corresponding to the black sphere attached to a handle shown in Fig. 2) provides the location for drawing the new road. As the users make contact with the map, they

can see the visual map and the tip's current location on it, feel a force representation of the haptic map (touch) and hear a sound map (sonification) of the data under the tip's current position. When in contact with the map, the users may activate a button on the stylus (comparable to clicking a mouse-button) to digitize a new road onto the map and still perceive additional data via force and audio feedback.

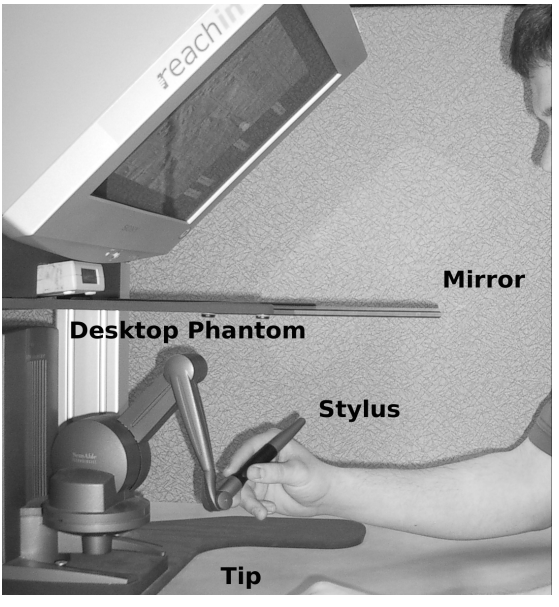


Fig. 3. Our phase 2 planning system on the mirrored Reachin display, the user looks into the mirror at a virtual map and interacts with it via a force-feedback pen

To provide a believable re-creation of drawing on a paper map, we use a mirrored desktop setup (Reachin display, Fig. 3) in which the user looks down into a horizontal mirror while moving the stylus beneath it. In this setup, the real-world workspace (i.e., the hand movement of the user) and the virtual workspace (what the user sees and feels) overlap perfectly.

3.1 Data Set Used in the User Study of the Multi-modal Road Planning Prototype

The data represents a 10 km by km area in Iowa and consists of raster data (terrain elevation, road-buffers and land use) and line data (existing roads). Using Stevens' taxonomy, terrain elevation and distance-to-nearest-roads are numerical data-sets (both in meters) and land use is a categorical data-set with four classes: city, river, forest and cornfields. For each layer we created a visual, haptic and auditory representation to create different data/modality combinations.

3.1.1 Data Layer Representation Via Visual Means

For the visual terrain layer, a 10 m resolution, digital elevation model (DEM) was colored with a smooth color scale and added hill shading to bring out elevation differences (Fig. 2).

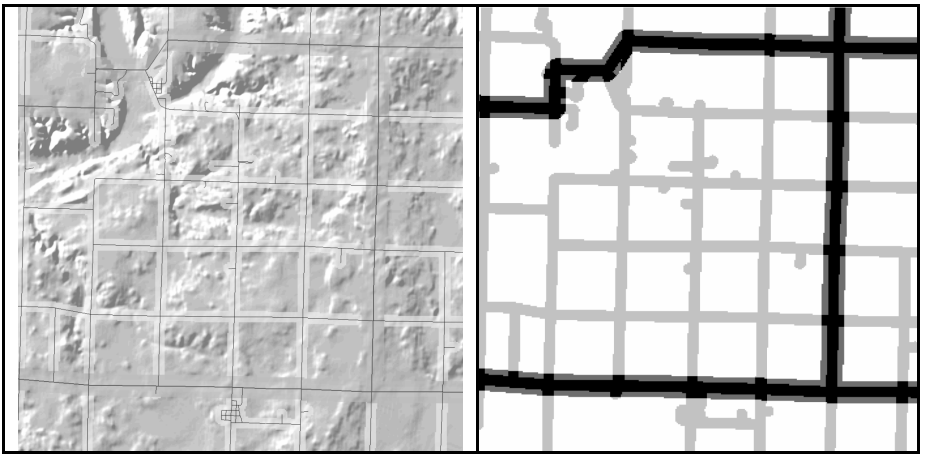


Fig. 4. Existing roads with buffer zones (left); audio/haptic representation of the road buffers (right), dark: > 120 m to a major road, light: > 60 m of minor roads

3.1.2 Data Layers Representation with Haptic (Touch) Effects

We used three different haptic effects, depending on the type of data presented (terrain, road-distance and land use. Terrain elevation (a numeric variable) is presented directly via "haptic bumpmapping", the lower the elevation the more the tip sinks into the surface. After some tests, we calibrated the difference from the lowest elevation to the highest elevation point to about 1 cm, enough to let the users perceive subtle features and use this feedback to run the tip along valleys or flat areas.

To tell the user about the distance to existing roads (a numerical variable from 0 m to 120 m), we set up a haptic gravity field. If the tip touches the map and is within a road's buffer-zone, the tip is attracted perpendicularly to the closest road. This attraction force is kept constant within a buffer. However, major roads have a stronger attraction force than minor roads. To avoid a buzzing caused by the stylus being pushed in random directions in the middle of the buffer, we created a graduated dead-zone directly on top of a road. If the users simply follow this "suggested" force vector, the tip is guided to follow the most important road in range. We used early tests to scale rate of change in force magnitude appropriately, the "width" of the gravity zone corresponds to the buffers shown in Fig. 4 (right).

For the categorical land use data, we created a friction map (Fig. 5, right) where each pixel's value represents the haptic equivalent to a unique, distinct "haptic color". We use suitably different values (based on a power series) to create different degrees of surface friction. We order the degree of stickiness according to "planning-suitability": fields (in the most abundant shade in Fig. 5 left) have the lowest friction as are more suited for building a road, forests, rivers and cities have higher friction.

Admittedly this strategy goes beyond the requirement for a categorical data type. We could have simply used an arbitrary but unique friction for each category. We are still working on a better approach for using haptic textures and not just friction. Despite documented problems in differentiating textures via non-tactile, point-haptic devices such as the Phantom, we may be able to use "haptic sandpaper" to design a haptic "forest texture" that feels different to a haptic "field texture".

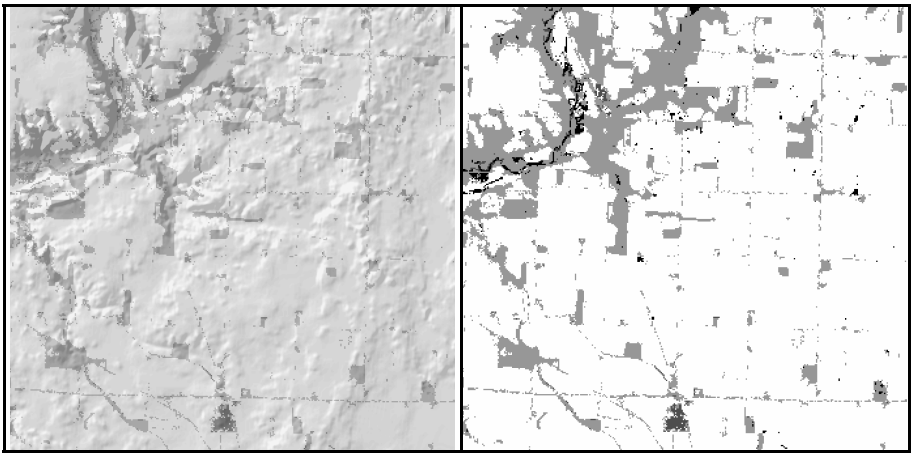


Fig. 5. Visualization of land use data with four categories: fields, forest, rivers and cities (left) and their haptic/audio representation (right)

3.1.3 Data Layer Representation Via Sound

We created three different audio maps where each pixel contains sound information that is interpreted depending on the meaning of its data. For numerical data (terrain elevation and distance-to-road) we use real-time MIDI audio with only pitch changes; pitch coding of numerical values capitalizes on the ear's ability to perceive increasing and decreasing pitch/data (Flowers, 2005). For terrain we increase pitch with elevation increase. After experimenting with different schemes we now use whole tones (1 octave) from low to high - this granularity allows the users to hear approximately level terrain as a steady pitch and is reasonably pleasing to most ears. For the distance-to-road, a sound is played when the tip enters a sound buffer, with a higher pitch for the major roads and a lower pitch for the dirt roads. No sound is played when the stylus is not inside a buffer.

We experimented with mapping the concept (symbol) contained within the land use data (town, forest, water) into a environmental sound, we created "sound snippets" (similar to auditory icons, Gaver, 1986) that use noises typical for the environments they are meant to symbolize. We used snapping branches and bears as a "symbol" for forest, people talking and cars honking for towns, insects chirping and wind for open plains and running water for rivers. We found it important to start each snippet with a characteristic, very short "key signature" sound (honk, cricket chirp). See www.vrac.iastate.edu/~charding/snippets for different sound files.

4 User Study of the Multi-modal Road Planning System

4.1 Setup and Research Questions

Given the limited amount of available data on the use of a MMI for road planning we choose to first conduct an informal user study, rather than to get quantitative measurements about the subject's performance, error rates, mental work load, etc.

Our 12 subjects were upper level civil engineering students (avg. age 24) that had gone through a semester-long road planning class and had experience with a similar task with comparable data on a traditional GIS – we therefore consider them "novice experts". We deliberately limited the data to three layers: terrain elevation, land use and existing roads; there are a total of nine different data-modality combinations (e.g. terrain/visual + roads/touch + land use/audio, etc.).

After completing a short background questionnaire, the subjects experimented with each data-modality combination for several minutes. We then reinforced the main rules for road planning to the subject (make the road as short as possible, avoid going through expensive land, such as forest, re-use existing roads if possible, etc.). We then told the subjects to connect two points on the map by creating a new road, they were free to switch to any data-to-sensory-modality configuration for as long as they wanted, there was no time-limit set and subject could digitize the new road as a whole or in pieces. We had the subjects use the think-aloud technique during the task and recorded their voice and interactions (position of stylus, current configuration, user-interface interactions) on video. Each subject filled out a short exit questionnaire (likes/dislikes, strategy when planning, confidence in their solution, etc.). Besides this general data gathering, we looked at three research questions in particular:

1. What data-modality combinations did the subjects prefer during the task?
2. Did the subjects have a strategy for switching between different data-modality combinations?
3. Was the visual sense used for global orientation and navigation, while the haptic/audio senses would be used locally to "add-value" while digitizing?

4.2 Results and Discussion

Because of space limitations, we can only touch on a few points here - for more in-depth results and discussions, see Newcomb, 2006. Fig. 6 shows the time that all subjects spend with-in each on the nine data-modality combinations. Terrain elevation (left) was mostly used in its haptic form, which may reflect its ability to find and stay in flat areas or valleys. Land use data (middle) was preferred in its sound form, which may reflect the ease of

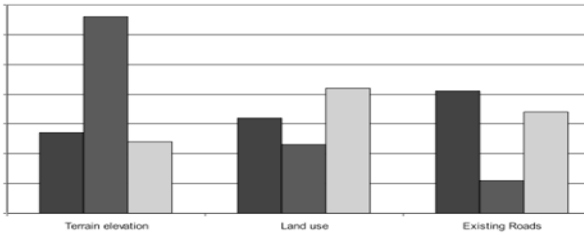


Fig. 6. Time spent on data-modality combination (black: visual, gray: touch, white: sound, left: elevation, middle: Land use and right: existing roads)

recognition of auditory icons. Data on existing roads (right) was preferred slightly more in the visual form, which we postulate is because of the grid like nature of the roads that required seeing all the information at once as opposed to getting one point of information that you get from sound or touch. It is interesting to note that this observed data corresponds with the subjective or user reported data where subjects were asked to pick their favorite multi-modal combinations. Haptics was picked most for terrain, sound with land use and visual for existing roads.

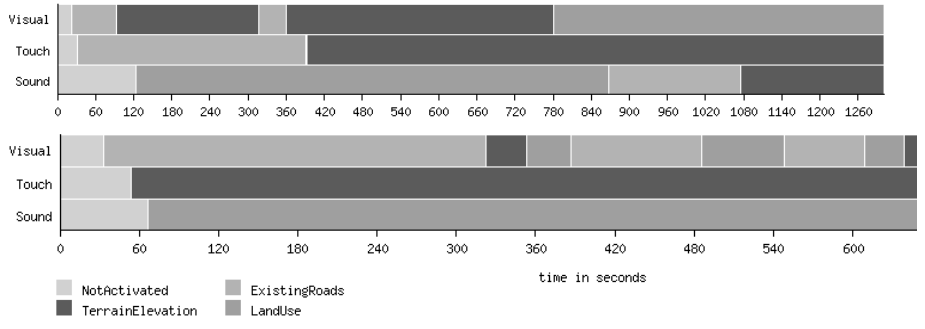


Fig. 7. Timeline of data-modality configuration for two of the 12 subjects

We created timeline diagrams for each subject to see if we could identify one or more strategies for switching between configurations. Fig. 7 shows diagrams for two of the 12 subjects - note that no data is activated at the start. The lower diagram shows an "average" use, other than on the visual modality there is very little switching and the number of switches and the durations are fairly typical. This visio-centric subject apparently had early on decided on the setup of the non-visual modalities and changed mostly the visual modality's data. The upper diagram is more typical example of switching within all modalities - most of the subjects would "play around" with different combinations (including redundant combinations), use a combination for a while and switched when a new strategy seemed to necessitate a new combination. On average there were typically 3-4 data switches for the non-visual modalities and 4-6 switches for the visual modality.

The evaluation of the video-tapes and exit questionnaires indicate that our MMI indeed enables both a sense of global orientation and a local but holistic idea of the data situation (akin to "gist" in Zhao, 2004). Our subjects seemed to use a global mental plan ("start here, then go here, avoid this and end up here"), and refer to this plan via visual means but in addition use a holistic "local zone of cognition" over the map to generate the positions of the new roads.

5 Conclusions and Future Work

We have created prototype system for multimodal interaction with geospatial data used in a road-planning context. The system offers nine combinations of presenting three types of data layers via vision, audio and touch for digitizing a new road. The

current system was scaled down to allow us to study the behavior of novice experts using the system. From this twelve-subject user study we observed some general usage patterns and collected valuable feedback. Our next goal is to use the results from this initial user study to create a system for road-planning experts and to quantitatively measure this group's task performance and mental workload when performing more sophisticated and realistic road planning tasks.

References

Emery, V.K., Edwards, P.J., Jacko, J.A., Moloney, K.P., Barnard, L., Kongnakorn, T., Sainfort, F., Scott, I.U. Toward Achieving Universal Usability for Older Adults Through Multimodal Feedback. Proc. of the 2003 Conference on Universal Usability, 2003.

Flowers (2005) Thirteen Years of reflection on Auditory Graphing, proc. of ICAD 2005, Limerick, Ireland, 2005

Fisher, P.F., Visualizing uncertainty in soil maps by animation. Cartographica, 30 (2&3), 20-27. 1993.

Gaver, W, Auditory Icons: Using sounds in computer interfaces, Human-Computer Interaction, 2, 167-177, 1986.

Griffin, A.L. Feeling it out: The use of haptic visualization for exploratory geographical visualization. Cartographic Perspectives. 39. 12 - 29. 2002.

Harding, C., Newcomb, M., Supporting Interactive Data Exploration for GIS Planning Tasks with a Multi-modal Virtual Environment. Proc. of HAVE 2004.

Jacobson, R.D. Representing Spatial Information through Multimodal Interfaces. Proc. of the Sixth International Conference on Information Visualisation. 2002.

Krygier, J.B., Sound and Geographic Visualization. In Visualization in Modern Cartography MacEachren, A. and Taylor D.R.F. (Eds.) New York: Pergamon. pp. 149-166. 1994

McGee, M.R., Gray, P., Brewster, S. The Effective Combination of Haptic and Auditory Textural Information. Proc. Of First International Workshop on Haptic Human-Computer Interaction. 2000.

Miller, G.A. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, vol. 63, pp. 81-97, 1956.

Newcomb, M.,C., Results from a user study of a multi-modal road planning system, MS thesis, Iowa State University, 2006 (in preparation).

Oviatt, S., Coulston, R., Lunsford, R. When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns. In Proc. of the Sixth International Conference on Multimodal Interfaces (ICMI 2004), Pennsylvania, USA, October 14-15, 2004.

Samman S., Stanney, K., Sims, V. Think Multimodal to Maximize Information Management Capacity. HCI International 2005.

Wall, S., Brewster, S.: Scratching the Surface: Preliminary Investigations of Haptic Properties for Data Representation, Eurohaptics, Dublin, Ireland, pp. 330-342, 2003

Wickens, C.D. Multiple resources and performance prediction. Theoretical Issues in Ergonomics Science, vol. 3 no. 2, 159-177, 2002.

Venation Pattern Analysis of Leaf Images

James Clarke, Sarah Barman, Paolo Remagnino[1], Ken Bailey, Don Kirkup, Simon Mayo, and Paul Wilkin[2]

[1] The Digital Imaging Research Centre, Kingston University, UK
{S.Barman, P.Remagnino}@kingston.ac.uk
[2] The Royal Botanic Gardens, Kew, UK
P.Wilkin@kew.org

Abstract. The work[1] investigates pattern recognition methods to detect venation patterns on leaves. An automated technique that involves scale space analysis and an automated technique that includes a combination of edge detectors are compared with a manual technique. A small data set of five images is considered in this initial exploratory work and the results are qualitatively evaluated. The results show that the technique involving scale-space analysis is demonstrated to be a promising research direction to pursue.

1 Introduction

A leaf needs to have three attributes to function effectively as the main photosynthetic organ of the plant, and thus the source of its nutrition. Firstly, it needs to have as large a surface area as possible to intercept light and absorb carbon dioxide. Secondly, it must have a means of allowing the carbon dioxide to enter and to control water loss (for temperature control). Finally, it needs a supply of water and basic nutrients. These raw materials are absorbed through the plant's roots and transported via the vascular system, or veins, through the stem and petiole (leaf stalk) to the leaf blade, where the veins ramify in an apparently hierarchical manner to form a network which supplies all parts of the blade. Evolution has acted upon this network to modify it in response to different environmental conditions and perhaps biotic factors like herbivory [1], [2]). Thus we now see a great diversity of forms in venation patterns of plant leaves [3]. There is a similar level of variation in the outline shape, thickness, texture and other characteristics of the leaf.

Botanists have been aware of this variation for centuries, and have used it as a comparative tool to study plants. For example, in traditional plant taxonomy (the study of classification), the approach used has been descriptive, with qualitative venation [4] and also shape and texture features separated into different categories, often on a subjective basis. This has been embodied in a complex and difficult terminology. However, traditional taxonomists have focused more

[1] The authors wish to acknowledge the assistance of Dr Elena Martnez-Prez and Professor Kim Parker for making their original scale-space code available.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 427–436, 2006.
© Springer-Verlag Berlin Heidelberg 2006

on floral and fruit morphology than leaf shape and venation because its variation was believed to be easier to interpret through being less influenced by environmental factors. An alternative explanation for this could be the inability of the human eye and brain to extract and process complex leaf features effectively. It appears that precise comparison within a statistical framework may be needed in order to do so successfully.

Leaf morphology is usually the most abundant type of data available in botanical reference collections (herbaria), and the easiest to obtain in field studies; the produc-tion of flowers and fruits occupies a relatively short part of the life of the plant, while leaves are present for most or all of its existence. Despite this, the continued use of a limited descriptive approach in the majority of studies means that leaf morphology is underexploited as a source of quantitative descriptors and discriminators. Although Fourier analysis, for example [5] has successfully been used for quantitative comparison of leaf outline shapes, there is still no way to quantitatively compare leaf venation patterns. Quantitative leaf feature information has many potentially powerful applications in comparative studies of plant diversity, ecology and physiology. The possibility of quantifying leaf venation has arisen at the same time as many herbaria initiating projects to digitize large numbers of specimens [6], presenting new opportunities for data mining. Perhaps most significantly, the quantification of leaf venation will help to automate plant identification - a very important aspect of plant systematics and conservation, medicinal plant use, vegetation and forestry inventory, and the study of fossil plants. In parallel with current plans to use gene sequence data as an identification tool ($DNA \quad Barcoding$ [7]), quantitative leaf venation data has the potential to revolutionise plant identification methods.

2 Previous Work

Researchers at the University of Florida [8] describe the preprocessing or skeletonization of a leaf so all of the laminar tissue of the leaf between the veins is removed, and all that is left is the vein network. The process uses chemical and mechanical methods to achieve the desired result. To further highlight the veins of the leaf, they can be immersed in a bleach and water mixture after clearing to whiten the veins. Once just the leaf venation structure remains, an image of it on a black background is very easy to turn into a binary image which will just show the venation of the particular leaf.

Another chemical technique involves dyeing either the laminar tissue of the leaf or the veins, so that there is more contrast between the two. Once this has been applied to a leaf, it is easier to threshold an image to extract the leaf skeleton. Some leaves have contrasting pigmentation in the veins and the rest of the lamina so this process is not necessary.

Ideally, we could apply the above methods to any leaf from which we needed an image of its vein structure. However, it is not plausible to do so because the skeletonisation or dyeing methods can only be applied to fresh leaves. Specimens

in herbaria are dried, can be hundreds of years old, and some species may only be represented by a single specimen with just one or a few leaves. A further significant problem is that both clearing and dyeing leaves is time- and labour-intensive. Thus it is not possible, and often not desirable, to apply these methods, even to recently collected herbarium specimens or fresh material.

Fu and Chi [9] produced a report which describes a two-stage process of leaf vein extraction using image processing. The first stage utilises segmentation to separate most of the background pixels (leaf lamina) from the foreground pixels (leaf veins). In the majority of cases, especially in fresh or air-dried herbarium specimen leaves which remain green, there is contrast between the leaf lamina and veins and this me-thod makes use of this. There is considerable variation in levels of contrast, and it can be poor in old or alcohol-treated herbarium specimens where the leaves tend to be a more uniform brown, so this method will work very well on some leaves and poorly on others. The segmentation works in stages. Initially, the edges of the whole leaf are identified using the Sobel edge detector to separate the leaf from the background. A *Laplacian* operator is used to calculate the second-order derivative of the pixels on the edges, and to identify edges within the leaf which are typically veins. Once the vein edges have been identified then the veins themselves can be located and the image thresholded in order to remove the unwanted parts of the image.

Once the image has been segmented the second stage of feature extraction begins. The segmentation gets rid of most of the non-vein elements of the lamina from the image; however some still remains and need to be eliminated; this is the job of the feature extraction. The *Sobel* Operator is used again to better classify the edge pixels of the veins and identify their gradients. A 7×7 mask looks at the neighbouring pixels around a particular pixel within a predicted vein area to determine if it is like its neighbours, in which case it is likely to be a vein pixel. The 7×7 neighbouring pixels around a root are used to calculate mean, standard deviation, maximum and minimum and compared against average values for the particular sample.

The method is successful to some extent. Most of the main veins are identified well but the fine tertiary veins are not as clear as the thicker primary and secondary veins and the lines are quite broken in places. The main disadvantage of this method is that even a small 640×480 image is quoted as taking as long as 20 minutes to process which is an unacceptably long time for most herbaria as they have many thousands or even millions of specimens to process; The Royal Botanic Gardens (RBG), Kew has 7.5 million herbarium specimens. The aims of the work described below are to investigate techniques that can be used to extract the venation pattern of a leaf without causing damage to it and to analyse the timescale of the algorithms used.

3 Methods

In this work, two methods were applied to analyse venation patterns in five plant taxa; fresh leaves of Ivy, Monstera, Nettle and Ribes and dried leaves of

Ochna[2]. These were (i) an application of an algorithm that includes a scale-space approach that has previously been shown to be useful on the detection of venation patterns on retinal images [10], (ii) an application of a simple smoothing and edge detection combination algorithm. Both methods were compared with a manually edited image of the venation pattern and the results of the three techniques were presented independently for evaluation to Kew botanists[3].

3.1 Algorithm Based Upon Scale-Space

An algorithm based upon scale-space analysis by Martinez et.al. [10] that was originally applied to retinal images, was adapted for use on images of leaves. The Martinez algorithm uses a segmentation process which identifies the blood vessels of the retina in the eye. It is feasible that this process can be adapted to identify the veins of a leaf because they have similar characteristics to blood vessels of the retina. Vessels in the retina tend to emanate from a single point much like those of a leaf, and there are also smaller vessels which branch from the larger vessels, similar to the pattern seen on leaves. These similarities occur because although a leaf and an eye are very different, the veins or vessels within them carry out similar functions of transporting water, oxygen, carbon dioxide and other materials to and from the cells.

Blood Vessel segmentation by using scale space analysis and region growing works using a different stages. Estes [11] explains that scale space analysis will modify an image whilst still maintaining the local structure such as edges which is important to image quality. Applying a scale-space analysis makes it easier for the computer to detect the details it is looking for, in this case the blood vessels. Marshall [12] explains that region growing works by selecting a seed pixel and comparing the pixels around it to see if it matches closely enough to the seed. If a pixel is a close enough match to the seed, then it is marked as part of the desired region. The pixels around this new pixel are examined to see if they match the seed pixel, and if they do they will be added to the selected region. This continues until no more pixels are added to the region. Once this occurs then a new seed pixel is selected until the entire region desired is identified.

Problems can occur with the region growing approach if a region being grown is conflicting with another region with a different seed. A poor choice of seed may give different results to other seeds, especially if a seed is located on an edge. To counte-ract these problems, different regions are able to grow simultaneously and no region can completely dominate the growth.

The scale-space analysis in this Martinez algorithm is achieved by convolving the original image with a Gaussian Kernel of different variances. This smoothes the image while keeping the large details intact. Different standard deviations are used for the Gaussian filter to allow for the different sizes of the blood vessels. The region growing examines the 8 pixels surrounding the seed pixel.

[2] In this paper, for simplicity, common English names of plants are adopted rather than their formal *Latin* names: *Hedera helix* (Ivy), *Monstera* sp., *Urtica dioica* (Nettle), *Ribes* sp. and dried leaves of *Ochna* sp.

[3] Don Kirkup (DK) and Simon Mayo (SM).

3.2 Smoothing and Edge Detection Algorithm

The most important part of the scale space algorithm is extracting the features (veins) from the leaf images. The difficulty in the technique is not the region-growing itself but automatically identifying the pixels which belong to a desired region. One advantage that extracting veins from a leaf image has over retinal vein extraction is that the samples tend to be uniform in colour and contrast. When taking an image of a retina, it is very difficult to uniformly light the eye and as some of the veins are deeper than others, the image contains some dark veins and some light veins. This does not tend to be the case in the leaf images as they are scanned and so they have constant illumination during the image capture process. The veins in most leaves may vary considerably in colour between species, but are usually concolorous within an individual plant.

As a result of the relatively consistent vein intensities in a leaf image it is possible that edge detection techniques could extract the veins from the leaf images. The techniques by Fu and Chi [9] rely heavily on edge detection techniques. Preliminary research of the application of various edge detectors to a smoothed leaf image showed that different edge detection algorithms will detect different veins, and areas of veins within a leaf image. It is proposed that the results of two or more edge detectors could be combined into a single image which will show the veins of a particular leaf sample.

The zerocross edge detector allows the application of a customised filter in its edge detection technique. From analysing the edge detection techniques the best technique for extracting the exterior edges of the veins was the Laplacian of Gaussian. The zerocross 9×9 filter was the best at extracting the vein itself rather than its edges. It also was better than other methods at extracting tertiary veins and omitting some of the unwanted leaf flesh details. The algorithm combined the output of the two edge detectors using addition.

4 Results

Two staff at RBG Kew (DK and SM) evaluated the results independently. The images were sent to the Kew testers in 5 JPEG files, each of which was stitched together from 4 different images: (i) the original unprocessed leaf image, (ii) an image created by the smoothing and edge detection algorithm, (iii) an image created by the scale space algorithm (iv) a control image which was the best effort at vein extraction using Adobe Photoshop in 10 minutes (per leaf). The original image is always in the top left, whilst the other 3 were placed randomly in the other spaces of the image. The images sent to Kew for analysis are Figures 1-2. The vein images are labelled A, B and C in a clockwise order as a reference for the Kew testers. Table 1, shows the reference to each image which details which venation extraction method the particular image was created from. This information was not available to the testers. Both testers rated the three samples of each leaf against the others on a $1-5$ scale, with higher scores representing more successful extraction in the opinion of the tester. These can be used for a quantitative evaluation, though due to the small sample size it is only of

Table 1. A summary of each figure with the relevant venation extraction method that has been applied (*Legend*: Adobe Photoshop (Adobe) and Smoothing and Edge (smooth edge))

Leaf	Fig.	Method	Leaf	Fig.	Method	Leaf	Fig.	Method
Ivy A	1	Adobe	Monstera A	2	smooth edge	Ribes A	5	Adobe
Ivy B	1	smooth edge	Monstera B	2	*scale space*	Ribes B	5	smooth edge
Ivy C	1	*scale space*	Monstera C	2	Adobe	Ribes C	5	*scale space*
Nettle A	3	smooth edge	Ochna A	4	Adobe			
Nettle B	3	Adobe	Ochna B	4	*scale space*			
Nettle C	3	*scale space*	Ochna C	4	smooth edge			

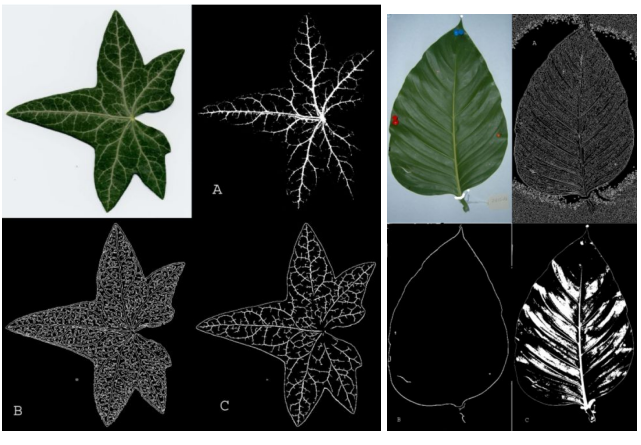


Fig. 1. *Left*: An image of the Ivy sample with different analysis output, *Right*: An image of the Monstera sample with different analysis output

limited use. The process with the highest score shows that it has been the most successful. Using the above scheme, the scores for each method are shown in Table 2(left).

Clearly the manual venation technique using Adobe Photoshop has, in the opinion of the testers provided the best venation extraction. As has been previously discussed, a quantitative evaluation is not a particularly fair one to do with so few samples so a qualitative evaluation will give a more detailed analysis of the comments of the testers for each of the leaf samples.

The Ivy sample produced good results and the testers both preferred the Adobe Photoshop version. The criticism of the smoothing and edge detection was that there were too many artefacts (unwanted leaf detail) present in the image. This could perhaps be fixed by a more efficient smoothing of the image. The scale-space algorithm was criticized here because not all of the veins were connected, and the primary veins (the seven large veins originating at the leaf blade/petiole junction) had the same thickness as the tertiary veins (those originating from the secondary branches of the primary veins).

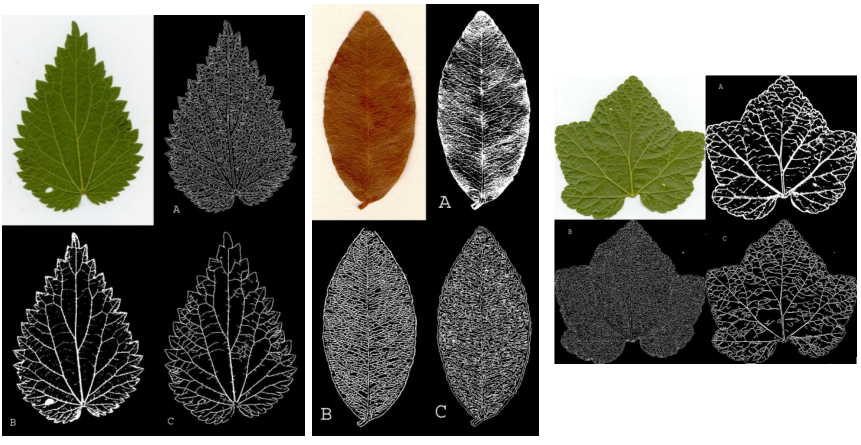


Fig. 2. *Left*: An image of the Nettle sample with different analysis output, *Middle*: An image of the Ochna sample with different analysis output *Right*: An image of the Ribes sample with different analysis output

The Monstera specimen was not scanned like the other samples but a digital photograph was used. Visually distinguishing the veins is not straightforward due to the relative lack of colour differentiation from the rest of the lamina, so it is understandable that all of the methods failed here.

The Nettle images were well received by the testers. The smoothing and edge de-tection was criticised because the smaller vein areas were too noisy; however the main (primary and secondary) vein extraction was quite successful. The Adobe Photoshop method was again chosen as the favourite and commended for maintaining the vein widths. The scale-space image was criticized for not maintaining the vein widths and lacking connectedness The connectedness could potentially be solved by changing the way which the seeding algorithm identifies the seed pixels.

In the analysis of the Ochna images, the edge detection technique was heavily criticised for producing unclear results. This dried leaf is a particularly difficult sample for this technique because the contrast between vein and leaf lamina is not very apparent. The Adobe Photoshop sample was commended for preserving vein orders, however it was heavily criticised for the unwanted white sections of the image. The scale-space sample was again criticized for not preserving vein widths but commended for finding nearly all the veins and it had very few unwanted artefacts.

The Ribes analysis showed that the smoothing and edge detection performed poorly here and was criticised for being visually confusing. It seems that in striving to obtain as much venation detail as possible that this method has fallen down by also picking up too much unwanted detail along with it. The Adobe Photoshop image was praised for having good vein widths but was criticized for lacking detail. The scale-space method was more successful here at maintaining vein widths and no major criticisms were reported about it. It does appear that some of the finer details are lacking in some areas.

Code was added to the beginning and end of the smoothing and edge detection and scale-space processes to determine how quickly the processes ran. The time taken to process an image will vary on different hardware. The implementation is in Matlab 7.01 on a 2.2GHz AMD processor with 1GB RAM, 256MB Graphics running Windows 2000 Professional. The results of the time testing for each of the test images is given in Table 2(right): The mean time for the smoothing and edge detection analysis is $2.5s$ The mean time for scale-space analysis is $40.2s$. Sixty seconds is considered to be a reasonable requirement for analysis of a leaf image. Only the Ribes image in the scale-space algorithm went over the 60 second requirement and the mean time taken for the scale-space algorithm was well below $60s$. All of the edge detection images were processed in well under $60s$.

Table 2. *Left*: A quantitative summary showing an indicative score of each method of analysis, *Right*: Time taken for different algorithms to run on different images

Method	**Score**
Adobe	27
scale space	19
smooth edge	14

Leaf	**Process**	**Time** (s)	**Leaf**	**Process**	**Time** (s)
Ivy	smooth edge	3.0	Ochna	smooth edge	1.2
Ivy	*scale space*	43.8	Ochna	*scale space*	9.0
Monstera	smooth edge	2.4	Ribes	smooth edge	3.6
Monstera	*scale space*	36.0	Ribes	*scale space*	62.4
Nettle	smooth edge	2.4			
Nettle	*scale space*	49.8			

5 Discussion and Conclusion

Following analysis of the comments made by the testers, the following section explains the problems with the smoothing and edge detection and scale space techniques and suggests ways in which they could be improved. Despite the manual Adobe Photoshop method providing the most highly rated results, it does not provide a viable option because it is not an automated venation extraction technique and is thus too time-consuming. It is possible to record a set of actions and apply them to images in a batch process within Photoshop. However, since the processes used for the 5 test images were all different this is not a viable option.

The criticisms of the smoothing and edge detection analysis were (i) too much unwanted detail/artefacts in the image (ii) main vein system not well separated enough from the rest of the leaf. These effectively amount to the same problem of the technique extracting too much unwanted information from the leaf images. To counter this problem and make the technique more useful, different smoothing algorithms could be tested which may further remove unwanted leaf detail. If a venation image of just the primary and secondary veins was required, it would be possible to smooth the tertiary veins out of the image along with the unwanted detail before running the edge detection techniques.

The criticisms aimed at the scale-space analysis were (i) there was little difference in vein thicknesses in the vein images, so that tertiary veins were the same thickness as primary veins, and (ii) veins were often not connected to one another. It was expected that the scale space technique would not perform perfectly on vein images as it was not fully optimised for this purpose. The process is promising and it could be significantly improved by changing the seeding process to pick up more vein seeds. The program performed remarkably well on the veins it did select and there was little evidence of unwanted leaf detail being selected by the process. In addition to changing the seeding process, the region-growing process could be further optimised to the pattern and style of veins in a leaf image. This technique provides the best opportunity for further research.

The evaluation has produced mixed results. The images created by the programs show potential in different areas, especially scale-space in leaves with veins of con-trasting colour to the rest of the lamina. However, more development of both methods would be required in order to produce a system which a Herbarium could regularly and reliably use. Many more samples will need to be tested and calculations of sensitivity and specificity against validated ground truths will need to be made in order to produce a complete evaluation and this is planned for future work.

References

1. Chase, M., Stevenson, D., Wilkin, P., Rudall, P.: Monocot systematics: a combined analysis. In Rudall, P., Cribb, P., Cutler, D., Humphries, C., eds.: Monocotyledons: systematics and evolution. RBG, Kew, UK (1995) 685–730
2. Niklas, K.: The evolution of leaf form and function. In Marshall, B., Roberts, J., eds.: Leaf development and canopy growth. Sheffield Academic Press, Sheffield, UK (2000) 1–35
3. Klucking, E.: The classification of leaf venation patterns. Volume 7 of Leaf venation patterns. Gebr. Borntraeger Verlagsbuchhandlung (1995)
4. Niklas, K.: A revised classification of the architecture of dicotyledonous leaves. In Metcalfe, C., Chalk, L., eds.: Anatomy of the Dicotyledons: Systematic Anatomy of Leaf and Stem, With a Brief History of the Subject. Volume 1. Clarendon Press, Oxford, UK (1979) 25–39
5. Jensen, R., Ciofani, K., Miramontes, L.: Lines, outlines and landmarks: morphometric analyses of leaves of acer rubrum, acer saccharinum (aceraceae) and their hybrid. Taxon **51** (2000) 475–492(8)
6. The Royal Botanic Gardens, Kew, U. (www.kew.org/data/herb_digitisation.html)
7. Schindel, D., Miller, S.: Dna barcoding a useful tool for taxonomists. Nature **435** (2005) 17
8. IFAS: Ifas extension, drying and preserving plant materials. `http://edis.ifas.ufl.edu/EP004` (2006) University of Florida.
9. Fu, H., Chi, Z.: A two-stage approach for leaf vein extraction. In: IEEE International conference on neural networks and signal processing. Volume 1. (2003) 208–211
10. Martnez-Prez, M., Hughes, A., Stanton, A., Thom, S., Bharath, A., Parker, K.: Retinal blood vessel segmentation by means of scale-space analysis and region growing. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Volume 1679. (1999) 90–97

11. Ford, G., Estes, R., Chen, H.: Space scale analysis for image sampling and interpolation. In: International Conference on Acoustics, Speech, and Signal Processing. Volume 3. (1992) 165–168
12. Marshall, A.: Computer Vision, Models and Inspection. Robotics and Automated Systems. World Scientific Publishing (1993)

A Mobile Low-Cost Motion Capture System Based on Accelerometers

Jan-Phillip Tiesel[1] and Jörn Loviscach[2]

[1] Universität Bremen
[2] Hochschule Bremen

Abstract. Low-cost accelerometers can be employed to create a motion-capture solution for below US$ 100. It may be used in mobile settings employing a portable digital recording device to capture the analog data of 15 degrees of freedom. The solution is integrated with standard 3D animation software. We introduce methods to extract and tweak kinematical as well as timing data from these acceleration sensors, which are attached to an actor's limbs. These methods take care of the fact that the measured acceleration data alone can neither provide complete nor accurate information to satisfactorily reconstruct the captured motion. Particular emphasis is placed on the ease of use, in particular concerning the calibration of the system.

1 Introduction

To create believable and lifelike motion manually by keyframing poses requires highly skilled animators and generally consumes hours of work for seconds of animation. In contrast to that, motion capture technology is theoretically able to provide such data in real time. However, commercial motion capture systems are too expensive for small film or game studios, not to speak of amateurs. These systems must be operated by trained professionals and usually take an hour or longer to be set up and calibrated, which renders them essentially immobile.

Inertial sensors in the form of micromachined accelerometers and miniaturized gyroscopes are widely available today. In order to fathom the capabilities of an ultra low-cost approach, we decided to exclusively use accelerometers as sensors. Note that whereas mathematically one could integrate the acceleration data twice to uncover position data, practically such an integration would blow up the data within seconds due to tolerances and noise.

This work demonstrates two approaches to how sensor data of accelerometers can be interpreted and edited in a standard animation software package:

1. The orientation of bones is estimated using acceleration sensors that are attached to an actor's limbs. By using the fixed direction of gravitational acceleration, profound assumptions on the posture of body segments can be made if the actor acts out the motion with no jerky movements.
2. The magnitude of a sensor's acceleration is used to retime existing keyframe animation.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 437–446, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Both approaches can be combined to first capture the orientation in slow-motion and then capture the timing in a second pass done at actual speed. As the interpretation of the accelerometer data will most likely contain errors and will not yield an accurate and complete description of the initial motion, we equip the user with ways to refine and augment the recorded data.

The overall capturing process can be broken down into five steps:

1. Accelerometer signal range calibration (once in the lifetime of a sensor)
2. Calibration of the orientation of the accelerometers relatively to the limbs (once per capturing session)
3. Real-time capture of estimated limb orientations (only slow-motion movements to minimize errors due to manual acceleration)
4. Smoothing of capture data and creation of the kinematical chain representation in the animation software
5. Re-enacting the previous motion with proper timing to retime the kinematical data (only if fast movements occurred during step 3)

The main contributions of this paper to the state of the art in motion capture are the mobile low-cost motion capturing system, the easy-to-use calibration methods, the drift-free and robust user-assisted interpretation of orientation data, and the retiming process assisted by acceleration data.

This paper is structured as follows: Section 2 provides an overview of related work. Section 3 introduces the special hardware we created. The calibration of the system is covered in Section 4. Section 5 describes the process employed to compute orientation data and apply them to a 3D model. How acceleration data can be employed to retime animations is treated in Section 6. Section 7 presents results; Section 8 concludes this paper.

2 Related Work

An overview of the research in vision-based motion capturing is given by Moeslund and Granum [1]. Fewer cameras or alternative low-dimensional controllers have been successfully used to synthesize new animations from existing motion capture data [2,3]. Whereas they succeed in creating realistic motion representations, they lack the ability to operate low-cost solutions with unlimited measurement volume. The decreased complexity in the hardware setup often leads to an increase in the complexity of system operation (e. g., the need for building an extensive motion capture database before the capturing process). The overall price of the used hardware could be lowered but is still in the range of several hundred to several thousand US$.

Mechanical motion capture employs sensors attached to the actor's body to estimate joint angles. Those systems are free of occlusion effects and often utilize wireless data transfer to permit mobile applications. While the *ShapeWrap*™ system by Measurand Inc. uses a flexible tape that bends in accordance with the actor's limb movements, inertial sensors are used by Animazoo in their *Gypsy-Gyro* suit. Both manufacturers state that the setup and calibration time as well

as the essential user training are minimal compared to vision-based solutions. Up to date those systems are not available for less than US$ 20,000 and US$ 60,000, respectively.

Zhu and Zhou [4] used MARG sensors (measuring *m*agnetic field, *a*ngular *r*ate and *g*ravity) to determine rigid body orientation. An extensive amount of research was done using inertial sensors for medical applications like gait analysis [5]. Mayagoitia et al. [6] presented a combination of accelerometer and gyroscope measurements to obtain kinematic parameters such as knee angle or thigh linear acceleration. Luinge and Veltink [7] investigated the maximum amount of information that can be extracted from a tri-axis accelerometer.

Hansson et al. [8] pointed out two of the major limitations of orientation estimation by the exclusive use of accelerometers. First, rotation about the line of gravity cannot be detected; second, the ability to measure rotations about the object's axes depends on its overall orientation. The work of Giansanti et al. [9] indicated that the reconstruction of both orientation and position from only accelerometer data does not yield feasible results. Rehbinder and Hu [10] proposed fusing data from rate gyroscopes and accelerometers to give long-term, drift-free attitude estimates. A combination of computer vision motion tracking and inertial tracking using the Kalman filter was presented by Welch [11].

One of the main challenges that animators face is getting the timing of the overall motion to be as lifelike and believable as possible. Terra and Metoyer [12] presented an approach that allows the user to draw a sketch of the desired motion using the proper timing. They then used the timing information of the user's sketch to retime an existing keyframed animation. Thorne et al. [13] implemented an animation system based entirely on sketching out the motion and using the kinematical and temporal information to reproduce the completed motion sequence.

3 Sensor Hardware and Data Flow

The sensor hardware (see Figure 1) consists of a battery-powered central circuit board wired to five boxes; each box contains an accelerometer chip LIS3L02AQ from STMicroelectronics. This micromechanical device measures acceleration along three perpendicular axes and outputs three corresponding voltages.

The central circuit board contains a clock generator chip (NE 555), a counter (MOS 4024), and an analog 16:1 multiplexer (MOS 4067). The 15 output voltages of the accelerometers are connected to the inputs of the multiplexer. A reference voltage, stabilized by an operational amplifier (one quarter of RC 4136), is connected to the 16^{th} input. The timer and the counter are used to continuously cycle the multiplexer's output through the 16 inputs, so that each appears 60 times per second.

The output is decoupled through another operational amplifier (one quarter of RC 4136) and fed into the left audio input channel of the PC's sound card, see Figure 2. The most significant (i. e., most slowly varying) bit of the counter is fed into the right channel. On the PC, our software synchronizes with this 60 Hz

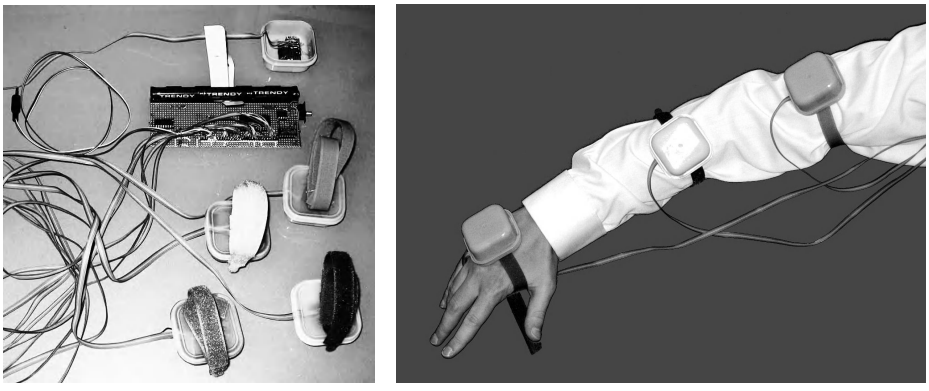


Fig. 1. The hardware interface consists of five tri-axis accelerometers and a main board connectable to an analog stereo audio input

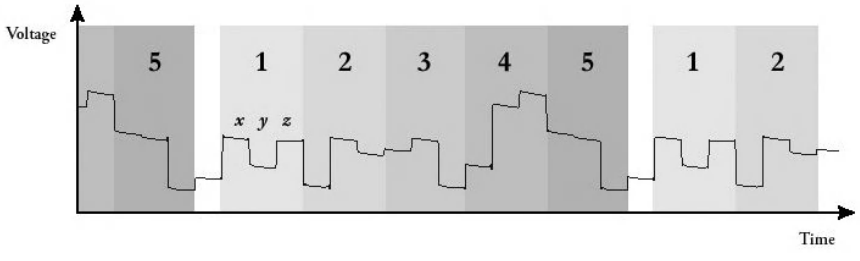


Fig. 2. The analog signal transmitted via one audio channel comprises the xyz values of the five sensors and a reference voltage

signal and processes the switched signals from the left channel in real time. Per cycle, the software forms 16 time slots over which the signal is averaged. This helps to suppress the overshooting and ringing transients introduced by the Nyquist low-pass filter of the sound card. As standard PC sound cards do not reproduce DC voltages, we recover them by forming differences with the measurement for the reference voltage, which forms the 16^{th} input.

Optionally, the analog output signal can be recorded by a portable audio recording device such as an MP3 player that has recording capabilities for uncompressed wave signals. The data thus stored can be transferred to a computer after the capturing process is completed.

A stand-alone .NET-based software is responsible for the initial processing of the input data. During the real-time processing, the data are saved to a file that also holds general information about the capture process such as sampling rate, number of active sensors, and sensor naming. To import, edit and utilize the captured data with a standard 3D software package, a library of procedures was created in MEL, the scripting language of Autodesk Maya®. It provides the user with a graphical interface that is fully integrated with the animation package.

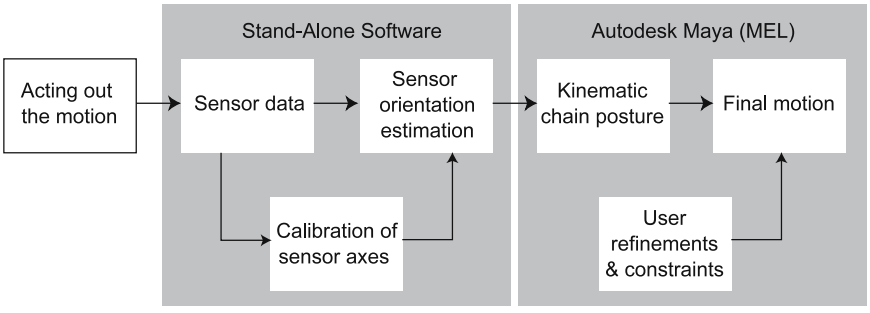


Fig. 3. A stand-alone data processor delivers the recorded motion to a set of MEL scripts

4 Calibration

Several types of coordinate frames occur within the system: There is a global one, the z axis of which points away from the earth's center; a coordinate frame for every limb or bone with its origin placed at the pivot point and its x axis pointing along the bone; and a coordinate frame for every sensor, the xyz axes of which are aligned to the sensor's ones, see Figure 4. Most of the computations deal with conversions between these three systems; the objective is to determine the relation between the limbs' systems and the global one. We use the subscripts "global," "limb," and "sensor" to specify the reference frame of a given vector quantity.

The accelerometers display combined offset and range errors of about 10 percent of earth's gravity. A huge part of this error can be corrected using a calibration step to be executed once in the lifetime of an accelerometer: The lowest and highest emerging values of the xyz components of the measurement are

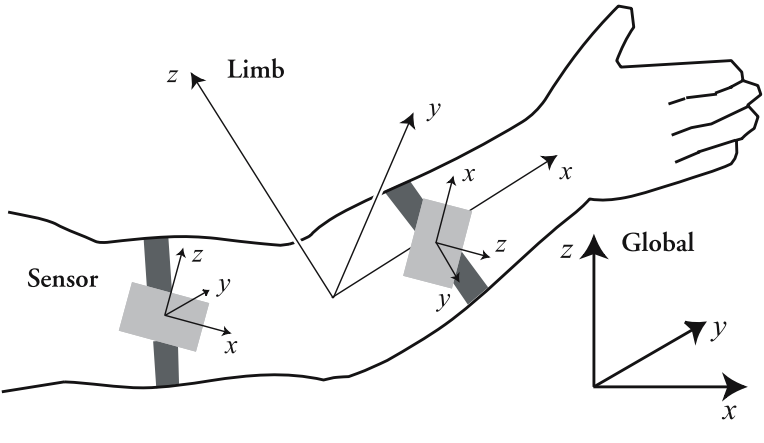


Fig. 4. Three types of reference frames are used in the computation

established while the user slowly rotates the sensor about all of its axes. Short peaks are suppressed to cancel quick motion and only capture the acceleration due to gravity. The resulting minimum and maximum values per component are used to linearly normalize the measurement range. After that, measurements done with a static sensor will yield values from the interval $[-1, 1]$, where 1 corresponds to a perfect alignment of the corresponding axis with the direction of gravity, that is, the negative direction of the global z axis.

The coordinate frames of the sensors typically do not match those of the limbs to which they are attached. This discrepancy requires a calibration that is to be executed every time the sensors are mounted. One has to establish a rotation matrix $R_{\mathrm{l}\leftarrow\mathrm{s}}$ that transforms the sensor's axes to the limb's axes.

To understand how this rotation matrix can be found easily consider a sensor that is attached to an actor's lower arm. Measure the sensor data when the actor extends his arm horizontally and vertically one after the other. Let $\boldsymbol{a}_{\mathrm{sensor}}$ and $\boldsymbol{b}_{\mathrm{sensor}}$ be the corresponding results, corrected to become perpendicular unit vectors, what they should be in theory. The desired rotation matrix $R_{\mathrm{l}\leftarrow\mathrm{s}}$ transforms these vectors into the corresponding gravity vectors in the limb frame: $(0, 0, -1)^{\mathrm{T}}$ and $(1, 0, 0)^{\mathrm{T}}$, respectively. This matrix can be computed as

$$R_{\mathrm{l}\leftarrow\mathrm{s}} = \left[\boldsymbol{b}_{\mathrm{sensor}}, -\boldsymbol{a}_{\mathrm{sensor}} \times \boldsymbol{b}_{\mathrm{sensor}}, -\boldsymbol{a}_{\mathrm{sensor}}\right]^{\mathrm{T}}, \tag{1}$$

where the matrix on the right hand side is formed from the three given column vectors.

The data gained in the calibration phase can be used to convert every acceleration measurement from the corresponding sensor frame to the limb frame. Note that since we are dealing with derivatives, we need not bother about the placement of the origin of a frame and are only concerned with its orientation.

5 Estimating and Using Limb Orientation Data

The main task in the motion capture process is to use the acceleration data given in the limb systems to determine the orientation of these systems within the global system. In this mode, we expect the actor to move slowly, so that mostly the acceleration due to gravity will be measured. This way, the accelerometers can be used as inclinometers. Even if the actor applies an acceleration of half the acceleration of gravity, the error in orientation is limited to 30 degrees and will mostly be about 15 degrees. On top of that, an acceleration that large cannot be sustained in linear direction for more than half a second, due to the limited position range of the body's limbs.

If one assumes slow motion, the calibrated measurement output of a sensor equals $\boldsymbol{a}_{\mathrm{limb}}$ and has approximately a length of one. To adjust the limb in the 3D software, we're seeking a matrix $R_{\mathrm{g}\leftarrow\mathrm{l}}$ that transforms the global orientation into that of the limb. In particular, $R_{\mathrm{g}\leftarrow\mathrm{l}}$ has to map $\boldsymbol{a}_{\mathrm{limb}}$ to the global direction of gravity $(0, 0, -1)^{\mathrm{T}}$.

There is an infinite number of rotations with this property. This stems from the fact that static rotations about the global z axis do not change the gravity

field. Thus, we aim to find the simplest possible rotation and offer the user a control on the remaining single degree of freedom. The simplest rotation is the (unique) one with the shortest turning angle. Its axis is perpendicular to both vectors $\boldsymbol{a}_{\text{limb}}$ and $(0, 0, -1)^{\text{T}}$ and thus can be found through their cross product. Its angle ϕ can be determined through

$$\phi = \operatorname{atan2}\left(\sqrt{a_x^2 + a_y^2}, a_z\right). \tag{2}$$

Typically, the sensors used during recording are part of a kinematical chain. By creating a skeleton rig according to the user's specifications, the connections of limbs and their according sensors are defined. The individual position of the root joint(s) and the length of every bone can be easily edited with the user interface provided by our scripts.

The aim constraint functionality of the Maya software is used to transfer the captured orientation information to the individual limbs. The first column of every matrix $R_{\text{g}\leftarrow\text{l}}$ specifies the global orientation of the corresponding limb's x axis. Adjusting these vectors by the limb's length and adding them along the kinematical chain provides the targets required as aim constraints.

Due to the choice of the simplest rotation, there is not yet a rotational component about the global z axis. To compensate for this, the user can manually animate the joint's rotation about the global z-axis. This is easily accomplished by concatenating this z rotation with $R_{\text{g}\leftarrow\text{l}}$.

6 Animation Retiming

In a second approach that can be used independently or in combination with the presented method of orientation estimation, we utilize the acceleration magnitude of a sensing unit to retime an existing motion. This can be either a hand-keyframed motion or a motion sequence previously captured by the accelerometers. The overall timing of the actor's motion is applied by fitting the acceleration curve of a given animated object over time to the measured acceleration information of the sensor(s). The instances of time where the acceleration reaches peak values serve as references for the fitting process.

The user chooses a keyframed 3D object. Its acceleration $d^2\boldsymbol{x}(t)/dt^2$ is determined numerically from its position $\boldsymbol{x}(t)$. Our MEL script determines peaks by looking at the slope of $|d^2\boldsymbol{x}(t)/dt^2|$ and inserts "peak markers" at those times. The script proceeds similarly for the measured acceleration data $\boldsymbol{a}_{\text{sensor}}$: It places peak markers at times where

$$\left|\,|\boldsymbol{a}_{\text{sensor}}| - 1\,\right| \tag{3}$$

attains a locally maximal value. Note that no calibration of the sensors to the limbs is necessary since we are only interested in the magnitude of the acceleration. The subtraction of 1 approximately compensates for the fact that the measured acceleration data include gravity, in contrast to the acceleration data from the object's keyframed motion in the 3D software.

The user can manipulate the established peaks of both curves independently using the keyframe editing tools supplied by the 3D animation software. Once the peaks of the two curves are put into a one-to-one correspondence, the script can redistribute all of the object's keyframes using linear interpolation. Keyframes situated at acceleration peaks of the original motion curve are placed at the time of the corresponding peak of the sensor data curve.

7 Results

The calibration procedure for establishing the rotation matrices for the individual sensors is easy and fast. It typically takes just a few seconds per limb or even less if several sensors are part of the same kinematical chain and can be calibrated simultaneously (e.g., upper arm, forearm, hand).

We evaluated the system's precision using two different kinds of measurement. After an accurate calibration which secured perpendicular calibration vectors, the inclination angle of the estimated sensor orientation was compared to a geometric measurement. This method showed a maximum error of 4° with the average error being smaller than 2°. The second method included manual and therefore less precise calibration by the actor wearing the sensor. This time, the estimated inclination angle was compared to angle measurements established by visual reference (digital photographs taken during the capturing process). As expected, the manual calibration resulted in higher error values. Regarding the quick and rudimentary calibration, the results are satisfying, yet amendable (average error less than 10° and maximum error less than 15°).

Several well-defined motion sequences of the upper torso with sensors attached to the actor's arm, forearm, and hand were executed and captured. The resemblance of the physical motion to the synthesized representation depended largely on the amount of rotation about the axis of gravity. No movements can be captured while limbs are moved on a plane perpendicular to the line of gravity (e. g., on a table), what requires a lot of user intervention to supply the missing rotation data. In contrast, movements along a vertical plane (e. g., swing of arms during walking, arm waving, arm reaching out for a glass of water, legs walking) are reproduced with good accuracy, see Figure 5.

Motion that implies high acceleration (e. g., ground contact during walking; fast or sudden movements in sports or action sequences) have to be acted out in slow motion. As no drift or offset error is introduced during periods of high acceleration, the user can either retime the captured slow motion sequence or act out the fast motion and fix kinematical errors that occur due to acceleration peaks in the Maya software.

The animation retiming method works reliably if the captured motion implies the occurrence of well-defined acceleration peaks (e. g., waving). Often, the actor has to exaggerate his movements in order to produce clearly identifiable acceleration peaks. In addition, the approach implies that the original keyframes need to have proper relative timing between acceleration peaks as the keyframes are distributed using linear interpolation.

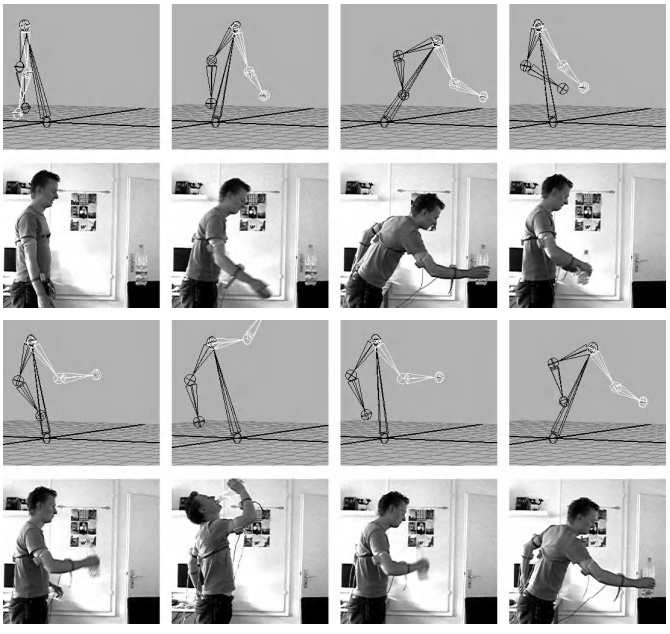


Fig. 5. Motion with vertical components can be captured well, often at its actual speed. The white arm in the 3D view is the one that is closer to the camera.

8 Conclusion and Future Work

We have presented a motion capture system that requires only inexpensive special hardware and can easily be used in mobile settings. The system is robust in that errors do not accumulate in time or along kinematical chains. However, the usefulness of the orientation measurement depends largely on the type of motion to be captured.

An obvious solution to the indeterminacy of the rotation about the z axis would be to bundle each accelerometer with a tri-axis electronic compass and fuse the data of both sensors more or less intelligently. The compasses would raise the system cost significantly (by around US$ 60 per sensor).

Ways to improve the orientation sensing with or without compasses range from data filtering [14] to statistical approaches that determine the most likely orientation given past (and, if recorded, future) sensor data as well as typical motion sequences. Such an approach may borrow many ideas from visual motion capture techniques such as [2].

In order to improve the presented animation retiming method, a curve matching algorithm such as Bruderlin's dynamic time warping [15] could be employed instead of redistributing the existing keyframes in a linear fashion.

References

1. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding: CVIU **81** (2001) 231–268
2. Chai, J., Hodgins, J.K.: Performance animation from low-dimensional control signals. ACM Transactions on Graphics **24** (2005) 686–696
3. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. In: SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, New York, ACM Press (2002) 491–500
4. Zhu, R., Zhou, Z.: A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. IEEE Transactions on Neural Systems and Rehabilitation Engineering **12** (2004) 295–302
5. Sabatini, A.M.: Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis. Medical and Biological Engineering and Computing **43** (2005) 94–101
6. Mayagoitia, R.E., Nene, A.V., Veltink, P.H.: Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. Journal of Biomechanics **35** (2002) 537–542
7. Luinge, H.J., Veltink, P.H.: Inclination measurement of human movement using a 3-d accelerometer with autocalibration. IEEE Transactions on Neural Systems and Rehabilitation Engineering **12** (2004) 112–121
8. Hansson, G., Asterland, P., Holmer, N., Skerfving, S.: Validity and reliability of triaxial accelerometers for inclinometry in posture analysis. Medical and Biological Engineering and Computing **39** (2001) 405–413
9. Giansanti, D., Macellari, V., Maccioni, G., Cappozzo, A.: Is it feasible to reconstruct body segment 3-D position and orientation using accelerometric data? IEEE Transactions on Biomedical Engineering **50** (2003) 476–483
10. Rehbinder, H., Hu, X.: Drift-free attitude estimation for accelerated rigid bodies. Automatica **40** (2004) 653–659
11. Welch, G.F.: Hybrid self-tracker: An inertial/optical hybrid three-dimensional tracking system (1995)
12. Terra, S.C.L., Metoyer, R.A.: Performance timing for keyframe animation. In: SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, New York, ACM Press (2004) 253–258
13. Thorne, M., Burke, D., van de Panne, M.: Motion doodles: an interface for sketching character motion. ACM Transactions on Graphics **23** (2004) 424–431
14. Luinge, H.J., Veltink, P.H.: Measuring orientation of human body segments using miniature gyroscopes and accelerometers. Medical and Biological Engineering and Computing **43** (2005) 273–282
15. Bruderlin, A., Williams, L.: Motion signal processing. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, New York, ACM Press (1995) 97–104

Fusing Individual Algorithms and Humans Improves Face Recognition Accuracy

Alice J. O'Toole, Fang Jiang, Hervé Abdi, and P. Jonathon Phillips*

The University of Texas at Dallas
*National Institute of Standards and Technology

Abstract. Recent work indicates that state-of-the-art face recognition algorithms can surpass humans matching identity in pairs of face images taken under different illumination conditions. It has been demonstrated further that fusing algorithm- and human-derived face similarity estimates cuts error rates substantially over the performance of the best algorithms. Here we employed a pattern-based classification procedure to fuse individual human subjects and algorithms with the goal of determining whether strategy differences among humans are strong enough to suggest particular man-machine combinations. The results showed that error rates for the pairwise man-machine fusions were reduced an average of 47 percent when compared to the performance of the algorithms individually. The performance of the best pairwise combinations of individual humans and algorithms was only slightly less accurate than the combination of individual humans with all seven algorithms. The balance of man and machine contributions to the pairwise fusions varied widely, indicating that a one-size-fits-all weighting of human and machine face recognition estimates is not appropriate.

1 Introduction

Face recognition algorithms have improved dramatically over the last decade and are now available commercially for security applications.The most common applications involve face verification, where a presented image of a person must be compared to a stored representation and be verified or rejected as an identity match. To achieve this task, algorithms produce an estimate of the likelihood that two images are of the same person. A match criterion is then set for making the decision.

According to a recent US Government sponsored test of state-of-the-art face recognition algorithms [1,2], current algorithms are impressively accurate at this task when the images to-be-matched are taken under controlled illumination conditions. Specifically, the Face Recognition Grand Challenge (FRGC), conducted at the National Institute of Standards and Technology (NIST) between 2004 and 2006, tested 17 algorithms on a task of matching face identity in roughly 128 million pairs of images taken under controlled illumination. The results showed an average verification rate of .91 at the .001 false acceptance rate.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 447–456, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In an analogous FRGC test of algorithms on the task of matching face identity in images taken under different illumination conditions, however, the performance of algorithms was less impressive. Only seven algorithms volunteered to participate in this experiment. These systems scored an average verification rate of only .42 at the .001 false acceptance rate. This suggests that current face recognition algorithms may not be ready for application environments that have natural variations in illumination. These kinds of environments are typical in airports and other public places where some part of the illumination comes from natural light, (e.g., sunlight filtered through windows).

How well must a face recognition algorithm perform to be useful for a security application? Although human performance on face recognition is often considered the standard to which algorithms should aspire, there are few direct comparisons between humans and algorithms. A recent exception to this general rule is a study comparing the performance of humans with algorithms competing in the uncontrolled illumination experiment of the FRGC [3]. In that study, "easy" and "difficult" face pairs were sampled from the FRGC test set, using a control algorithm based on principal components analysis (PCA) of the aligned and scaled images. Human subjects rated the likelihood that the pairs of face images were of the same person. ROC curves computed for the humans and for the algorithms revealed that three algorithms [4,5,6] performed more accurately than humans on the difficult face pairs. Nearly *all* algorithms performed more accurately than humans on the easy face pairs. Post-hoc analyses of the human subject data gave no indication that subject attention waned toward the end of the experiment. Moreover, both humans and algorithms were more accurate on face pairs prescreened by the PCA to be easy than on the face pairs prescreened to be difficult. The results indicate, therefore, that although algorithms appear to perform poorly on the task, they are nonetheless competive with the performance of human subjects.

The comparison between algorithms and humans on the face matching task gives an indication of the quantitative ranking of algorithms by accuracy, but does not offer any information about how similarly humans and algorithms perform the task. To gain insight into the qualitative aspects of the performance of humans and algorithms, and to see if performance could be improved, a fusion study of the algorithms and humans was carried out [7]. If algorithms and humans employ different recognition strategies, it should be possible to fuse the their estimates of face similarity to improve performance. The fusion was carried out in two parts. In the first part, the similarity estimates of the seven algorithms from the FRGC were fused with partial least squares regression (PLS)[8,9] and used to predict the match status of individual pairs of faces. The robustness of the PLS was tested with cross-validation. The results indicated that fusing the algorithms cut the error rate of the best-performing algorithm by a factor of two [7].

The second part of the study examined whether algorithm performance could be improved by fusing human similarity estimates with the estimates of the seven algorithms. Specifically, human similarity estimates for the difficult face pairs [3]

were averaged across 49 subjects and fused with the algoritms estimates using PLS. This reduced error rate to near perfection and indicated that human face recognition strategies differ sufficiently from algorithms to make a substantial contribution to recognition performance through fusion.

Although it is of general interest to know that humans and algorithms can be fused to increase face matching accuracy, it is of more practical value to know how to combine *individual* algorithms with *individual* humans. Which algorithms combine most beneficially with which humans? Can a general rule be established, or do humans and algorithms differ sufficiently in recognition strategy to suggest that individual human-machine fusions be done on a case-by-case basis. The purpose of the present study was to explore the benefits of fusing individual humans with indidvidual algorithms. This is of practical value given that in most real-world applications, one algorithm works under the supervision of a single human operator. We assessed the suitability of particular man-machine combinations using a fusion approach.

We approached this problem in two parts. First, we fused the similarity scores produced by the seven available algorithms and individual humans. This provides data on the best-case scenario, (i.e. where humans can benefit from all available algorithm expertise). This fusion is similar to the one described previously [7], with the exception that humans, in this case, are treated as *individuals*, rather than being represented by their global average. Next, we fused individual human subjects with individual algorithms to assess performance of particular man-machine hybrids.

2 Methods

The methods for this study make use of human subject and FRGC algorithms' estimates of face pair similarity that were collected previously [3]. For completeness we provide a sketch of the methods used to collect the relevant data in that study and then proceed to describe the fusion methods applied in the present work.

2.1 Stimuli

The face images used to test algorithms in the face matching task were drawn from a database developed for the FRGC [1]. Face pairs consisted of a target image, taken under controlled illumination conditions, and a probe image, taken under uncontrolled illumination conditions, (e.g., in a corridor). Target images had a resolution of 1704 x 2272 pixels and probe images had a resolution of 2272 x 1704. A sample face pair appears in Figure 1.

To make the task as challenging as possible, we sampled the images from a homogenous face population that included only male and female Caucasians in their twenties and thirties. Each face pair was matched by sex. Combined these constraints eliminated the possibility that humans could base identity judgments on surface facial characteristics associated with sex, race, or age.



Fig. 1. An example non-match pair with the probe image on the left and the target probe image on the right

In the present study, we used the 120 difficult pairs of faces (60 male and 60 female) presented to subjects in the previous study [3]. Half of the face pairs were of matched identity, (i.e., the target and probe images were of the same person), and half were non-match pairs (i.e., the target and probe images were of different people). As noted, face pairs were prescreened by a PCA of the aligned and scaled images before sampling for the human experiment. Difficult match pairs had similarity scores that were less than two standard deviations below the match mean, (i.e., two dissimilar pictures of the same person). Difficult non-match pairs had similarity scores greater than two standard deviations above the non-match mean, (i.e., similar images of two different people).

2.2 Human Estimates of Face Similarity

Humans judged the similarity of the 120 face pairs by rating each pair as follows, "1.) sure they are the same person; 2.) think they are the same person; 3.) don't know; 4.) think they are not the same person; 5.) sure they are not the same person." The ratings of 49 subjects on the 120 face pairs served as the human input data for the fusion analysis.

2.3 Algorithm Estimates of Face Similarity

To participate in the FRGC uncontrolled illumination experiment, algorithm developers were asked to compute a matrix containing similarity scores between all possible pairs of 16,028 target images and 8,014 probe images. The resulting matrix, therefore, contained 128,448,392 similarity scores, each representing the likelihood that a target and probe images in the pair were of the same person. These matrices were scored by NIST and complete performance results for the algorithms are available elsewhere [1,2]. For present purposes, the similarity scores for the 120 difficult face pairs presented to subjects were extracted from

each of seven participating algorithms' similarity matrices. These scores served as the algorithms' input to the fusion analysis.

2.4 PLS Fusion

Fusion was performed by partial least squares (PLS) regression, a statistical technique that generalizes and combines features from principal component analysis and multiple regression [8,9]. The technique is used to predict a set of dependent variables from a set of independent variables (predictors). Though less known in the pattern recognition literature, PLS is widely used in chemometrics, sensory evaluation, and neuroimaging data analysis, (cf. [9]).

The predictors for the present study were the human- and algorithm-generated similarity scores for the 120 pairs of face images. We define this more specifically in the context of particular fusion analyses. The dependent variable was the match status of the face pair (i.e., same versus different person). with match pairs assigned a value of 1 and non-match pairs a value of -1. PLS regression gives a set of orthogonal factors, sometimes called latent vectors, from the covariance matrix of predictors and dependent variables. These can be used to predict the dependent variable(s), by appropriately weighting the predictors. This set of weights is called $\mathbf{B}_{\text{pls}}$in the PLS-regression literature [8,9].

The predictive power of a PLS solution is assessed generally with cross-validation techniques such as a bootstrap or jackknife procedure. All factors, or only a subset of them, can be used to compute the prediction of the dependent variable(s), which are obtained as a weighted combination of the original predictors given by $\mathbf{B}_{\text{pls}}$. The larger the number of factors kept, the better the prediction of the "learning set" but, in general, a smaller number of factors is optimal for robust prediction (i.e., for test set predictions).

3 Results

3.1 Baseline Human Performance

We first assessed the baseline performance of individual humans in a way that is comparable with the output of the fusion test. Accuracy in the fusion test is given as the error rate for match status classifications determined in a jack-knife procedure. As noted, humans rated each pair of faces on a 5-point scale varying from "sure the same person" to "sure different people". Given that one of the possible human responses was "don't know", we computed the number of errors in two ways by: a.) assigning the "don't know" to a match response; and b.) assigning the "don't know" response to a non-match response. We computed the number of errors for each subject in both ways and averaged these two values. Across all subjects, the human error rate averaged .141 (see column one of Table 1), indicating good, but not perfect performance.

Table 1. The error rate information for individual human performance appears in the first column. Analogous information appears in the second column for the fusion of individual humans combined with the seven algorithms. The data for the best-paired individual human-algorithm fusions appear in the third column.

Error rate	Human	Human+7 Algorithms	Human+Algorithm Best-paired Fusions
average	.141	.057	.078
minimum	.041	.033	.033
maximum	.258	.108	.117

3.2 Fusing Individual Humans with All Algorithms

Next, we fused each individual with all seven algorithms using PLS. The purpose of this analysis was to assess the relative importance of algorithms when combined with individual humans. We also use the performance of this simulation as "best case" control for the individual human-algorithm fusions that follow.

A predictor matrix for the PLS was created for each of the 49 subjects as follows. Estimates of face similarity for the 120 face pairs from the human subject were combined in a column-wise matrix with the estimates taken from the seven algorithms for the same face pairs. A jack-knife procedure operated by deleting each of the face pairs in turn and computing the PLS on the remaining 119 pairs. Accuracy was measured as the number of correct classifications for the deleted, "left-out" face pairs. In all cases, jack-knife procedures tested the accuracy of human algorithm fusion with 1 through 5 factors PLS solutions. The optimal number of factors for the 49 subjects varied from 1 to 5, with a median of 1, and a mean of 1.36. For each subject, only the solution that yielded highest accuracy is reported.

Fusing individual humans with the seven algorithms cut the human error rate by more than a factor of two (cf., Table 1). For reference, the error rates for the individual algorithms from [7] appear in first column of Table 2. As can be seen, the average error rate achieved by fusing individual subjects with the combination of expertise found in the seven algorithms is less than half of that achieved by the best algorithm.

To interpret the role of individual algorithms and humans in tbe fusion, we looked at the PLS-derived weights for combining similarity estimates in the PLS. These weights emerge from the PLS as a recipe for optimally combining algorithm and human similarity estimates to predict match status. Averaged across the 49 subject-based fusions, the strongest contributor to the fusion was the algorithm from Viisage Corporation [6]. This was followed by the algorithm from the New Jersey Institute of Technology (NJIT) [4]. Anonymous algorithms B and D also played a role in the fusion. The weights for humans were well below these algorithm weights indicating that the role of humans in the fusion was minor. In previous work [7], we fused the *average* human with the seven algorithms and found that the human role in improving performance in the fusion

Table 2. Results compilation for human-algorithm fusions. The first column contains error rates obtained previously [7] for each algorithm operating alone. The human error rate from Table 1 appears at the top of the column for comparison. The second column shows the average error rates achieved by fusing individual humans with each of the seven algorithms. The percentage reduction in error rate from the individual algorithms to human-algorithm pair fusion appears in column 3.

Source	individual error rate[7]	paired fusion error rate	% error rate reduction
human	.14	–	
NJIT	.12	.08	33%
Viisage	.20	.12	40%
CMU	.14	.09	36%
Algorithm A	.37	.13	65%
Algorithm B	.23	.11	52%
Algorithm C	.25	.12	52%
Algorithm D	.26	.13	50%

was more substantial. Here we found that fusing *individual* subject performance with the seven algorithms and *averaging* the weights afterwards revealed a much smaller role for humans. This suggests that individual human performance may vary strategically and that particular person-tailored mixtures of individuals and algorithms may provide for better balance in human-machine fusions. This finding motivates the next analysis in which we fused all possible pairs of individual humans and individual algorithms.

3.3 Fusing Individual Humans and Individual Algorithms

The fusion was carried out as described previously, but this time combining each of the 49 subjects inidvidually with each of the 7 algorithms. By definition, only 1 and 2-factor solutions are possible, and so we carried out a pre-test of all human-model combinations using PLS to determine the optimal number of factors for each pairing. In what follows, only the better of these two solutions is analyzed.

The average performance for the best pair-wise fusion between individual human subjects and algorithms appears in column 3 of Table 1. Performance for this fusion is reduced only slightly from the performance we found when combining individuals with all seven algorithms. This indicates that the performance of pair-wise fusions can come close to that achieved by fusing humans with all available algorithm experitise. For reference, the average performance of possible pairs of human-algorithm fusions was 0.111, indicating that the best pair-wise fusions were substantially better than the overall average of paired-fusions. This suggests that human-machine fusions are not all equally beneficial.

We note also that in previous work [7], fusing the 7 algorithms (without humans) produced an error rate of .059. This is comparable to the error rate of individual humans fused with the seven algorithms, but is substantially larger than the near-zero error rate achieved when fusing the average human with the 7 algorithms [7].

Overall, the NJIT algorithm [4] produced the best performance when paired with individual human subjects. Forty of the 49 subjects produced their best fused performance with this algorithm. An additional 8 subjects paired best with Carnegie Mellon University's algorithm [5] and one person paired best with Algorithm B. That said, when comparing the performance of the algorithms individually (column 1 of Table 2) with the performance of the paired human-algorithm fusions (column 3 of Table 2, the error rate reductions are remarkable(column 4 of Table 2 shows percentage error rate reductions for the seven algorithms). On average, the error rate is reduced by 47 percent. Clearly, much of this improvement comes from the reduction of error rate for lesser performing algorithms. It bears noting, however, that even with the large error rate reductions, the average performance of a human alone is only slightly worse than the paired fusions for the lesser performing algorithms. Given that the paired fusions are worthwhile only when they better the error rates of the algorithm or human operating alone, the pairwise fusions of CMU[6] and NJIT [4] are the most promising candidates for fusion.

Finally, the balance of man and machine input in these fusions can be assessed by looking at the PLS-derived weights for the pairwise fusions. These are displayed in Table 3, with the PLS-derived weights for the human and algorithm scores in the first and second columns, respectively. The proportion of human contribution to these fusions is given in column 3. The human contribution varies from a minimum of .0439 for Viisage [6] to almost complete dominance for anonymous algorithms A and C. Notably, the man-machine balance varies widely with the algorithm.

4 Discussion

In most security applications, the use of face recognition algorithms is under the supervision of a human operator. Previous studies indicate that the performance of algorithms can compete with humans in some cases [3]. The accuracy of *individual* humans and *individual* algorithms, however, can be quite variable. In these cases, the question of whether algorithms or humans perform more accurately may be less important than understanding how particular man-machine combinations perform.

In this study, we fused individual algorithms and individual humans and show that these pairwise fusions performed substantially better than the algorithms operating alone. This suggests that humans can contribute, through fusion, to better face recognition performance. One caveat, however, is that these pairwise fusions must compete, not only with the performance of the algorithm alone, but with the performance of the human alone. From this perspective, pairwise

Table 3. The balance of human-algorithm contributions in the pairwise fusions. The weights for humans and algorithms appear in the first and second columns, respectively. The third column gives the proportion of human contribution to the overall fusion. This proportion is computed as $\frac{\text{Human}}{\text{Human+Algorithm}}$.

Source	human weight	algorithm weight	proportion human
NJIT	−0.2683	−2.4902	0.0973
Viisage	−0.3268	−7.1175	0.0439
CMU	−0.2992	−0.0900	0.7689
Algorithm A	−0.4073	−0.0025	0.9940
Algorithm B	−0.3283	−2.4525	0.1180
Algorithm C	−0.3422	−0.0009	0.9974
Algorithm D	−0.4052	−2.3027	0.1496

fusions compete with both humans and algorithms only for the best performing algorithms.

Finally, we show that it is important to combine human and algorithm-generated responses in quantitatively precise ways to improve performance optimally. The strong variability in man-machine balance we found across the 7 algorithms illustrates the importance of considering algorithms and humans as individuals.

References

1. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, K., Hoffman, J., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challeng. In: Proceedings of the IEEE Computer Vision and Pattern Recognition. (2005) 947–954
2. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Worek, W.: Preliminary face recognition grand challenge results. In: Proceedings of the Seventh International Conference on Automatic Face Recognition. (2006) 15–24
3. O'Toole, A., Phillips, P., Jiang, F., Ayyad, J., Penard, N., Abdi, H.: Face recognition algorithms surpass humans. submitted (2006)
4. Liu, C.: Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. IEEE:Transactions on Pattern Analysis and Machine Intelligence (2006) 725–737
5. Xie, C., Savvides, M., Kumar, V.: Kernel correlation filter based redundant class-dependence feature analysis (kcfa) on frgc2.0 data. IEEE International Workshop on Analysis and Modeling Faces and Gestures **1** (2005) 32–43
6. Husken, M., Brauckmann, B., Gehlen, S., von der Malsburg, C.: Strategies and benefits of fusion of 2d and 3d face recognition. In 1, ed.: Proceedings of the IEEE Workshop on Face Recognition Grand Challenge Experiments. Computer Society Digital Library. Volume 3., IEEE Press (2005) 174

7. O'Toole, A., Abdi, H., F., J., Phillips, P.: Fusing face recognition algorithms and humans. submitted (2006)
8. Abdi, H.: Partial least squares regression. In Beck, M., A., B., Futing, T., eds.: Encyclopedia for Research Methods in the Social Sciences, Thousand Oaks, CA, Sage (2003) 792–795
9. McIntosh, A., Lobaugh, N.: Partial least squares analysis of neuroimaging data: applications and advances. Neuroimage **23** (2004) 250–263

A Method for the Automatic Analysis of Colour Category Pixel Shifts During Dichromatic Vision

Mike Bennett and Aaron Quigley

Imaging, Visualisation & Graphics Lab,
Systems Research Group, School of Computer Science & Informatics,
University College Dublin, Dublin 4, Ireland
{mike.bennett, aaron.quigley}@ucd.ie

Abstract. In this paper we present a method for automatically evaluating the amount of colour changes images undergo when perceived by individuals with colour deficient vision. This measure enables the classification of images based on the extent images visually change when viewed by people with one of the three classes of dichromatic (protanopia, deuteranopia, and tritanopia) colour vision. By measuring the extent that colour images appear perceptually different a designer, or automated layout technique, will have an indication of whether the choice of colour usage in an image could lead to colour ambiguity or colour confusions.

1 Introduction

Colour is widely used in information visualisations, user interfaces, the design of physical spaces, safety systems and in many other areas. When colour is used correctly it enhances people's abilities to carry out tasks, such as distinguishing objects and classifying them into groups (Figure 1[1]).

Yet the ability to perceive colour is something which is often taken for granted. There exists many people for whom trichromatic colour vision is impossible. Colour deficient vision (dyschromatopsia), or colour blindness as it is more commonly known, affects approximately 8% of males and 0.4% of females [1][2].

In this paper we present a method for automatically evaluating the amount of colour changes images undergo when perceived by individuals with colour deficient vision (CDV). This measure enables the classification of images based on the extent images visually change when viewed by people with one of the three classes of dichromatic (protanopia, deuteranopia, and tritanopia) colour vision. By measuring the extent that colour images appear perceptually different a designer, or automated layout technique, will have an indication of whether the choice of colour usage in an image could lead to colour ambiguity or colour confusions.

For example, colour confusions can arise if a dichromatic viewer is watching a football game on TV where one team is wearing a red jersey and the other

[1] Colours used in the figures in this paper may not appear correctly when printed.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 457–466, 2006.
© Springer-Verlag Berlin Heidelberg 2006

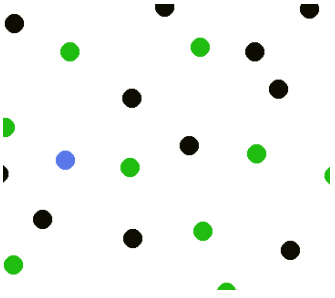


Fig. 1. An example of colour used to create three groups of dots (green, black, blue). Blue is also used to create a visual pop out effect.

team is wearing a green jersey. Both teams could easily be perceived as wearing the same colour jersey. Depending on the wavelengths of light an individual with protanopia or deuteranopia will not be able to distinguish between red and green. The problem is that red and green become weakly perceptually different. The colours become metamers due to the physiology of CDV eyes. That is, the colours are perceived as belonging to the same colour category [3] and may only vary in saturation and lightness.

In the following sections we provide background information on CDV and related aids for evaluating the impact of CDV. Following this is an outline of our Colour Category Pixel Shifts (CCPS) method, and related Refined Colour Category Pixel Shifts (RCCPS) method, with an elaboration on how both work. We then present the results of an analysis of three hundred natural images processed with CCPS.

2 Background

2.1 Colour Deficient Vision

A common misconception about people with CDV [4] is that they only see in shades of gray, as though continuously seeing the world as a black-and-white photograph. This is not the case, the vast majority of people with CDV perceive a range of colours.

The human eye consists of approximately 6 to 7 million cones and 120 million rods [5][6]. There are known to be three classes of cones, each of which has a peak sensitivity to a different wavelength of light. One way of thinking of these classes of cones is that they classify light into three frequency ranges which correspond to the colours perceived as red, green and blue. Rods enable humans to see at night and contribute little to colour vision. Rods are a lot more sensitive to light and are somewhat overwhelmed in normal daylight. Thus cones enable photopic (colour) vision and rods enable scotopic (low light) vision.

Colour vision deficiencies are the result of a reduced light sensitivity of the cones or a complete lack of one or more classes of the cones. When there is

a reduced sensitivity in one class of cones a person is said to have anomalous trichromatic colour vision. When one class of cone is completely missing, or non-functioning, a person has dichromatic vision. Cone monochromacy is when only one class of cones are functioning, and if no cones are functioning or present a person has achromatic vision.

Inherited variations in colour vision cannot currently be 'fixed'. These variations have a physiological basis, which is most often associated with X chromosome genes. Colour vision deficiencies can be acquired due to the normal aging process and may also develop due to a range of illnesses, such as cataracts, diabetes and age-related macular degeneration.

2.2 Related Work

A number of approaches have been taken to helping designers understand, evaluate and reduce the impact of CDV.

One approach to helping people understand the impact of CDV is by providing tools that simulate various kinds of colour defective vision [7][8][9]. These tools can be used by visual designers to evaluate designs. Designers can 'see' the visual designs through another's eyes. For example the Aging Vision Simulator [10] transforms images based on an approximation of how the lens in the human eye develops a yellow tint as people get older. Another example is a colour blindness simulator called Vischeck [9], which we used as part of testing the CCPS and RCCPS methods (Section 3.2). Vischeck transforms the colour in images based on a human visual system model [8] of how dichromatic perceivers see colours.

Examples of approaches for automatically reducing the impact of CDV are the Daltonize algorithm [11] and the work of Rasche et al [12]. In both cases automatic adjustments are made to how colours are used in images. With the Daltonize algorithm visual information which would be hidden from red-green dichromats becomes perceivable. The Daltonization algorithm works by establishing the amount of information in an image that is encoded with red-green colours. Based on this analysis it attempts to shift the red-green colour dimension into the blue-yellow and light-dark dimensions. With colour shifting colours that were distinct could start to become similar, thus introducing colour confusions and ambiguity. Colour constancy over multiple images may also become an issue, i.e. red could be converted into a light blue in one image and into a dark purple in another image.

3 Colour Category Pixel Shifts

3.1 Overview

CCPS is a measure of the percentage of pixels in an image that change colour category when viewed by individuals with dichromatic colour vision. RCCPS is a measure of how many pixels change colour category in each colour category.

CCPS provides an overview of how much pixel change occurs, RCCPS provides details about which colour categories undergo changes along with how much change per category.

Both the CCPS and RCCPS measures rely on image transforms as part of the process of generating the measures. Before detailing how CCPS and RCCPS are calculated (Section 3.4) we outline the two key transforms.

The first transform converts an image into an approximation [8] of what an individual with dichromatic CDV would see (Section 3.2). The second transform is a form of colour image segmentation where pixels in an image are grouped according to colour category (Section 3.3).

3.2 Modeling Dichromatic Colour Vision

In vision science there are numerous models of different functions in the human visual system [13] but there are a limited number of models of CDV. Existing models [8][9] tend to focus on dichromatic CDV, with an emphasis on protanope and deuteranope vision.

We utilised the Vischeck plugin [9] for Adobe Photoshop to simulate CDV. Three hundred images from the Berkeley Segmentation Data Set (Section 4) were transformed with the plugin. These images where then used for testing CCPS and RCCPS.

Each image was transformed into three new images, each of which approximates how a protanope, deuteranope and tritanope perceives the image (Figure 2). When more diverse and accurate models of each of the other sub-types of CDV are developed they can also be used to carry out the image transforms.

3.3 Colour Category Classification of Pixels

Classifying pixels into perceptual colour categories [15] is a non-trivial task. There is a considerable body of work in both human and automated colour category naming and identification [3][16][17]. Based on the findings in colour category theory we partitioned the HSL colour space into the colour categories of White, Black, Red, Green, Blue, Yellow, Gray, Pink, Purple, Orange and Brown.



Fig. 2. First image unchanged from [14]. Second (protanope), third (deuteranope) and fourth (tritanope) images were generated with the Vischeck colour blindness simulator.

There are indications that these eleven colour categories tend to be commonly shared worldwide across languages and cultures [18][3] - though there are many unresolved questions about the number and nature of colour categories [19]. As algorithms for perceptually classifying pixels improve they can be utilised in generating the CCPS and RCCPS measures.

When classifying a pixel by colour category it is converted from the RGB to the HSL colour space. Then predominately based on Hue a pixel is classified into one of the eleven colour categories. Saturation and Lightness are used for classifying pixels when they are near their upper and lower bounds, corresponding to Black and White. Brown and orange were treated as a single colour category because distinguishing them often led to pixels which were brown getting classed as orange, and vise versa.

For example, if a pixel is light or dark red it is changed into a single RGB value representing both variations of red. This can result in images getting partially (depending on image complexity) segmented into blocks of colour, i.e. a blue sky with a gradient from light to dark blue becomes a single block of blue.

3.4 Calculating CCPS and RCCPS

A CCPS measure is generated by taking the following steps:

1. Transform an image into that which a dichromatic viewer would perceive (Section 3.2).
2. Classify each pixel in both the original and transformed images into one of the colour categories (Section 3.3).
3. Compare the original and transformed images checking whether each pixel has switched colour category (Section 3.4).
4. Calculate a percentage measure (Eqn. 1) based on the number of pixels which have switched colour categories divided by the total number of pixels in the original image (Section 3.4).

$$CCPS = \frac{Shifted\,Pixels}{Total\,Pixels} * 100 \tag{1}$$

It is possible to obtain a refined breakdown of the single percentage indicator generated by CCPS. This provides more information about how the pixels are shifting colour categories. To get a more detailed breakdown the amount of pixels in each colour category is calculated (Eqn. 2) in both the original and dichromatic transformed image. The difference in the change in the number of pixels per colour category is then used to generate a percentage indicator:

$$RCCPS = \frac{(CPO - CPT)}{Total\,Pixels} * 100 \tag{2}$$

RCCPS = Refined CCPS, CPO = Count of pixels in a specific colour category in the original image, CPT = Count of pixels in a specific colour category in the transformed image, Total Pixels = Count of total number of pixels in the original image.

Rather than producing a single CCPS value RCCPS is used to obtain a positive or negative percentage indicating how many pixels shifted away from or into a specific colour category. RCCPS can be applied to each colour category, producing a set of numbers that provide detail on how the pixels in each colour category are affected.

For example imagine that CCPS was integrated in a tool for designing user interfaces and that a designer uses CCPS to analyse an image consisting of 50000 pixels. After applying the dichromatic image transform and counting the number of pixels which shifted categories it is found that 42% of the pixels shifted colour category. A 42% change sounds serious, but when the designer visually examines the image they establish that most of the pixels which switched category changed from a light gray to a white. If RCCPS was carried out on the image, rather than visually examining it, it would tell the designer that pixels in the gray colour category suddenly decreased while pixels in the white colour category suddenly increased.

The ability to distinguish which colour categories are most affected is important because the three classes of dichromatic CDV are more sensitive to the effects of specific colours switching colour categories, i.e. red or green pixels switching colour categories for protanopes and deuteranopes.

4 Evaluating the Berkeley Segmentation Data Set

In this section we present the results of using CCPS to analyse 300 images. The images come from the Berkeley Segmentation Data Set (BSDS) [14], which is "*a large data set of natural images that have been segmented by human observers*". BSDS was originally created for comparing the performance of image segmentation algorithms.

Each image from BSDS was converted into three further images - a protanope, deuteranope and tritanope image (Figure 3). Then a C++ implementation of CCPS and RCCPS was used to process the 1200 images. The original images had to be processed (Section 3.4, Step 2) to work out the original number of pixels in each colour category in each image. Figure 3 shows an example of an

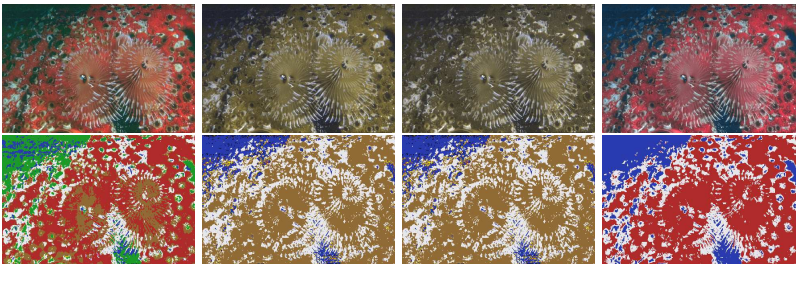


Fig. 3. First row is an example of an original, protanope, deuteranope and tritanope image. The second row shows the images after the pixels have been classified by colour category.

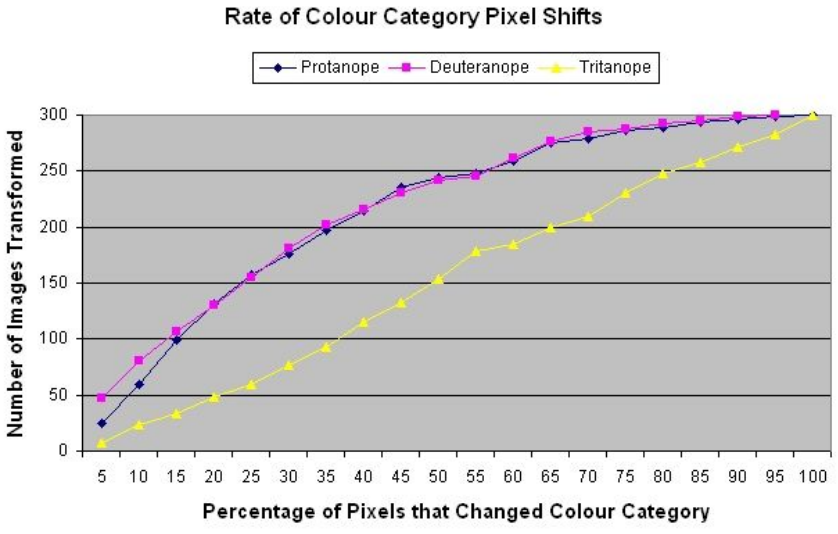


Fig. 4. Rate that CCPS pixels switch colour categories

original, protanope, deuteranope and tritanope image where the pixels have been classified by colour category.

After running CCPS on all the images we graphed the results (Figure 4 and Figure 5). One important aspect of the results is that there were a large number of pixels classified into the black, white and gray categories. This was primarily an artifact of how pixels are classified into colour categories. Distinguishing a light gray from a light blue is non-trivial so we errored on the side of caution by treating light colours as gray, very light colours as white and very dark colours as black. This had the effect that colour category pixel shifts tended to be weighted towards colours that were darker and more saturated. An emphasis was placed on pixels for which colour category shifts would be more perceptually distinct.

Looking at Figure 4 we can see that for protanopes and deuteranopes 100 of the images undergo a 15% pixel change or less, and that 150 of the images undergo less than a 25% change in pixel colour catgories. The graph also shows the rate of colour category shifts for deuteranopes and tritanopes. This graph indicates the BSDS images are most likely to undergo change when viewed by tritanopes.

Figure 5 provides a related graph that shows the distribution of percentage changes in colour categories. From this we can see that for protanopes 40 images have colour category pixel changes in the 10% to 15% range, and for deuteranopes 33 images have colour category pixel changes in the 10% to 15% range. The distribution for tritanopia is extremely varied - this may be due to a number of factors, ranging from the amount of yellow and blue in the BSDS images (a lot are in outdoor settings) to the underlying model used to model tritanopia vision. A more varied image set is required to analyse the interplay between CCPS and tritanopia.

It is important to note that until more refined models of CDV are developed, and very accurate ways of classifying the colour category of pixels become

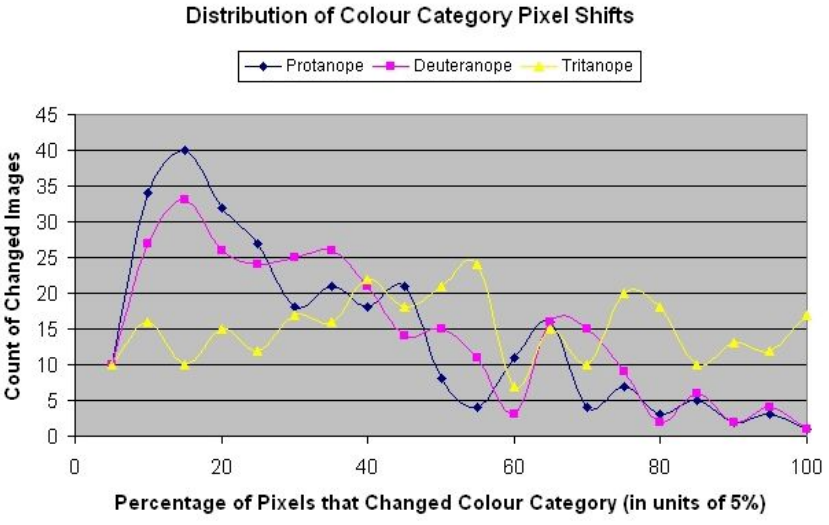


Fig. 5. The distribution of the CCPS percentage of pixels that switched colour categories in the images

available, CCPS and RCCPS should be treated as heuristic indicators. The results in the graphs should not be interpreted as definitive measures of colour category changes perceived in the world by individuals with dichromatic CDV.

5 Conclusions

In this paper we have presented two useful methods, CCPS and RCCPS, for automatically evaluating colour category changes images undergo when perceived by individuals with colour defective vision. CCPS provides an overview of how much colour category pixel change occurs in an image, while RCCPS provides details about which colour categories undergo changes along with how much change occurs per category.

We implemented CCPS and RCCPS and demonstrated how CCPS provides a measure of colour category changes in images due to dichromatic CDV. This was demonstrated by using CCPS to analyse a collection of images from a publicly available data set.

The CCPS and RCCPS methods could easily be implemented as part of an interface or information visualisation design tool. With the potential benefit of helping designers become aware of how they use colour, while also helping them understand the impact inappropriate colour usage can have on individuals with CDV.

5.1 Future Work

This paper has highlighted a number of areas where further research could be carried out. For example research could be focused on developing human visual

system models of anomalous trichromatic colour deficient vision. These models could then be used in conjunction with CCPS and RCCPS.

Pixel level colour category changes are a useful measure but it might prove more beneficial to decompose an image into higher-order objects and measure whether these objects undergo colour category changes. Related to this is a method for measuring whether a pixel shifts colour categories such that it becomes the same colour as the pixels surrounding it. This would indicate whether foreground and background colour differences are reduced in images for CDV perceivers.

Finally, better techniques for classifying pixels according to the colour categories they belong to would improve the performance of CCPS and RCCPS.

Acknowledgments. This work would not have been possible without the ongoing support from the EU Marie Curie CoViAn project, the IBM Dublin Center for Advanced Studies and the School of Computer Science and Informatics UCD, Ireland.

References

1. Hunt, R.: Measuring Colour. 3rd edn. Fountain Press (2001)
2. Hurvich, L.M.: Color Vision. Sinauer Associates Inc. (1981)
3. Berlin, B., Kay, P.: Basic Color Terms: Their Universality and Evolution. London: University of California Press (1969)
4. Mollon, J.D., Pokorny, J., Knoblauch, K.: Normal and Defective Colour Vision. Oxford University Press (2003)
5. Bruce, V., Green, P.R., Georgeson, M.A.: Visual Perception: Physiology, Psychology and Ecology. Hove: Psychology (1996)
6. Palmer, S.E.: Vision Science: Photons to Phenomenlogy. MIT Press (1999)
7. Meyer, G., Greenberg, D.: Color-defective vision and computer graphics displays. IEEE Computer Graphics and Applications **8** (1988) 28–40
8. Brettel, H., Viénot, F., Mollon, J.D.: Computerized simulation of color appearance for dichromats. Journal of the Optical Society of America. A, Optics and image science **14** (1997) 2647–2655
9. Dougherty, R., Wade, A.: Vischeck, http://www.vischeck.com (2006)
10. Becker, S., Tu, C., Kim, H.: Software tools for the effective use of color in web designs. In: Proceedings of the 2004 National Conference on Digital Government Research, Seattle, Washington (2004) 385–386
11. Dougherty, R., Wade, A.: Daltonize algorithm, http://www.vischeck.com/daltonize (2006)
12. Rasche, K., Geist, R., Westall, J.: Re-coloring images for gamuts of lower dimension. In Alexa, M., Marks, J., eds.: Eurographics. Volume 24., The Eurographics Association, Blackwell Publishing (2005)
13. Carney, T., Klein, S.A., Tyler, C.W., Silverstein, A.D., Beutter, B., Levi, D., Watson, A.B., Reeves, A.J., Norcia, A.M., Chen, C.C., Makous, W., Eckstein, M.P.: The development of an image/threshold database for designing and testing human vision models. In: Human Vision, Visual Processing, and Digital Display IX, Proc. SPIE. Volume 3644. (1999) 542–551

14. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the 8th Int'l Conf. Computer Vision. Volume 2. (2001) 416–423
15. Kay, P.: Color categorization. MIT Encyclopedia of the cognitive sciences (1999)
16. Lammens, J.M.G.: A Computational Model of Color Perception and Color Naming. PhD thesis, State University of New York at Buffalo (Computer Science) (1994)
17. Belpaeme, T.: Factors influencing the origins of colour categories. PhD thesis, Artificial Intelligence Lab, Vrije Universiteit Brussel (2002)
18. Yendrikhovskij, S.N.: Computing color categories from statistics of natural images. Journal of Imaging Science and Technology **45** (2001) 409–417
19. Saunders, B., van Brakel, J.: Are there non-trivial constraints on colour categorization? Behavioral and Brain Sciences **20** (1997) 167–228

Rendering Dynamic Real-World Scenes Using Image Spheres

Stephan Behrendt

Institute for Real-Time Computer Systems (RCS)
Technische Universität München
D-80290 Munich, Germany
stephan.behrendt@rcs.ei.tum.de

Abstract. We introduce *Image Spheres* - a novel approach for rendering dynamic real-world scenes. As sensors for online geometry reconstruction enhance more and more, it is desirable to render their output while up-to-date color information is applied. Images received from cameras mounted on moving pan-tilt units are uploaded to the graphic memory almost immediately after they are received. With very low preprocessing they are used to color the perpetually updated geometry immediately. To control graphic memory usage, our method includes a fast image replacement algorithm which regards redundancy when deciding which image to delete. We demonstrate our results on high polygonal laser scan models combined with geometry data received by a Photonic Mixer Device range sensor.

1 Introduction

It is still a challenge to render dynamic real-world scenes in photo-realistic quality. An arbitrary real-world scene consisting of a various number of changing objects is quite difficult to reproduce.

Two major stages play important roles. On the one hand it is not a trivial task to gather all the information needed, namely geometry and color data. Fast update rates and sufficient accuracy are hard to achieve. On the other hand displaying with interactive frame rates is difficult as well.

As methods and sensors aiming at the first objective become better, new techniques are necessary to render high amounts of temporal changing range sensor data. Additionally, color information is very important for a human observer to feel immersed in the real-world scene. Unfortunately combining the output of geometry sensors with video streams cannot be achieved in real time by standard texture mapping.

In the following pages the concept of Image Spheres is introduced: A new approach which is capable to handle huge amounts of input data from different sensors by storing the gathered data and updating perpetually. For example a basic geometry model acquired by a laser scan (about one million triangles) can be updated over time using a PMD (Photonic Mixer Device) [1] range sensor. The sensor we use produces up to ten depth images per second at a resolution

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 467–479, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of 160 x 120 pixels. At the same time one or more video cameras mounted on pan-tilt units are used to apply color information to the current scene. Up to two cameras at a resolution of 640 x 480 pixels at 33 frames per second are used in our setup. Furthermore the scene is rendered with interactive frame rates using current of the self hardware components. For this purpose it is necessary to cope with large geometry models because geometry sensors produce high amounts of raw data.

As immersiveness increases when the field of view is not restricted to the current center of interest we try to keep older sensor data as far as possible (see [2][3]). A memory management component allows to show a maximum part of the scene with the newest sensor data received. The following pages focus on the processing of the video or image data and how to render high polygonal models using the gathered color information. After a survey of related literature the basic idea of our approach is described followed by a description of the image spheres. A system description as well as results can be found in the following chapters.

2 Related Work

Since geometry sensors able to capture dynamic environments are quite new, related work mainly focuses on non dynamic methods for rendering real-world scenes.

First of all the various approaches of image based rendering (IBR) have to be mentioned. Since the mid-nineties many solutions have been developed in this field: plenoptic modeling [4], light field rendering [5] and the lumigraph [6] are only three of them. These methods result in quite nice new generated views from sets of images. While the mentioned IBR methods are mainly based on a relative dense sampling of images (light field rendering) or geometry-assisted image interpolation (lumigraph), layered depth images [7] use multi layered images where depth is added to reproduce views from arbitrary view positions. These techniques all have in common that they cannot use standard graphic pipelines without major modification. Hence they do not benefit much from newer generation graphic cards. Using them for dynamic changing scenes is not possible at all. In the field of telepresence virtualized reality [8] is a Method which is very well focused on dynamic scenes. As this approach requires a complex setup of sensors, it is not viable for many applications in practice. Improvements of traditional IBR approaches could lead to interesting results too. Todt et al. [9] propose a system using a PMD sensor as well. But to our knowledge there are no methods at this time which are capable to combine arbitrary geometry sources which produce large data amounts.

Besides the IBR techniques, approaches based on traditional computer graphics exist as well. Burkert et al. [10] describe efficient techniques for texture extraction from camera images which are later on rendered traditionally. Photo-realistic results are achieved. Unfortunately extracting a texture for each rendered triangle is required. So this approach is limited by the complexity of the

geometry models and hence difficult to apply to unprocessed sensor data. In our approach we bypass this step thus widening the limits for the size of geometry models dramatically.

3 Display Concept

Our display has two major types of input data: geometry data provided as indexed face sets, and image data received as video streams containing the current 6 degree-of-freedom (DOF) pose of the capturing cameras. As we will describe later on, spatial partitioning of the scene is necessary for our approach. Subdividing the room in cubes with a certain size suits quite well for example (see section 4.3). The geometry input is designed in such a way that it holds a list of cubes that contain new geometry replacing older one. To be able to show new color information as fast as possible, the received images are uploaded to the graphics card's memory immediately after receiving them. Accordingly the new geometry is uploaded to the graphic memory. It is important to ensure that geometry and images are captured at the same time otherwise inconsistencies can occur. Fortunately efficient mechanisms on current graphics boards allow fast transfers for geometry as well as for image data. (see Vertex Buffer Objects and Pixel Buffer Objects [11]). Uploads are performed in the time between rendering two frames. To guarantee a certain frame rate the maximum number of geometry updates and image updates between two frames can be set.

3.1 Preselection

Already uploaded geometry is rendered consecutively. To apply color information, for each cube a preselection of the stored images together with their intrinsic and extrinsic camera parameters are passed to our vertex and fragment programs. This set of images we call candidates. The selection is based on the position of the cube region and the 6 DOF pose of the stored camera images. To select the candidates the camera's optical axis at the moment the image was grabbed is used. An image is supposed to be a good candidate if the distance of current cube center to the optical axis is small. The fragment programs we use are capable to distinguish for each drawn pixel which candidate should be used for coloring and which position of the image has to be used. Alternatively colors of two or more candidates can be merged (see section 5).

Preselecting candidates is important since the number of textures that can be used at once is limited by the graphics hardware (32 on most current graphic boards). Moreover support of passing a high number of candidates leads to more complex shader programs, resulting in longer per frame rendering times. The maximum cube size is also limited to ensure that a small number of candidates per cube is sufficient to provide color information for all geometry included in the cube (see 4.3).

The preselection computational time has a major impact on the system performance. By rendering a single frame for each of the cubes the preselection has

to be done. The overall time needed for preselections increases linearly with the number of drawn cubes. A well performing algorithm is needed for the preselection process.

3.2 Image Replacement

Another important point is the management of graphic memory usage. As stated above, new arrived images of the several camera streams are immediately uploaded to the graphics card. Considering a single video stream at a resolution of 640 x 480 with 24 bit per pixel at 30 frames per second, more than 26 Megabyte per second are needed to save the raw data. As current graphic cards provide about 256 to 512 Megabyte of memory its utilization must be controlled. After uploading a certain number of images we must delete older ones, thus reducing the required memory per received video stream to a maximum. The maximum number of images kept in memory depends on how many video streams are currently received, their resolution and color depth and the available memory on the graphics board. The accumulated memory usage should not be higher than the graphic card's memory. Otherwise textures are swapped to main memory resulting in poor performance. A simple FIFO buffering concept could be used as replacement criterion, but would lead to the following problem. If we want to present the real scene with the maximum amount of color information we have received until the current moment, we should rather delete images redundantly covering the same parts of the scene than simply the oldest ones. The decision which image to delete should search for an image which content is as much as possible contained by an newer one. Again the computation time has major impact on the system overall performance since it is needed once by receiving a new image.

4 Image Spheres

For our concept a well performing algorithm is needed to select candidates for rendering and to decide which images should be deleted. When capturing images with automatically moving cameras it is a common technique in many areas of application to use stationary pan-tilt units. Exploiting this, a technique we introduce as image spheres is capable to fulfill both requirements.

First of all we pretend that our cameras rotate around their nodal point. In real world applications this is seldom given but the introduced error is calculable (see 4.3). We section the room around the nodal point into frustums which are determined by longitude and latitude of their centers. This parametrization is not uniform, and it has singularities on the poles but leads to good results anyway.

Each of our cameras is assigned to one sphere. When receiving a new image we insert it into the according sphere, where yaw and pitch of the image's 6 DOF pose correspond to longitudes and latitudes of the sphere. By doing so, it is quite easy to determine a set of images suiting best for a certain cube. We simply have to calculate the direction of the vector pointing from the sphere's nodal point to the cube and look up the images that are nearest to that direction.

If the maximum number of images is reached, the image which has to be deleted is found using the density of our sphere. We define the density by the number of images per area on the sphere surface. Thus, the nearer two already inserted images are, the higher the density at their locations is. Selecting the older one of the two images, resulting in the spheres current maximum density, we have quite a good approximation of the prior required characteristic for the deletion of images: A high redundancy in coverage of the scene in comparison to the overall redundancy in two images.

Calculation of the image sphere's densest region seems not suitable way when desiring a fast method. By mapping the image sphere concept to a quad tree [12] like data structure, we can perform both types of lookups quite efficient.

4.1 Extended Quad Tree Structure

The two dimensions of the quad tree correspond to longitudes and latitudes. Because in our context we traverse the tree rather bottom-up than top-down we also use a lookup table to ensure fast access to leaf nodes given a certain pair of longitude and latitude. When inserting an image, we set the corresponding leaf node and all ascending nodes until the root node point to this image. When we need to find an image close to a given longitude and latitude we only need to ascend the tree until we find a not empty node. A traditional quad tree structure causes a problem. The regions corresponding to the consecutively looked up nodes are not growing around the starting coordinates. (Fig. 1 shows an example). That behavior can result in finding an image quite distant to the desired location. If we modify the quad tree structure by adding new connections to

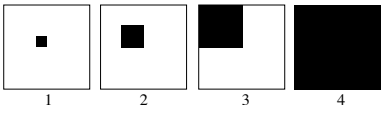


Fig. 1. Order of Regions that are looked up when ascending an ordinary Quad Tree form level one to four

each node we can improve this behavior. Each node gets three new connections upwards besides the father node: two uncles and one grand cousin (see Fig. 2).

If the additional connections of each node are also checked when ascending the tree structure, the center of the searched area grows around is nearer the starting coordinates than before (see Fig. 3). Performing that lookups adds additional cost when ascending the tree. For each level, four lookups instead of one lookup must be done. In practice these lookups are simply checking if a pointer has a NULL value, therefore the overall time for ascending the tree does not increase dramatically.

The described tree structure enables us to find the nearest saved image to a given position quite easily. For finding the oldest image at the densest region of the sphere we need additional hints. Besides our tree structure we use

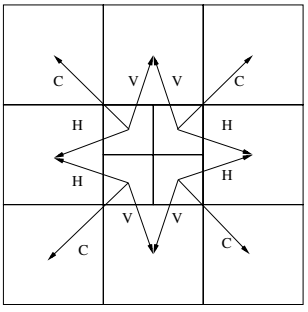


Fig. 2. Additionally inserted connections of the inner four nodes. H: Horizontal Uncle V: Vertical Uncle C: Grand Cousin.

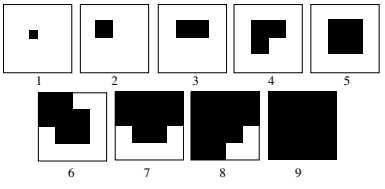


Fig. 3. Order of Regions that are looked up when ascending the modified Quad Tree for each level

several FIFO buffers, one for each level of our tree. This buffers hold a list of inserted images for each level of the sphere. We call them Level FIFOs.

4.2 Usage of Image Spheres

Insert an Image: With the help of the previously described tree structure and the Level FIFOs, insertion of an image is accomplished as follows:

1. Determine the leaf node of the tree corresponding to the new image's yaw and pitch by using the lookup table.
2. Walk up the tree until the root node is reached and set the passed nodes to point to the new image. While walking up the tree check every node (including the leaf node) if it points already to an image. When an image is found, it is inserted into the corresponding level FIFO.
3. If the number of maximum images is reached we check all Level FIFOs beginning with the lowest level and delete the first image found. All references in the tree and in the Level FIFOs to that image have to be invalidated as well.

Find an Image: Finding the best possible image for a certain cube is achieved like follows:

1. Determine yaw and pitch of the direction the cube is located with respect to the sphere center and lookup the corresponding node of the tree.
2. Check if the node contains an image. If not check the father node as well as the two uncle nodes and the cousin node if an image is contained there. Ascend the tree until an image is found. If at least one image has already been inserted it will be found.

If more than one image should be returned the child nodes have also to be checked while ascending the tree. Normally one candidate is sufficient for achieving good results. But in case that the nodal error is large fetching two candidates can help. Both operations, insertion and selection of the candidates can be realized quite easily by the use of image spheres. In order to invalidate all references in the tree and in the FIFOs when deleting an image a Smart Pointer Concept can be used [13].

4.3 Restrictions

Even if our tests show good results and the described technique is suitable for our field of use, two major errors are introduced.

We use a non uniform parametrization for the dense estimation on our spheres and we do not take into account that in many applications a camera does not rotate around the nodal point. In our system we seldom deal with pitch values leading to latitudes near the poles, so the dense approximation works quite fine (see section 8).

In order to be able to select candidates for spatial portions of the geometry it was stated that we need to subdivide the space. This subdivision can be done in various manners. The important thing is that we have a maximum distance $size_{max}$ from the subset centers to the contained Vertices. It is desirable to have a criteria for $size_{max}$ that ensures that an image candidate selected by our algorithm covers a given geometry subset. If we divide the problem in horizontal and vertical coverage such a criteria can be formulated quite simply (see figure 4 for a two dimensional view of the problem).

Given the field-of-view (FOV) of the camera fov, a minimum distance of the camera to the geometry d_{min} and the displacement of the camera's nodal point to the intersection of the two pan-tilt axis $d_{nodoffset}$, $size_{max}$ can be calculated as follows:

$$size_{max} = \sqrt{d_{nodoffset}^2 + d_{min}^2} * sin(\frac{fov}{2} - arctan(\frac{d_{nodoffset}}{d_{min}}); \qquad (1)$$

Note that the given equation considers only one dimension. In practice $size_{max}$ has to be determined for the horizontal FOV and the horizontal $d_{nodoffset}$ as well as for the vertical counterpart. The minimum of the resulting values gives the overall $size_{max}$.

Motion blurring has to be considered also when using our approach in practice. We use a camera motion threshold to be sure that an image is not blurred. Using blurred images would result in annoying artifacts when used for permanent

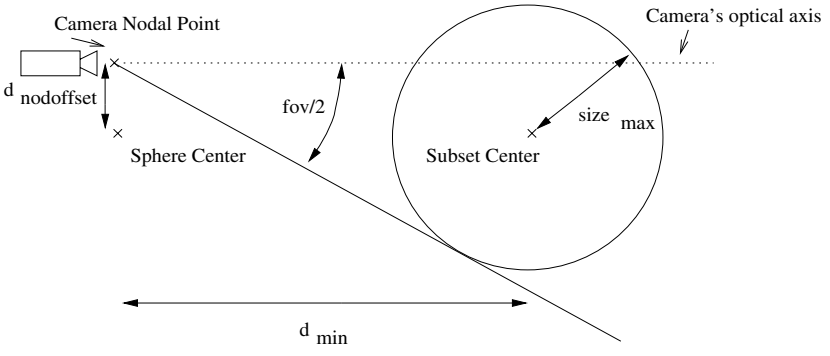


Fig. 4. Two dimensional view of the setup

coloring. To display the newest information an image supposed to be blurred is only used for rendering until a new one arrived but is not inserted in an image sphere.

Even if we try to arrange the images we get from the sensors in a efficient way, our approach is not able to guarantee a full coverage of the geometry as the area the camera pans is too large for a set maximum of images by sphere. If the density of the images is too low, our algorithm will result in visible holes in the scene. This fact has to be considered when setting a maximum number of images for the spheres.

5 Rendering

After retrieving one or more candidates from our image sphere for a cube we need to render the contained geometry together with the color information in a fast way. The fact that current graphics hardware supports projective texture mapping [14] helps us here. When setting up an image sphere for a camera the intrinsic camera parameters are used to compute a projection matrix while the extrinsic parameters (the 6 DOF pose) coming with the image can be used to compute the modelview matrix for each image. In the vertex shader we additionally transform the current vertex according to these matrices. Passing these data to the fragment shader, the perspective projection of the texture is done. When we have more than one candidate resulting from one or more image spheres, the color information can be merged if more than one candidate contains the current processed fragment. Blending according to the distance of the taken pixel to the cameras optical axis achieves quite nice changeovers [15]. If the scene does not contain occluded objects which are not seen from one or more cameras the projective texture mapping is all which has to be done at the rendering stage.

Occlusion detection: To be able to correctly present occluded objects we also benefit from the features current hardware offers. Since hardware shadow mapping [16][17] is a feature which is very well supported by current graphic chips,

we use it for this purpose too. For shadow mapping a depth map generated from the light source‘s position needs to be generated. For occlusion detection, we also create a depth map but in our case from the camera point of view. The previous mentioned projective and modelview matrices are used here too. For each new camera image a new depth map is created and also passed to the fragment shader when the associated texture is selected as a candidate for coloring. In the fragment shaders we deal with the depth texture just in the same way as we would if we would apply shadow mapping. The decision if the candidate is valid or not results on the result of the depth compare. If a light source would be in the camera's point of view the occluded region would lie in the shadow of the occluding object. In our application this means that the candidate is only valid for the occluding object. Please refer to [16][17] for more detailed information on shadow mapping. By doing so we are able to handle occlusions by having low additional cost for it.

6 Application Overview

The overall system is designed for a typical telepresence scenario where a human operator controls a remote robotic manipulator to execute a certain task. Firewire cameras mounted on pan-tilt units observe the robot's workspace. On one of the pan-tilt units, a lasermeter as well as a PMD Camera [1] are also mounted for geometry reconstruction. The PMD 19k model we use, produces about ten depth images per second at a resolution of 160x120 pixels. All cameras are calibrated intrinsically. A hand-eye calibration provides the extrinsic camera parameters [18]. A server controls the pan-tilt units, grabs the RGB camera images and sends them as xvid streams together with the 6 DOF poses of the cameras to the receiver. The PMD images are directly processed at the sending side. A simple meshing is applied on the sensor data. The resulting geometry is then also sent via network using the Mpeg 4 BInary Format for Scenes (BIFS) [19]. For displaying the scene, an AMD Athlon 64 X2 ™ 4800+ processor equipped with two Nvidia 7800 GTX ™ graphic boards working in the Scalable Link Interface (SLI) Split Frame Rendering Mode receives the network streams. The measured times in chapter 7 are based on that system.

7 Results

We demonstrate the results of our approach on the basis of our scenario described above. We use a model generated with the lasermeter for the background. This model contains nearly one million triangles. Figure 5 shows the geometry rendered using one image sphere containing 24 images with enabled occlusion detection and a photograph of the scene. With our system about 24 ms are needed to render one frame if no updates are done. A complete table of frame rates is given in table 1. The given rates correspond to a view port resolution of 800 x 600 pixels. The video streams have a rate of 33 frames per second at a resolution of 640 x 480 pixels. All measurements include an update of 18921

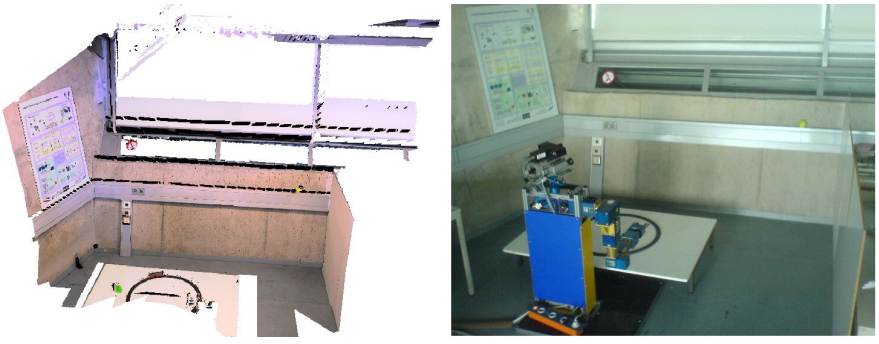


Fig. 5. Full scene rendered using a set of 24 images and a laser scan model containing 989.902 triangles(left)and a photograph of the scene (right)

Table 1. Measured Frame Rates

Occlusion detection	One sphere without update	One sphere with image update	Two spheres without update	Two spheres with image update
disabled	40,3 Fps	39,4 Fps	37,3 Fps	30,9 Fps
enabled	39,9 Fps	39,1 Fps	34,4 Fps	21.9 Fps

triangles between two frames. This is half of the triangles generated of one frame the PMD sensor produces. In figure 6 we show the result of starting with an empty sphere moving the camera from bottom left to to up right of the scene. The result of panning with two cameras is also shown. We state that our approach enables us to combine different geometry sources as well as image sources and update them. Using a laser scan model of the non changing background and combining this model with geometry generated by a PMD Sensor gives quite nice results. We use two different image spheres, one which is not updated coloring

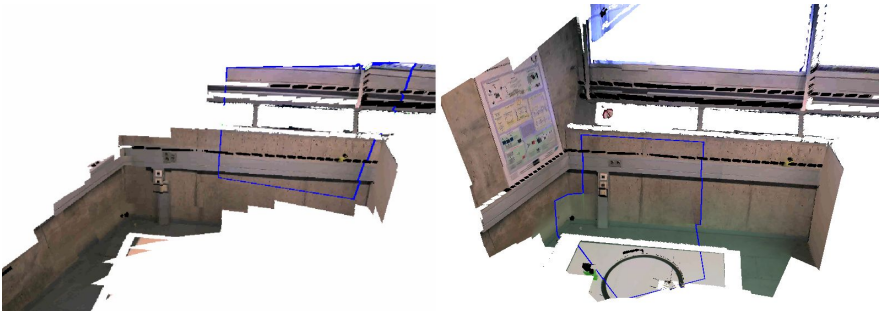


Fig. 6. Laser scan model rendered using video streams. The current video image is marked by a blue border. Left: diagonal pan Right: two streams simultaneously coloring the scene.

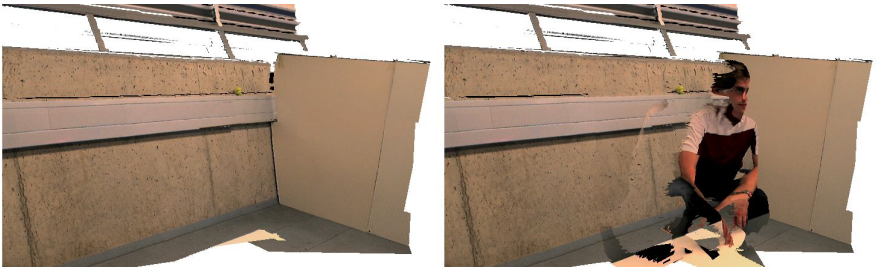


Fig. 7. Rendered laser scan model (left) combined with geometry generated with a PMD range sensor of a person (right)

the background and one which is connected to a camera stream. Doing so it is possible to render new foreground objects while the background does not suffer from occlusions. In figure 7 the scene is rendered after a person has entered the scene. We combine the advantages of the different sensor types. The accuracy of the laser scan model and the fast geometry creation by the PMD Sensor.

8 Conclusion and Further Work

We presented a system which is capable to apply fast updated color information on large polygonal models. By the use of image spheres we can balance the use of graphic memory while we can select good image candidates for geometry subsets of complex models as well. For deletion of older images we come to a fast decision based on the image density in our sphere just as well.

Nevertheless our approach has some weaknesses. If a whole coverage of directions is needed the non uniform parametrization results in favoring the pole regions of the image spheres. Thinning out the tree structure in a sophisticated way could be a viable solution. Reducing the nodes representing pairs of longitudes and latitudes in the pole regions of the spheres could solve this behavior. Unfortunately proceeding this way is quite difficult. Changing the tree by deleting some nodes and reconnecting the references of this node to neighbors causes problems when approaching the poles where many nodes have to be mapped to a single one.

As can be seen in table 1, the performance decreases when more live video streams are simultaneously updated. In that case rendering times also vary a lot. The reason seems to be that capturing more streams can result in grabbing all streams in between rendering two frames while between the next two frames nothing is captured. Reducing the update frequency can be used as a workaround but is not a satisfying solution. Introducing a scheduling which balances the amount of update which is done between two frames and thus giving guarantee on frame rates is desirable.

The presented method uses massively the potential of the current graphic chips. Looking at the immense growing capabilities of shader pipelines and their parallelization we expect that our method will strongly benefit of further enhancements in this area too.

Acknowledgments

This work is supported in part by the German Research Foundation (DFG) within the Collaborative Research Center SFB 453 on "High-Fidelity Telepresence and Teleaction".

References

1. Kraft, H., Frey, J., Moeller, T., Albrecht, M., Grothof, M., Schink, B., Hess, H.: 3D-Camera of High 3D-Frame Rate, Depth-Resolution and Background Light Elimination Based on Improved PMD (Photonic Mixer Device)-Technologies. In: OPTO. (2004)
2. Rosenblum, L., Macedonia, M.: Tangible Photorealistic Virtual Museum. IEEE Computer Graphics and Applications **25** (2005) 15–17
3. Tang, W.K., Wong, T.T., Heng, P.A.: A System for Real-Time Panorama Generation and Display in Tele-Immersive Applications. IEEE Transactions on Multimedia **7** (2005) 280–292
4. McMillan, L., Bishop, G.: Plenoptic modeling: An image-based rendering system. In: SIGGRAPH '95. Volume 14., New York, NY, USA, ACM Press (1995) 39–46
5. Levoy, M., Hanrahan, P.: Light field rendering. In: SIGGRAPH '96. Volume 15., New York, NY, USA, ACM Press (1996) 31–42
6. Gortler, S., Grzeszczuk, R., Szeliski, R., Cohen", M.: The lumigraph. In: SIGGRAPH '96. Volume 15., New York, NY, USA, ACM Press (1996) 43–54
7. Shade, J., Gortler, S., He, L., Szeliski, R.: Layered depth images. In: SIGGRAPH '98. Volume 17., New York, NY, USA, ACM Press (1998) 231–242
8. Kanade, T., Rander, P., Narayanan, P.: Virtualized reality: Constructing virtual worlds from real scenes. IEEE Multimedia, Immersive Telepresence **4** (1997) 34–47
9. Todt, S.S., Rezk-Salama, C., Kolb, A.: Real time fusion of depth and light field images (2005)
10. Burkert, T., Leupold, J., Passig, G.: Hardware accelerated texture extraction for a photo-realistic predictive display. In: Vision, Modeling, and Visualization 2003. (2003) 11–18
11. Segal, M., Akeley, K.: The OpenGL Graphics System version 2.0 (2004)
12. Samet, H.: The quadtree and related hierarchical structures. Computing Surveys **16** (1984)
13. Edelson, D., Pohl, I.: Smart pointers: They're smart but they're not pointers. In: Usenix C++ Conference. (1991)
14. Heckbert, P., Moreton, H.: Interpolation for polygon texture mapping and shading (1991)
15. Leupold, J., Behrendt, S.: The hybrid display. In: Joint International COE/HAM-SFB453 Workshop on Human Adaptive Mechatronics and High-Fidelity Telepresence. (2005) 25–30

16. Everitt, C., Rege, A., Cebenoyan, C.: Hardware shadow mapping. In: SIGGRAPH '02, New York, NY, USA, ACM Press (2002)
17. Kilgard, M.: Shadow Mapping with Today's OpenGL Hardware. In: CEDEC Tokyo. (2001)
18. Horaud, R., Dornaika, F.: Hand-eye calibration. Intern. Journal of Robotics Research **14** (1995) 195–210
19. International Organization for Standardization: ISO/IEC 14496-1:1999: Information technology — Coding of audio-visual objects — Part 1: Systems. (1999)

Arithmetic Discrete Parabolas

I. Debled-Rennesson, E. Domenjoud, and D. Jamet

LORIA, Technopole de Nancy-Brabois, BP 239, F-54506 Vandœuvre-lès-Nancy Cedex
{Isabelle.Debled-Rennesson, Eric.Domenjoud, Damien.Jamet}@loria.fr

Abstract. In the present paper, we propose a new definition of discrete parabolas, the so-called *arithmetic discrete parabolas*. We base our approach on a non-constant thickness function and characterized the 0-connected and 1-connected parabolas in terms of thickness function. This results extend the well-known characterization of the κ-connectedness of arithmetic discrete lines, depending on the norm $\|\cdot\|_\infty$ and $\|\cdot\|_1$ of their normal vector.

1 Introduction

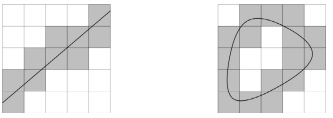
Discrete geometry attempts to provide the geometry on the grid $\mathbb{Z}^n$ with a theoretical framework, intuitively close to the Euclidean geometry. For the last 30 years, the discrete lines, namely the analogues of euclidean lines in the discrete space $\mathbb{Z}^2$, have been widely studied. On the one hand, J. Bresenham [1], H. Freeman [2] and A. Rosenfeld [3] followed an algorithmic approach and defined discrete lines as digitizations of euclidean ones. On the other hand, J.-P. Reveillès initiated the arithmetic discrete geometry [4] and introduced the arithmetic discrete lines as subsets of $\mathbb{Z}^2$ satisfying a double diophantine inequality.

The arithmetic definition of discrete lines allowed important improvements in the understanding of theses objects. For instance, it permits to obtain efficient algorithms for recognition or construction of arithmetic discrete lines [5].

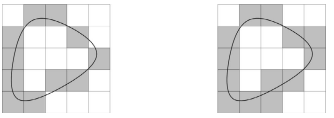
In [6], D.Cohen-Or and A. E. Kaufman introduced a general scheme of digitization, namely, the *supercover*. Roughly speaking, given a subset $A \subseteq \mathbb{R}^2$, its supercover digitization is the set of pixels (unit squares centered on a integer point) intersected by A. Equivalently, the supercover of a set A is obtained by sliding all along A a square of radius $1/2$ and centered on the points of A (see Figure 1(a)). There exist several extensions of the supercover model [7], namely the *naive closed model* and the *Pythagorean closed model*. While the supercover model of a set A is the set of pixels within a distance smaller that $1/2$ for the norm $\|\cdot\|_\infty$, the *naive closed model* (resp. the *Pythagorean closed model*) of a set A is the set of pixels within a distance smaller that $1/2$ for the norm $\|\cdot\|_1$ (resp. $\|\cdot\|_2$) (see Figure 1(b)).

As far as we know, only linear objects, namely the lines (and hyperplanes) and circles have been analytically described [7,8,9]. In the present paper, we deal with straight parabolas and give, for them, an arithmetic discrete definition based on a non-constant thickness depending on the gradient.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 480–489, 2006.
© Springer-Verlag Berlin Heidelberg 2006



(a) Supercover of a segment and a curve.



(b) Left to right: The naive and the Pythagorean close model of a set.

Fig. 1. Supercover, naive and Pythagorean model sets

2 Arithmetic Discrete Lines and Arithmetic Discrete Circles

Let us first recall some basic notions we use all along the present paper. Let $\{\mathbf{e_1}, \mathbf{e_2}\}$ denote the canonical basis of the euclidean vector space $\mathbb{R}^2$. One represents a point $\mathbf{x} \in \mathbb{R}^2$ by the pair $(x_1, x_2) \in \mathbb{R}^2$ of its coordinates in the basis $\{\mathbf{e_1}, \mathbf{e_2}\}$. For $\alpha \in [1, +\infty[$,

$$\begin{array}{rcl} \|\cdot\|_\alpha : \mathbb{R}^2 & \longrightarrow & \mathbb{R}_+ \\ \mathbf{x} & \longmapsto & (|x_1|^\alpha + |x_2|^\alpha)^{\frac{1}{\alpha}}, \end{array} \quad \text{and} \quad \begin{array}{rcl} \|\cdot\|_\infty : \mathbb{R}^2 & \longrightarrow & \mathbb{R}_+ \\ \mathbf{x} & \longmapsto & \max\{|x_1|, |x_2|\}. \end{array}$$

For $\kappa \in \{0, 1\}$, two points $\mathbf{x} \in \mathbb{Z}^2$ and $\mathbf{x'} \in \mathbb{Z}^2$ are said to be *κ-adjacent* if $\|\mathbf{x} - \mathbf{x'}\|_\infty = 1$ and $\|\mathbf{x} - \mathbf{x'}\|_1 \leq 2 - \kappa$. In other words, $\mathbf{x} \in \mathbb{Z}^2$ and $\mathbf{x'} \in \mathbb{Z}^2$ are *κ-adjacent* if their are distinct, the differences between their coordinates are at most 1 and $\mathbf{x}$ and $\mathbf{x'}$ have at most (resp. at least) $2 - \kappa$ different (resp. κ identical) coordinates. A *κ-path* is a (finite or infinite) sequence of consecutive κ-adjacent pixels. If $(\gamma_i)_{1 \leq i \leq n}$ is a finite κ-path, then we say that γ *links* the pixel γ_1 to the pixel γ_n. A subset $E \subseteq \mathbb{Z}^2$ is said to be *κ-connected* if, for each pair of pixels $(\mathbf{x}, \mathbf{x'}) \in E^2$, there exists a κ-path in E linking $\mathbf{x}$ to $\mathbf{x'}$.

2.1 Arithmetic Discrete Lines

Definition 1 (Arithmetic discrete lines [4]). *The* arithmetic discrete line *with* normal vector $\mathbf{v} \in \mathbb{R}^2$, translation parameter $\mu \in \mathbb{R}$ *and* arithmetic thickness $w \in \mathbb{R}_+$ *is the subset* $\mathfrak{D}(\mathbf{v}, \mu, w)$ *of* $\mathbb{Z}^2$ *defined by:*

$$\mathfrak{D}(\mathbf{v}, \mu, w) = \left\{ \mathbf{x} \in \mathbb{Z}^2,\ -\frac{w}{2} \leq \langle \mathbf{v}, \mathbf{x} \rangle + \mu < \frac{w}{2} \right\}, \tag{1}$$

where $\langle \mathbf{v}, \mathbf{x} \rangle = v_1 x_1 + v_2 x_2$ *denotes the usual euclidean scalar product in* $\mathbb{R}^2$. *If* $w = \|\mathbf{v}\|_\infty$ *(resp.* $w = \|\mathbf{v}\|_1$*), the arithmetic discrete line* $\mathfrak{D}(\mathbf{v}, \mu, w)$ *is said to be* naive *(resp.* standard*).*

The naive and standard arithmetic discrete lines play a key role within the arithmetic discrete lines. Indeed:

Theorem 1 ([4]). *Let* $\mathfrak{D}(\mathbf{v}, \mu, w)$ *be the arithmetic discrete line with normal vector* $\mathbf{v} \in \mathbb{R}^2$, *translation parameter* $\mu \in \mathbb{R}$ *and arithmetic thickness* $w \in \mathbb{R}_+$. *Then* $\mathfrak{D}(\mathbf{v}, \mu, w)$ *is* 0*-connected (resp.* 1*-connected) if and only if* $w \geq \|\mathbf{v}\|_\infty$ *(resp.* $w \geq \|\mathbf{v}\|_1$*).*

In order to introduce our approach, let us establish a notation.
NOTATION. — Let $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ be a differentiable function (most of the ones we consider in the following are polynomial functions). We denote by $\nabla f : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ the *gradient* function of f defined by:

$$\begin{aligned} \nabla f : \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ \mathbf{x} &\longmapsto \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \frac{\partial f}{\partial x_2}(\mathbf{x}) \right) \end{aligned}$$

With this notation, Theorem 1 can be reformulated as follows:

Theorem 2. *Let $\mathfrak{D}(\mathbf{v}, \mu, w)$ be the arithmetic discrete line with normal vector $\mathbf{v} \in \mathbb{R}^2$, translation parameter $\mu \in \mathbb{R}$ and arithmetic thickness $w \in \mathbb{R}_+$. Then $\mathfrak{D}(\mathbf{v}, \mu, w)$ is 0-connected (resp. 1-connected) if and only if $w \geq \frac{\|\nabla f(\mathbf{x})\|_\infty}{2}$ (resp. $w \geq \frac{\|\nabla f(\mathbf{x})\|_1}{2}$).*

In other words, the κ-connectedness ($\kappa \in \{0, 1\}$) of a discrete line is entirely characterized by the norm of the gradient of the affine map defining the considered arithmetic discrete line. Moreover, the arithmetic definition of discrete lines, with two diophantine inequalities permits to design efficient (incremental) algorithms to construct and recognize such discrete lines [5].

It thus becomes natural to try to extend these results to other non-linear discrete objects. Moreover, such properties on non-linear discrete objects would provide us with new discrete primitives extending the class of arithmetic discrete lines.

Let us recall in the following section a first work extending J.-P. Reveillès' approach.

2.2 Arithmetic Discrete Circles

In [9], C. Fiorio, D. Jamet and J.-L. Toutant provided a first extension of J.-P. Reveillès' arithmetic discrete lines to discrete circles. They proposed an arithmetic definition of the well-known Bresenham's circles and gave a complete characterization of the 0-connected and 1-connected discrete circles. Let us recall the main result of their work.

Definition 2 (Arithmetic discrete circles). *Let $\mathbf{X} \in \mathbb{Z}^2$, $R \in \mathbb{N}$, $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ be the function defined by $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{X}\|_2^2 - R^2$ and let $w : \mathbb{R}^2 \longrightarrow \mathbb{R}$. The* arithmetic discrete circle *with* center $\mathbf{X} \in \mathbb{Z}^2$, radius $R \in \mathbb{N}$ *and* thickness w *is the subset* $\mathrm{C}(\mathbf{X}, R, w)$ *defined by:*

$$\mathrm{C}(\mathbf{X}, R, w) = \left\{ \mathbf{x} \in \mathbb{Z}^2,\ -\frac{w(\mathbf{x})}{2} \leq f(\mathbf{x}) < \frac{w(\mathbf{x})}{2} \right\}$$

Let us notice that in the latter definition, the thickness w is not constant anymore but a function of $\mathbf{x}$. See Theorem 3 below for an example of such a function.

Theorem 3 ([9]). *Let* $\mathbf{X} \in \mathbb{Z}^2$, $R \in \mathbb{N}$, $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ *be the function defined by* $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{X}\|_2^2 - R^2$. *The arithmetic discrete circle* $\mathrm{C}(\mathbf{X}, R, \|\nabla f\|_\infty)$ *(resp.* $\mathrm{C}(\mathbf{X}, R, \|\nabla f\|_1)$*) is* 0*-connected (resp.* 1*-connected).*

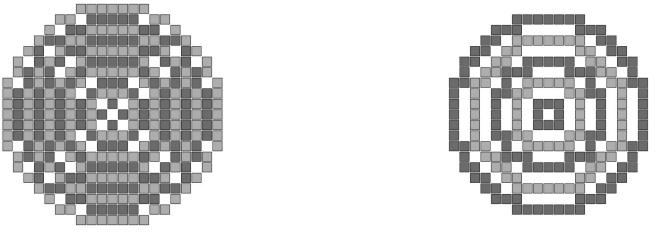


Fig. 2. A 0-connected discrete circle — a 1-connected discrete circle

In the present paper, we introduce an arithmetic definition of discrete parabolas. Similarly to the discrete circles of [9], the thickness of those discrete parabolas is defined by a norm of the gradient of its implicit equation.

3 Arithmetic Discrete Parabolas

Due to lack of space, in the present work, we only study *straight* parabolas, that are, the ones the axis of which is parallel to the y axis.

3.1 Definitions

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a polynomial function of degree 2 in one indeterminate. One associates with f the polynomial function in two indeterminates $\widetilde{f} : \mathbb{R}^2 \longrightarrow \mathbb{R}$ defined by $\widetilde{f}(\mathbf{x}) = f(x_1) - x_2$ and

$$\begin{aligned} \nabla\widetilde{f} : \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ \mathbf{x} &\longmapsto \left(\frac{df}{dx}(x_1), -1\right) \end{aligned}$$

Following the approach of the previous section, one defines the arithmetic discrete parabolas as follows:

Definition 3 (Arithmetic discrete parabolas). *Let* $f : \mathbb{R} \longrightarrow \mathbb{R}$ *be a polynomial function of degree 2 in one indeterminate and let* $\widetilde{f} : \mathbb{R}^2 \longrightarrow \mathbb{R}$ *defined by* $\widetilde{f}(\mathbf{x}) = f(x_1) - x_2$. *Let* $w : \mathbb{R}^2 \longrightarrow \mathbb{R}$ *a map. The* arithmetic discrete parabolas *with* equation $\widetilde{f}(\mathbf{x}) = 0$ *and* thickness w *is the subset* $\mathrm{P}(\widetilde{f}, w)$ *of* $\mathbb{Z}^2$ *satisfying:*

$$\mathrm{P}(\widetilde{f}, w) = \left\{ \mathbf{x} \in \mathbb{Z}^2,\ -\frac{w(\mathbf{x})}{2} \leq \widetilde{f}(\mathbf{x}) < \frac{w(\mathbf{x})}{2} \right\}$$

The main advantage of such a definition, is that it provides us with a quite easy way to test whether a pixel belongs to a given arithmetic discrete parabolas. As

far as we know, only arithmetic discrete lines [4] and arithmetic discrete circles [9] provide such a test.

Of course, one can consider any kind of functions for the thickness w. In the following, we investigate particular cases of thickness functions, namely the ones homothetic to $\|\nabla\widetilde{f}\|_\alpha$, with $\alpha \in [1, +\infty]$.

For clarity issue, let us introduce a notation and a terminology:
NOTATION. — Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a polynomial function in $\mathbb{R}[x]$ of degree 2: there exists $(a, b, c) \in \mathbb{R}^3$ such that, for all $x \in \mathbb{R}$, $f(x) = ax^2 + bx + c$. Let $\alpha \in [1, +\infty]$ and let $\lambda \in \mathbb{R}_+$. The *arithmetic discrete parabola* of *equation* f, *thickness* $\|\cdot\|_\alpha$ and *thickness factor* λ is the set $\mathrm{P}(\widetilde{f}, \lambda\|\nabla\widetilde{f}\|_\alpha)$. For short, we denote the set $\mathrm{P}(\widetilde{f}, \lambda\|\nabla\widetilde{f}\|_\alpha)$ by $\mathrm{P}(f, \alpha, \lambda)$.

This family has the nice property that the κ-connectedness of its elements can be easily determined with respect to their thickness (see Theorem 4). For that reason, we state the following definition.

Definition 4 (Naive and standard discrete parabolas). *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a polynomial of degree 2 in one indeterminate over $\mathbb{R}$. The arithmetic discrete parabola* $\mathrm{P}(f, \infty, 1)$ *(resp.* $\mathrm{P}(f, 1, 1)$*) is said to be* naive *(resp.* standard*) (see Figure 3).*



Fig. 3. A naive and a standard arithmetic discrete parabola for $\widetilde{f}(\mathbf{x}) = 7x_1^2 + 4x_1 + 21 - 56x_2$. Notice that in the general case, a arithmetic do not have any symmetry.

3.2 Topological Properties of Arithmetic Discrete Parabolas

Leaning curves: Let $\mathrm{P}(f, \alpha, \lambda)$ be an arithmetic discrete parabola. With no loss of generality, up to a symmetry, let us state $f(x) = ax^2 + bx + c$ with $a \in \mathbb{R}_+$. Then the pixels of $\mathrm{P}(f, \alpha, \lambda)$ are exactly the ones located between the curves $f(x) \pm \lambda\|\nabla\widetilde{f}(\mathbf{x})\|_\alpha$ (see Figure 4(a) and Figure 4(b) for the case of the usual discrete norms $\|\nabla\widetilde{f}(\mathbf{x})\|_1$ and $\|\nabla\widetilde{f}(\mathbf{x})\|_\infty$). Let us consider the case of $\mathrm{P}(f, \infty, \lambda)$. Then, for all $\mathbf{x} \in \mathbb{Z}^2$,

$$\mathbf{x} \in \mathrm{P}(f, \infty, \lambda) \iff -\lambda\frac{\max\{|2ax_1 + b|, 1\}}{2} \leq f(x_1) - x_2 < \lambda\frac{\max\{|2ax_1 + b|, 1\}}{2},$$

and the set $\mathrm{P}(f, \infty, \lambda)$ is bounded by the four curves of equation: $f(x_1) - x_2 = \pm\lambda\frac{2ax_1+b}{2}$ and $f(x_1) - x_2 = \pm\frac{\lambda}{2}$. More precisely, the *upper and lower leaning curves* are respectively constituted with three parts of parabolas :

$$l_\infty^U(x_1) = \begin{cases} f(x_1) - \frac{\lambda}{2}(2ax_1+b), & x \in]-\infty, \frac{-1-b}{2a}[\\ f(x_1) + \frac{\lambda}{2}, & x \in [\frac{-1-b}{2a}, \frac{1-b}{2a}] \\ f(x_1) + \frac{\lambda}{2}(2ax_1+b), & x \in]\frac{1-b}{2a}, \infty[\end{cases}$$

$$l_\infty^L(x_1) = \begin{cases} f(x_1) + \frac{\lambda}{2}(2ax_1+b), & x \in]-\infty, \frac{-1-b}{2a}[\\ f(x_1) - \frac{\lambda}{2}, & x \in [\frac{-1-b}{2a}, \frac{1-b}{2a}] \\ f(x_1) - \frac{\lambda}{2}(2ax_1+b), & x \in]\frac{1-b}{2a}, \infty[\end{cases}$$

In the same way, one obtains that the set $\mathrm{P}(f,1,\lambda)$ is bounded by the curves of equation $f(x_1) - x_2 = \frac{\pm(2ax_1+b)\ \pm\lambda}{2}$ and that the set $\mathrm{P}(f,2,\lambda)$ is bounded by the curves $f(x_1) - x_2 = \pm\lambda\frac{\sqrt{(2ax_1+b)^2+1}}{2}$.

Connectedness of arithmetic discrete parabolas

Theorem 4. *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a polynomial function of degree 2 in one indeterminate over $\mathbb{R}$, let $\alpha \in [1, +\infty]$ and let $\lambda \in [1, +\infty[$.*

(i) The arithmetic discrete parabola $\mathrm{P}(f,\alpha,\lambda)$ is 0-connected.
(ii) The arithmetic discrete parabola $\mathrm{P}(f,1,\lambda)$ is 1-connected.

In order to prove Theorem 4, we use the following lemmas (due to lack of space, we do not give any proof of them):

Lemma 1. *Let $I \subseteq \mathbb{R}$ be an interval and let $f : I \to R$ be a function satisfying, $\forall x, x' \in I,\ |f(x') - f(x)| \leq |x' - x|$. Then the set of integer points $\mathbf{x} = (x_1, x_2) \in \mathbb{Z}^2$ such that $f(x_1) \leq x_2 < f(x_1) + 1$ is 0-connected.*

Lemma 2. *For any $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\lambda \in \mathbb{R}^+$, let $B(\mathbf{x}, \lambda)$ denote the semi-open ball $B(\mathbf{x},\lambda) = \left\{\mathbf{x'} \in \mathbb{R}^2, -\frac{\lambda}{2} \leq x'_1 - x_1 < \frac{\lambda}{2} \text{ and } -\frac{\lambda}{2} \leq x'_2 - x_2 < \frac{\lambda}{2}\right\} \cup \{(x_1 + \frac{\lambda}{2}, x_2 - \frac{\lambda}{2})\}$. Then, for any path-connected set $S \subseteq \mathbb{R}^2$ and for any $\lambda \geq 1$, the set*

$$S \oplus B(0,\lambda) = \mathbb{Z}^2 \cap \bigcup_{\mathbf{x} \in S} B(\mathbf{x},\lambda)$$

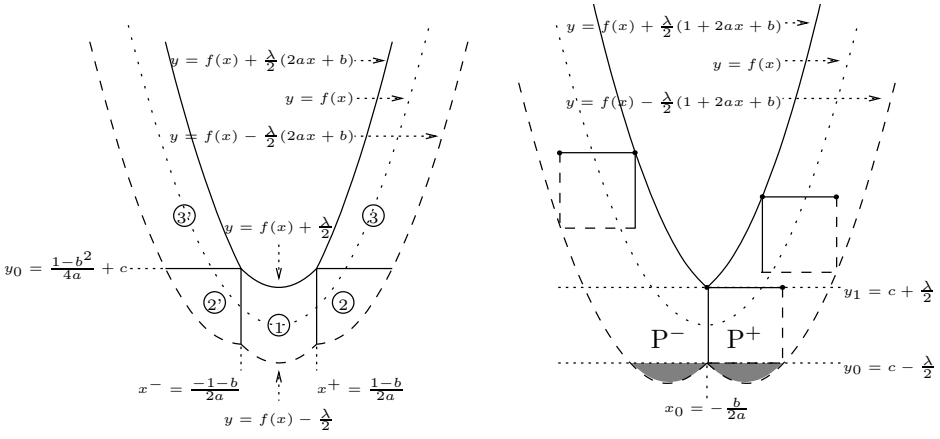
is 1-connected.

Lemma 3. *Let $I \subseteq \mathbb{R}$ be an interval and let f, g, f_1, g_1 be real functions from I to $\mathbb{R}$ satisfying:*

(i) for all $x_1 \in I$, $f_1(x_1) \leq f(x_1)$ and $g(x_1) \leq g_1(x_1)$;
(ii) for all $x_1 \in I \cap \mathbb{Z}$, there exists $x_2 \in \mathbb{Z}$ such that $f(x_1) < x_2 \leq g(x_1)$.

If $\mathbb{Z}^2 \cap \{\mathbf{x} = (x_1, x_2),\ x_1 \in I \text{ and } f(x_1) < x_2 \leq g(x_1)\}$ is 0-connected (resp. 1-connected) then so is $\mathbb{Z}^2 \cap \{\mathbf{x} = (x_1, x_2),\ x_1 \in I \text{ and } f_1(x_1) < x_2 \leq g_1(x_1)\}$.

Proof. (of Theorem 4) With no loss of generality, we assume $a > 0$ in $f(x) = ax^2 + bx + c$ (if $a = 0$, the function f is affine and the corresponding arithmetic discrete parabola is an arithmetic discrete line).



(a) $\mathrm{P}(f, \infty, \lambda)$ (b) $\mathrm{P}(f, 1, \lambda)$

Fig. 4. Tessellation of $\mathrm{P}(f, \infty, \lambda)$ and $\mathrm{P}(f, 1, \lambda)$

i) Let us first prove that $\mathrm{P}(f, \infty, 1)$ is 0-connected. The general case directly follows. For any $\lambda \in [1, +\infty[$, $\mathrm{P}(f, \infty, \lambda)$ is divided into 5 areas labeled 1, 2, 2′, 3 and 3′ (see Figure 4(a) below). Area 1 is the set of points $\mathbf{x} \in \mathbb{Z}^2$ satisfying $-x^- \leq x_1 \leq x^+$, $f(x_1) - \frac{\lambda}{2} < x_2 \leq f(x_1) + \frac{\lambda}{2}$. For $\lambda = 1$, the vertical distance between the upper and lower delimiting curves is exactly 1. Since the absolute value of the slope of these curves is at most 1, this area is 0-connected by lemma 1. We prove in the same way that areas 3 and 3' are 0-connected by noticing that the horizontal distance between the delimiting curves is exactly 1 and the absolute value of the slope is at least 1 and by exchanging the roles of x_1 and x_2. For area 2, we distinguish 2 cases:

   (a) If area 2 is not empty then it contains (at least) the integer point $(\lceil x^+ \rceil, \lfloor y_0 \rfloor)$ (see Figure 4(a)). A somehow tedious case analysis shows that area 1 or 2' or even 3' contains the point $(\lceil x^+ \rceil - 1, \lfloor y_0 \rfloor - 1)$ or $(\lceil x^+ \rceil - 1, \lfloor y_0 \rfloor)$ or $(\lceil x^+ \rceil - 1, \lfloor y_0 \rfloor + 1)$ and area 3 contains the point $(\lceil x^+ \rceil, \lfloor y_0 \rfloor + 1)$ or $(\lceil x^+ \rceil + 1, \lfloor y_0 \rfloor + 1)$.
   (b) In case area 2 is empty can be similarly investigated by a technical but non-complicated enumeration of cases.

   Part *(i)* of Theorem 4 directly follows from Lemma 3 by noticing that for any $\alpha \in [1, \infty]$ and any $\lambda \in [1, +\infty[$, one has $\lambda \|\nabla \widetilde{f}\|_\alpha \geq \|\nabla \widetilde{f}\|_\infty$.

ii) Once again, we prove that $P(f, 1, 1)$ is 1 connected (see Figure 4(b)) and we use the same argument as above to obtain the result for any $\lambda \geq 1$. It is easily shown that the subset P^+ bounded by the leaning curves of $\mathrm{P}(f, 1, 1)$ and satisfying $x \geq x_0$ and $y > y_0$ is exactly

$$\mathrm{P}^+ = \bigcup_{\mathbf{x} \in S} B(\mathbf{x}, \lambda) \quad \text{where} \quad S = \left\{ (x_1, f(x_1) - a\frac{\lambda^2}{4}) \,|\, x_1 \geq x_0 \right\}.$$

Hence P^+ is 1-connected by Lemma 2. By symmetry, the set P^- (bounded by the leaning curves of $\mathrm{P}(f,1,1)$ and satisfying $x \leq x_0$ and $y > y_0$) is 1-connected too.
Now either $x_0 \in \mathbb{Z}$ and $(x_0, \lfloor y_1 \rfloor) \in P^+ \cap P^-$ or $x_0 \notin \mathbb{Z}$ and $(\lfloor x_0 \rfloor, \lfloor y_1 \rfloor) \in P^-$ and $(\lfloor x_0 \rfloor + 1, \lfloor y_1 \rfloor) \in P^+$. In both cases, $\mathrm{P}^- \cap \mathrm{P}^+$ is 1-connected. We get the 1-connectedness of the whole parabola by an immediate application of lemma 3. ■

It directly follows from Theorem 4:

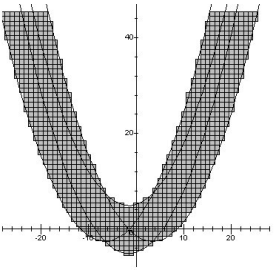
Corollary 1. *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a polynomial function of degree 2 in one indeterminate over $\mathbb{R}$. The naive arithmetic discrete parabola $\mathrm{P}(f,\infty,1)$ (resp. standard arithmetic discrete parabola $\mathrm{P}(f,1,1)$) is 0-connected (resp. 1-connected).*

This property is a quite natural extension of Theorem 1 concerning the arithmetic discrete lines.

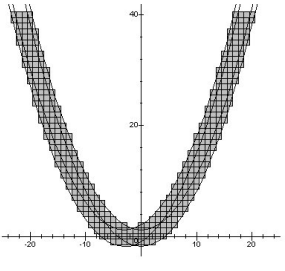
On the thickness factor: We remark that for a given discrete parabola $\mathrm{P}(f,\alpha,\lambda)$, the thickness factor λ could imply a deformation of the graphical representation of P (see Figure 4(a) and Figure 4(b)).

If $\alpha = \infty$, a sufficient condition to obtain a *good* graphical representation is that the abscissa of the top of the first parabola defining the lower leaning curve of P is greater than $\frac{-1-b}{2a}$. This condition corresponds to $\lambda \leq \frac{1}{a}$ (see Figure 5(a)).

If $\alpha = 1$, a sufficient condition to obtain a *good* graphical representation is: the difference between the ordinate of the top of one parabola defining the lower leaning curve of P and the ordinate of the intersection point of both parabolas defining the lower leaning curve is less or equal to 0.5. This condition corresponds to $\lambda^2 \leq \frac{2a-b^2}{a^2}$ (see Figure 5(b)).



(a) $\mathrm{P}(f,\infty,9)\ldots$



(b) $\mathrm{P}(f,1,3)\ldots$

Fig. 5. ... with $f(x) = 0.1x^2 + 0.3x + 0.1$

3.3 Construction Algorithm

We propose in this section two incremental construction algorithms of an arithmetic discrete parabola $P(f,\alpha,\lambda)$. The first one is presented for a parabola with a thickness factor equal to one and the second one in the general case.

$P(f, \alpha, 1)$ construction algorithm
This algorithm relies on Corollary 1. We consider a discrete parabola $P(f, \alpha, 1)$ such that $\widetilde{f}(\mathbf{x}) = ax_1^2 + bx_1 + c + dx_2$ with $(a, b, c, d) \in \mathbb{Z}^4$, $a < 0$ and $d > 0$. We remark that:

$$\widetilde{f}(x_1 + 1, x_2) = \widetilde{f}(x_1, x_2) + 2ax_1 + a + b$$
$$\widetilde{f}(x_1, x_2 \pm 1) = \widetilde{f}(x_1, x_2) \pm d$$

Algorithm 1 is directly deduced from the previous remarks.

Algorithm 1. Incremental construction algorithm of $P(f, \alpha, 1)$

Input : a, b, c, d the characteristics of $P(f, \alpha, 1)$, N number of points, (p_x, p_y) first point
Initialization: $x = p_x$; $y = p_y$; $F = ax^2 + bx + c + dy$;
Display pixel (x, y);
while $n <= N$ **do**
  if $x < -\frac{b}{2a}$ **then** $s_y = 1$;**else** $s_y = -1$;
  if $F + 2ax + a + b \geq -\frac{\|\nabla \tilde{f}\|_\alpha(x+1)}{2}$ *and* $F + 2ax + a + b < \frac{\|\nabla \tilde{f}\|_\alpha(x+1)}{2}$ **then**
    $F = F + 2ax + a + b$;
    $x = x + 1$;
  else
    if $F + s_y d \geq -\frac{\|\nabla \tilde{f}\|_\alpha(x)}{2}$ *and* $F + s_y d < \frac{\|\nabla \tilde{f}\|_\alpha(x)}{2}$ **then**
      $F = F + s_y d$;
      $y = y + s_y$;
    else
      $F = F + 2ax + a + b + s_y d$;
      $x = x + 1$;$y = y + s_y$;
    end
  end
  Display pixel (x, y);
  $n = n + 1$;
end

Sketch of $P(f, \alpha, \lambda)$ construction algorithm: The main idea of this algorithm is to use the two leaning curves of $P(f, \alpha, \lambda)$. For each value of abscissa, we calculate the real points located on the lower and upper leaning curves of P. We then deduced incrementally the sequence of integer points located between these two points with the same abscissa.

4 Conclusion and Perspective

In the present paper, we presented a new approach and definition of discrete parabolas. We have seen that this approach depends on a non-constant thickness, contrary to arithmetic discrete lines [4] and regular circles [8]. In a forthcoming work, we plan to study the recognition problem of such discrete primitives, and the investigation of the other conics, namely the hyperbolas and the ellipses. In

Section 3.2, we showed how the thickening of discrete parabola can make strange shapes appear. An interesting way of investigation is the definition of another thickness function depending on the analytical properties of the parabolas. Afterward, we will investigate the general case of algebraic discrete function of any degrees.

References

1. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Systems Journal **4**(1) (1965) 25–30
2. Freeman, H.: Computer processing of line-drawing images. ACM Computing Surveys **6**(1) (1974) 57–97
3. Rosenfeld, A.: Digital straight lines segments. In: IEEE Transactions on Computers. (1974) 1264–1369
4. Reveillès, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique. Thèse d'état, Université Louis Pasteur, Strasbourg (1991)
5. Debled-Rennesson, I., Reveillès, J.P.: A linear algorithm for segmentation of digital curves. IJPRAI **9**(4) (1995) 635–662
6. D.Cohen-Or, Kaufman, A.E.: Fundamentals of surface voxelization. CVGIP: Graphical Model and Image Processing **57**(6) (1995) 453–461
7. Andres, E., Nehlig, P., Françon, J.: Supercover of straight lines, planes and triangles. In: Discrete Geometry for Computer Imagery, 7th International Workshop. Volume LNCS n°1347 of Lecture Notes on Computer Science., Springer Verlag (1997) 243–257
8. Andres, E.: Modélisation Analytique Discrète d'Objets Géométriques. Habilitation à diriger des recherches, Université de Poitiers (2000)
9. Fiorio, C., Jamet, D., Toutant, J.L.: Discrete circles: an arithmetical approach with non- constant thickness. In Longin Jean Latecki, David M. Mount, A.Y.W., ed.: Vision Geometry XIV, Electronic Imaging, SPIE. Volume 6066., San Jose (CA), USA, SPIE (2006) 0C–1–0C–12

Retinal Spot Lesion Detection Using Adaptive Multiscale Morphological Processing

Xin Zhang and Guoliang Fan

School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK 74078

Abstract. We present a new spot lesion detection algorithm for retinal images with background diabetic retinopathy (DR) pathologies. The highlight of this algorithm is its capability to deal with all DR-related spot lesions of various sizes and shapes that is accomplished by a unique adaptive multiscale morphological processing technique. A scale map is generated to delineate lesion areas based an edge model, and it is used to fuse multiscale morphological processing results for lesion enhancements. The local/releative entropy thresholding techniques are employed to segment lesion regions, and a scale-guided validation process is used to remove over-detections based on the scale map. The proposed algorithm is tested on 30 retinal images where all spot lesions are hand-labelled for performance evaluation. Compared with two existing algorithms, the proposed one significantly improves the overall performance of spot lesion detection producing higher sensitivity and/or predictive values.

1 Introduction

Diabetic Retinopathy (DR) is the leading cause of blindness among working-age Americans. It is shown that early diagnosis and timely treatment could efficiently prevent visual loss [1]. To standardize the procedure of DR diagnosis, the National Eye Institute (NEI) of the National Institutes of Health (NIH) has developed Early Treatment Diabetic Retinopathy Study (ETDRS) protocols which have become the "gold–standard" for evaluating DR [2]. According to ETDRS, DR-related abnormalities can be divided into three major classes: *abnormal spot class*, *abnormal blood vessel class*, and *abnormal stereo measurement*. The spot lesions are usually the preliminary sign of DR, e.g., microaneurysms, hemorrhages, drusen, soft exudates (cotton wool), hard exudates, and vitreous hemorrhage. Automated spot lesion detection can greatly facilitate the computer-aided grading process. Specifically, microaneurysm (MA) is a red spot, $125\mu m$ in its longest dimension. Hemorrhages (HE) have irregular shapes, red color and and are larger than MA. Hard exudates (HardEX) are small white or yellowish-white deposits with sharp margins. Soft exudates (SoftEX) are round or oval in shape, white or pale yellow-white in color, and have ill-defined edges. Drusen appears as deep, yellowish-white dots with blurred edges. Fig. 1 shows MA, HE, HardEX, SoftEX and drusen. Based on their intensities, we name MA and HE as *dark lesions* and HardEx, SoftEx, and drusen as *bright lesions*.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 490–501, 2006.
© Springer-Verlag Berlin Heidelberg 2006

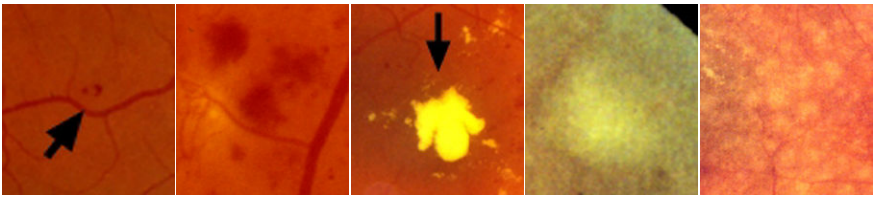


Fig. 1. Examples of spot lesions: MA, hemorrhage, HardEx, SoftEx, and drusen

There are three major challenges in the retinal lesion detection, i.e., (1) poor visibility of lesions due to non-uniform intensity distribution and low contrast with background, (2) lesion variability with various sizes, shapes, and intensities, and (3) the vascular tree as a major distracting element for dark lesion detection. Many algorithms were proposed for spot lesion detection. Spencer *et. al.* [3] proposed a framework to detect microaneurysm in fluorescein angiographic images where lesions have better visibility than in optical images. Bilinear top-hat morphological transformation with a specific size structuring element is employed to enhance MA and depress blood vessels. Region growing and feature based classification methods finalize the results. Niemeijier *et. al.* [4] proposed a method to detect red lesions, i.e., microaneurysm and hemorrhage, based on a hybrid approach combining two prior works by Spencer *et. al.* [3] and Frame *et. al.* [5] with pixel-wise classification and additional new features. The system achieves 100% sensitivity and 87% specificity. Walter *et. al.* [6] proposed an algorithm to extract hard exudates using their high gray level variations. The contours are found by thresholding the difference between original grayscale image and a morphologically reconstructed one. It reached 92.8% sensitivity and 92.4% predictive value. Sbeh *et. al.* [7] used morphological reconstruction to determine drusen's position. Adaptive threshold selection provides clear and precise boundaries. Recently, Zhang *et. al.* [8] proposed a method to detect two kinds of bright lesions (exudates and cotton wool spot, also called soft exudates according to ETDRS) and one type of dark lesion (hemorrhage). In the bright lesion detection part, a bottom–up process is employed to classify bright non-lesion areas, exudates and cotton wool spots, by using Improved Fuzzy C-means prior segmentation and a hierarchical classification structure. In hemorrhage detection, a top–down approach is adopted where hemorrhages are firstly located in regions of interest (ROI) and then their boundaries are delineated in the post-processing stage.

In this work, we aim to develop a comprehensive and effective lesion detection algorithm that can handle all DR-related lesions of various sizes, shapes, and intensities in relatively low contrast retinal images. The key of this work is a unique adaptive multiscale morphological processing technique for lesion delineation. Moreover, the algorithm can extract measurable features and characteristics for all detected lesions that will be useful for ETDRS-based disease diagnosis and staging. The proposed algorithm has is robust, flexible, and applicable to other similar medical image analysis or industrial inspection applications.

2 Preliminary

2.1 Mathematical Morphology Operations

We will briefly review morphological operations used in this paper as the following, where $f(x, y)$ is a finite-support grayscale image function defined on grid Z^2 and B is a binary structuring element. Details can be found in [9].

- **Dilation.** $(f \oplus B)(x, y) = \max\{f(x - s, y - t)|(s, t) \in B\}$.
- **Erosion.** $(f \ominus B)(x, y) = \min\{f(x + s, y + t)|(s, t) \in B\}$.
- **Opening.** $f \circ B = (f \ominus B) \oplus B$.
- **Closing.** $f \bullet B = (f \oplus B) \ominus B$.
- **Bottom-Hat.** $BH(f, g) = (f \bullet g) - f = (f \oplus B) \ominus B - f$.
- **Top-Hat.** $TH(f, g) = f - (f \circ g) = f - (f \ominus B) \oplus B$.
- **Reconstruction.** $\rho_f(g) = \bigvee_{n \geq 1} \delta_f^{(n)}(g)$, where $\bigvee$ stands for the pointwise maximum, and $\delta_f^{(n)}(g) = (g \oplus nB) \bigwedge f$, where nB is the self-dilation of B $(n - 1)$ times and $\bigwedge$ denotes the point–wise minimum.

The "top-hat" filter can enhance relatively bright areas, and the "bottom-hat" is used to enhance relatively dark areas. The local maxima are used to locate bright lesions where an appropriate scale will be selected for "top-hat" filtering. The local minima, which are used to locate dark lesions, can be easily obtained by inverting the gray-scale image followed by morphological reconstruction [10].

Given a shape, the structuring element is controlled by a *scale* parameter that plays an important role in the morphological processing. We can apply a set of structuring elements of different scales to detect and extract spot lesions of various shapes and sizes. The morphological operations at multiple scales are named as *multiscale morphology* [9]. If the structuring element B is convex, we can obtain B_n, the structuring element at scale n by dilating B recursively $n - 1$ times with itself as $nB = B \oplus B \oplus \cdots \oplus B(n - 1\ times)$ where n is an integer representing the scale factor of the structuring element. The multiscale morphological operations satisfy the properties like causality (no reginal extrema and edge introduced as the scale increases) and edge location (no drift of edge introduced) [11]. How to determine the optimal scale for each pixel is the key of this research. In the following, we introduce a parametric edge model that will be combined with morphological processing to estimate the optimal scale for each pixel.

2.2 Edge Model

A 2-D edge in an image can be modelled by a 1D edge along the gradient direction. A parametric edge model allows us to formulate and analyze

edge-related problems [12,13]. A 1-D edge $s(x)$ at $x = 0$ is modelled as the *Gaussian smoothed step edge* defined by

$$s(x) \equiv s(x; b, c, w) = h(x; b, c) * g(x; w) = b + \frac{c}{2}\left(1 + erf\left(\frac{x}{w\sqrt{2}}\right)\right) \quad (1)$$

where

$$h(x; b, c) = b + cU(x); \quad g(x; w) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \text{ with } \sigma = w,$$

and w is the width of the edge, c the contrast across the edge, and b the intensity at the base of the edge, as shown in Fig. 2. The first-order derivative of the edge function is a *Gaussian function*, $cg(x; w)$, where the peak corresponds to the edge center with the largest slope. More than 95% of the edge are is between $x = -2w$ and $x = 2w$ where the slope can be computed as

$$s'(x)|_{x=\pm 2w} = \frac{c}{w\sqrt{2\pi}} \exp -\frac{(2w)^2}{2w^2} = 0.1353cg(x; w)|_{x=0} = 0.1353s'(x)|_{x=0}. \quad (2)$$

In the following, we will use the slope relationship at $x = 0$ and $x = 2w$ to determine the optimal scale for the multiscale structuring elements.

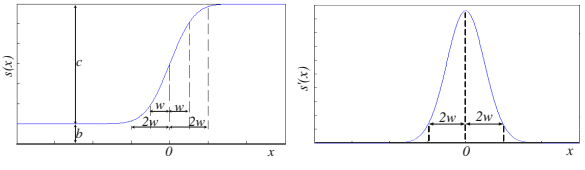


Fig. 2. The 1–D edge model (Left) and its first-order derivative (Right)

3 Proposed Algorithm

The key idea in this work is a unique combination of non-linear morphological processing with a linear parametric edge model. As shown in Fig. 3, the proposed algorithm is composed of five major steps. The first step is to select potential lesion areas by morphological reconstructions. The second step is to enhance all pathological/anatomical structures via adaptive multiscale morphological processing, where the edge model is used for optimal scale estimation. The third step is to generate a scale map that is a guide for lesion delineation and also used as a reference for later lesion validation. The forth step involves the local entropy/relative entropy thresholding techniques to create a binary lesion map, and the last step is scale-based lesion validation to remove over-detections. This work is applicable to both dark and bright lesion detections both of which just use the counterpart morphological operators. Therefore, we will focus on the dark lesion detection in the following.

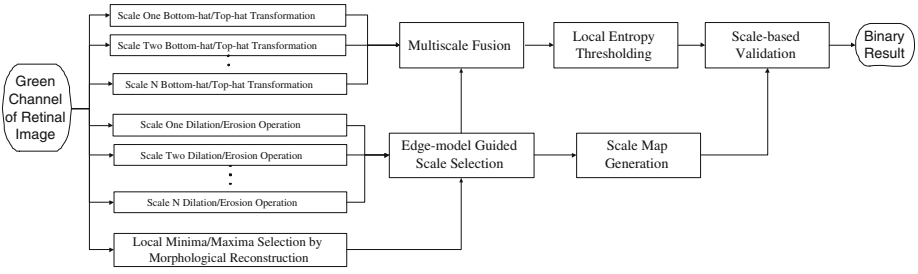


Fig. 3. Algorithm flowchart for dark/bright lesion detection

3.1 Local Minima/Maxima Extraction

Local minima are the reference points to determine the proper scale for dark lesions. Sbeh *et. al.* [7] applied adaptive morphological reconstruction to select local maxima as corresponding positions of drusen. Inspired by this idea, we use morphological reconstruction to select local minima, i.e., potential dark lesion areas. In order to include all dark lesions, the morphological reconstruction result might also have some false minima. Let the green channel image be f, then the mask image for reconstruction is $(f - \xi)$. The reconstruction result is

$$R_f(f-\xi) = \bigvee_{n \geq 1} \delta_f^{(n)}(f-\xi).$$

The choice of the parameter ξ determines the areas extracted as the starting points. The larger ξ is, the more areas are involved. $\xi = 10$ was found to be a reasonable value. Then the selected local minima are

$$S_f(\xi, \zeta) = ((f - R_f(f-\xi)) > \zeta).$$

Usually, the dark lesion has relatively low contrast compared with the bright lesion. Therefore, we use different ζ values to minimize mis-detection and over-detection for dark/bright lesions. The empirical setting of ζ is 2 for dark lesions and 6 for bright lesions, which is consistent with our observations.

3.2 Adaptive Multiscale Morphology

Given a local minimum indicating a potential dark lesion, we use a two–sided edge model to define the lesion area, as shown in Fig. 4. The structuring element of the appropriate scale should be the one just covering the whole "valley". As mentioned before, we want to use the variation of the edge slope to select the optimal scale. According to the definition, the dilation result is the maximal value within the area covered by the structuring element. Therefore, we can use multiscale dilation increments to estimate the variation of the edge slope.

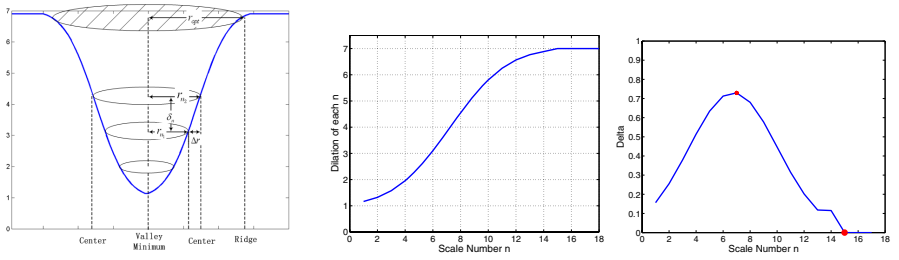


Fig. 4. The two sided edge model centered at one local minimum (left), the multiscale dilation result (center) and the dilation increment result (right)

Let n represent the scale, and the radius of the corresponding structuring element be r_n, based on the definition of multiscale morphology, the scale increase between the two consecutive structuring elements is constant, defined as

$$\triangle r = r_{n_2} - r_{n_1}, \tag{3}$$

where r_{n_2} and r_{n_1} are the radii of two consecutive scales. We define the dilation increments of two consecutive scales as,

$$\delta_n = F_n^D - F_{n-1}^D, \tag{4}$$

where F_n^D is the dilation result of scale n, and F_0^D is the original image. Then the first order derivative of the edge model can be approximated as:

$$s'(x)|_{x=r_n} \approx \lim_{\triangle r \longrightarrow 0} \frac{\delta_n}{\triangle r}. \tag{5}$$

When $\triangle r$ is one pixel, the first derivative of the edge model can be approximated by the dilation increments. The largest dilation increment, $\delta_{\max}$, indicates that the structuring element reaches half-way in the "valley". As shown before, 95% edge area is covered between $x = -2w$ and $x = 2w$, where the slope is related to the maximum slope according to (2). Therefore, the proper structuring element which just covers the whole "valley" should be the one that has the dilation increment to be $0.1353\delta_{\max}$, and the appropriate scale is determined by:

$$n_{opt} = \arg\min_n \{\delta_n^D < 0.1353\delta_{\max}^D | n = n_{\max}, \cdots, N\}, \tag{6}$$

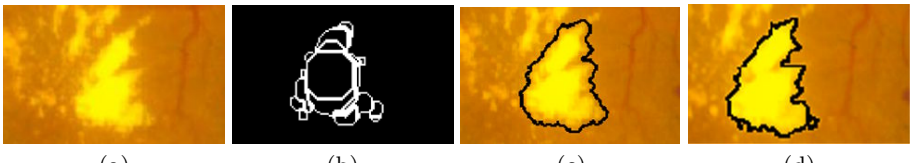
where $n_{\max}$ is the scale corresponding to $\delta_{\max}$ and N is the maximum scale.

We use Fig. 4 to illustrate how the optimal scale is selected. Given the two-sided edge in Fig. 4 (left) that is parameterized by $w = 7$ on each side, the optimal scale for this area should be around 14. We show multiscale dilation and the dilation increment results in Fig. 4 (center and right). We can see that the dilation increment reaches the maximum when $n = 7$ and the optimal scale is set to be 15 according to (6). Since we use the disc-shaped structuring element whose minimum diameter is 3, indicating $\triangle r = 1$. To cover all possible lesions, we set the scale to be ranging from 1 to 8 (or 18) for dark (or bright) lesions.

3.3 Scale Map Generation

After determining the proper scale for each local minimum, we need to assign a scale number to the other pixels in the same region. All pixels inside the "valley" are given the same scale as the local minimum because they are covered by the same structuring element. Some pixels may be assigned with different scales when there is more than one local minimum involved in the region. For that case, we assign the largest one for those pixels. In this way, any shape and size lesions can be approximated by multiple structuring elements of different scales. Pixels which are not assigned to a scale number belong to non-lesion areas, and their scales are set to be zero for convenience.

We use Fig. 5 where a bright lesion area is the region of interest to illustrate how to use the scale map to cover a lesion region. Fig. 5 (b) shows how one bright lesion is approximated by composite structuring elements of various scales that are specified the scale map. The lesion delineation and detection results (after thresholding discussed next) are shown in Fig. 5 (c) and (d). It shows that the edge model guided scale selection effectively creates a scale map to delineate the lesion. Based on the scale map, the local contrast of each local minimum will be extracted by the "bottom–hat" transformation with the optimal scale. Accordingly, the multi-channel morphological filtering results are fused into one grayscale image where all dark lesions and blood vessels are enhanced.



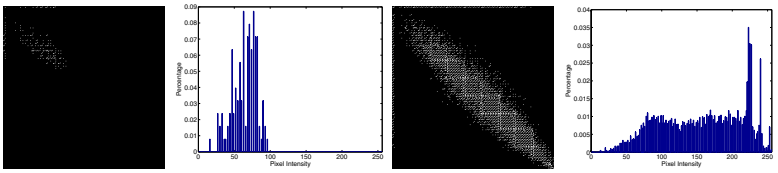
(a) (b) (c) (d)

Fig. 5. (a) A bright lesion area. (b) The composite structure elements of different scales specified by the scale map. (c) The superimposition result of (b) onto (a). (d) The lesion detection result after entropy-based thresholding.

3.4 Entropy Thresholding Techniques

After lesion enhancement by multiscale morphological processing, we want to use certain thresholding techniques to segment out pathological/antomatical structures. Therefore, we apply two entropy–based thresholding techniques to segment dark and bright lesions with the consideration of the characteristics of intensity distributions of both dark and bright lesions. Usually, pixel intensities of bright areas usually have large variation while those of dark areas are relatively concentrated, as shown in Fig. 6. Consequentially, the co-occurrence matrix of the bright lesion, which records transition probabilities of adjacent pixel intensities [14], is much sparser than that of the dark lesion.

On the one hand, the optimal threshold of the local entropy thresholding, which is defined on a co-occurrence matrix, maximizes the summation of the



(a) (b) (c) (d)

Fig. 6. The normalized intensity distributions in a lesion enhanced image. The co-occurrence matrices (with the log operation) of dark/bright lesions are shown in (a) and (c), and the histograms of dark/bright lesions are given in (b) and (d).

local entropy of background and foreground. Since large intensity variation in bright areas (bright lesions and optic nerve) leads to a sparse co-occurrence matrix, the local entropy method is suitable for bright lesion detection. On the other hand, the relative entropy thresholding is to minimize the discrepancy, i.e., the relative entropy, between the co-occurrence matrix of the original image and that of the binarized one [15]. Therefore, the thresholded image will be the best approximation to the original one. Due to the narrow intensity distribution of dark areas (dark lesions and blood vessels), the co-occurrence matrix of dark regions has strong and narrow peaks, and the relative entropy thresholding algorithm was found effective to keep all dark lesions along with blood vessels.

3.5 Scale-Guided Lesion Validation

The validation step aims to remove blood vessels and over-detections. The scale map generated previously is employed as a crucial guideline for lesion validation.

Blood Vessel Removal: This step is only necessary for dark lesion detection. We can first remove the top largest connected components in the dark lesion segmentation map to delete major vascular trees. However, some small blood vessels may still be left, and we want to use the scale map obtained to validate all lesion candidates. Let R denote a dark lesion as shown in Fig. 7 where the set of local minima included in R is represented as $\Omega(R)$ and the region covered by structuring elements, denoted as Q, is defined as:

$$Q = \bigcup_{\rho \in \Omega(R)} B_{s(\rho)}(\rho), \tag{7}$$

where ρ is a local minimum in $\Omega(R)$, $s(\rho)$ is its optimal scale, and $B_{s(\rho)}(\rho)$ is the structuring element of scale $s(\rho)$ and centered at ρ. For small blood vessels, $\#Q$ (the pixel number in Q) is much larger than $\#R$ due to their long linear shapes. In this work, if $\#R/\#Q \geq 0.60$, region R is accepted, otherwise it is removed.

Over-detection Removal: It is possible that some background regions surrounded by multiple bright lesions may be wrongly detected as dark lesions (Fig. 7 (c)), or vice versa. Let R and Q represent the detected region and the region covered by structuring elements respectively, and we define $S = Q - R$

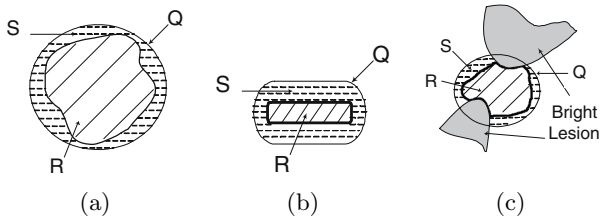


Fig. 7. (a) True lesion. (b) Small blood vessel. (c) Over-detection. The shaded area, R, is the detected region, and the circle area, Q, is the region covered structuring elements.

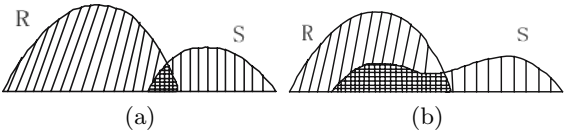


Fig. 8. Intensity densities of R and S for a true lesion (a) and a false detection (b)

as the disparity between Q and R, respectively. For a true lesion, the intensity density (normalized histogram) of S, H_S, has a very small overlap with that of R, H_R, and two densities have more overlap in an over-detected region when both R and S mainly contain background pixels, as shown in Fig. 8. Therefore, we evaluate the overlap percentage of the two densities in H_S and H_R, as

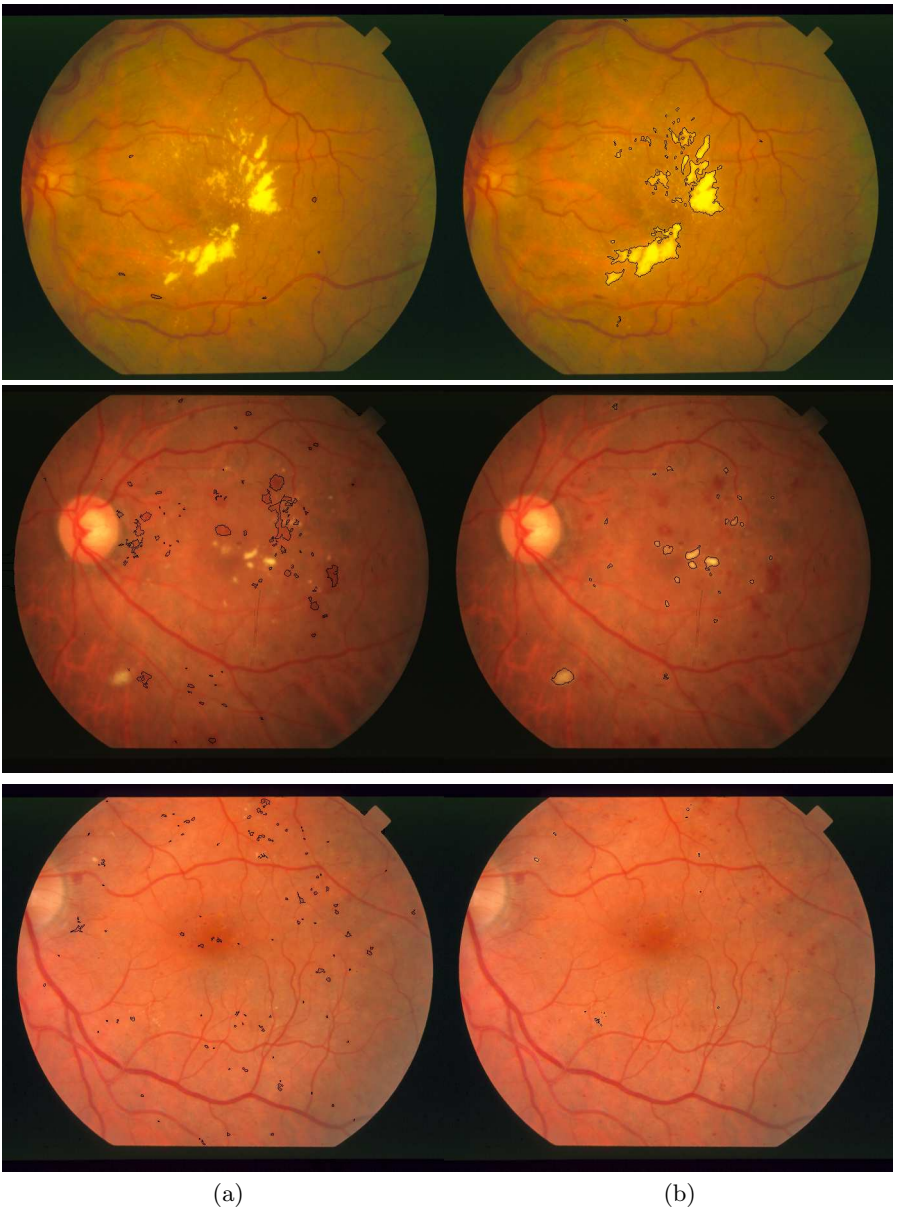
$$\alpha = \sum_i \min\left(H_R(i), H_S(i)\right). \tag{8}$$

When $\alpha < 0.4$, region R is accepted or it is removed from the lesion mask. It is noticed that final detection results are not sensitive to these two thresholds.

4 Simulation Results and Discussions

We have tested the proposed algorithm on 30 retinal images from the Clemson database which includes all five types of spot lesions. These images have been diagnosed as background DR or drusen. To qualitatively evaluate our algorithm, we have manually delineate all dark and bright lesions in 30 retinal images that are used as the benchmark for performance evaluation. We also implemented two recent lesion algorithms for comparative studies.

We first compute sensitivity and predictive value of lesion detection results based on the benchmark. Second, we compare our algorithm's lesion detection performance with two recent algorithms–the hard exudate (bright lesion) detection method (Method-I) by Walter *et. al.* [6] and the microaneurysm (dark lesion) detection method (Method-II) by Spencer *et. al.* [3]. Specifically, we did not implement the last training and classification step in [3] that would enhance



(a) (b)

Fig. 9. (a) Dark lesion detection result superimposed onto the retinal image. (b) Bright lesion detection result superimposed (optical nerve is manually removed). *First row: an image with hard exudates, microaneurysms and hemorrhage. Second row: an image includes hard exudates, soft exudates, microaneurysms and hemorrhage. Third row: an image includes many microaneurysms and little exudates.*

Table 1. Performance comparison with two recent algorithms

Bright Lesion	**Sensitivity**	**Predictive Value**
Before Validation	98.3%	23.9%
After Validation	90.2%	88.6%
Method-I [6]	83.3%	91.2%
Dark Lesion	**Sensitivity**	**Predictive Value**
Before Validation	97.4%	10.3%
After Validation	84.1%	89.2%
Method-II [3]	40.3%	29.5%

the predictive value of Method-II, since we mainly focus on the detection performance in this work. Here we also use the same evaluation method proposed in [6] to calculate the *sensitivity* and *predictive value*. The sensitivity indicates the percentage of our algorithm rightly detected among all human marked lesion pixels. The predictive value indicates the percentage of right detected pixel numbers among all detected pixels. For *bright lesion* calculation, the optical disc was manually removed. The lesion results are presented in Table. 1 and Fig. 9. It is shown that the proposed algorithm can attain very high sensitivity after local entropy/relative thresholding, and the predictive values are improved significantly after lesion validation while keeping high sensitivity as well. Method-I in [6] had slight better predictive value and worked well on small and median size bright lesions. Our database has large bright lesions like shown in Fig. 9. Method-II [3] was mainly designed for detecting MA in fluorescein angiograms, and it cannot deal with hemorrhages, which is usually larger than MA. However, one limitation of the proposed algorithm is that lesion validation may remove some dark lesions that are connected with blood vessels.

5 Conclusions

We have proposed a new retinal lesion detection algorithm that can handle all five types of spot lesions of various intensities, shapes, and sizes. The key contribution is a novel combination between a *linear* parametric edge model and the *non-linear* multiscale morphological processing techniques that leads to a scale map for optimizing structuring elements during lesion enhancement as well as for lesion validation. The proposed algorithm is tested on a set of 30 retinal images with promising performance, and it is also applicable to other medical imaging or industrial inspection applications with similar problems.

Acknowledgments

We thank Dr. Adam Hoover (Clemson University) for providing the retinal image database. This work was supported by an OHRS award (HR03-33) from the Oklahoma Center for the Advancement of Science and Technology (OCAST). This work is also partially supported by the NSF under Grant IIS-0347613.

References

1. ETDRS Research Group: Grading dabetic retinopathy from seteoscopic color fundus photographos – an extension of the modified Airelie House Classification. (ETDRS report number 10). Ophthalmology **98** (1991) 786–806
2. Fransen, S.R., Leonard-Martin, T.C., Feuer, W.J., Hildebrand, P.L.: Clinical evaluation of patients with diabetic retinopathy: Accuracy of the Inoveon diabetic retinopathy-3DT system. The American Academy of Ophthalmology **109** (2002) 595–601
3. Spencer, T., Olson, J.A., McHardy, K.C., Sharp, P.F., Forrester, J.V.: An image-processing strategy for the segmentation and quantification of microaneurysms in fluorescein angiograms of the ocular fundus. Computers and Biomedical Research **29** (1996) 284–302
4. Niemeijer, M., Ginneken, B.V., Staal, J., Suttorp-Schulten, M.S.A., Abramoff, M.D.: Automatic detection of red lesions in digital color fundus photographs. IEEE Trans. Medical Imaging **24** (2005) 584–592
5. Frame, A.J., Undrill, P.E., Cree, M.J., Olson, J.A., McHardy, K.C., Sharp, P.F., Forrester, J.V.: A comparison of computer based classification methods applied to the detection of microaneurysms in ophthalmic fluorescein angiograms. Computers in Biology and Medicine (1998) 225–238
6. Walter, T., Klein, J.C., Massin, P., Erginary, A.: A contribution of image processing to the diagnosis of diabetic retinopathy–detection of exudates in color fundus images of the human retina. IEEE Trans. Medical Imaging **21** (2001) 1236–1243
7. Sbeh, A.B., Cohen, L.D., Mimoum, G., Coscas, G.: A new approach of geodesic reconstruction for drusen segmentation in eye fundus images. IEEE Tran. Medical Imaging **20** (2001) 1321–1333
8. Zhang, X., Chutatape, O.: Top-down and bottom-up strategies in lesion detection of background diabetic retinopathy. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2005) 422–428
9. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Trans. Pattern Analysis and Machine Intelligence **11** (1989) 701–716
10. Vincent, L.: Morphological grayscale reconstruction: Definition, efficient algorithms and applications in image analysis. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (1992) 633–635
11. Mukhopadhyay, S., Chanda, B.: An edge preserving noise smoothing technique using multiscale morphology. Signal Processing (2002) 527–544
12. vanBeck, P.J.L.: Edge-Based Image Representation and Coding. PhD thesis, Delft University of Technology, the Netherlands (1995)
13. Fan, G., Cham, W.K.: Model-based edge reconstruction for low bit-rate wavelet-compressed images. IEEE Trans. Circuits and Systems for Video Technology **10** (2000) 120–132
14. Pal, N.R., Pal, S.K.: Entropic thresholding. Signal Processing **16** (1989) 97–108
15. Chang, C.I., Chen, K., Wang, J., Althouse, M.L.G.: A relative entropy-based approach to image thresholding. Pattern Recognition **27** (1994) 1275–1289

Iterative Closest SIFT Formulation for Robust Feature Matching

Rafael Lemuz-López and Miguel Arias-Estrada

Ciencias Computacionales, Instituto Nacional de Astrofísica Óptica y Electrónica
Apdo. Postal 51 y 216, Puebla, Pue. C.P 72100 México
{rlemuz, ariasm}@inaoep.mx

Abstract. This paper presents a new feture matching algorithm. The proposed algorithm integrates the Scale Invariant Feature Transform (SIFT) local descriptor in the Iterative Closest Point (ICP) scheme. The new algorithm addresses the problem of finding the appropriate match between repetitive patterns that appear in manmade scenes. The matching of two sets of points is computed integrating appearance and distance properties between putative match candidates. To demonstrate the performance of the new algorithm, the new approach is applied on real images. The results show that the proposed algorithm increases the number of correct feature correspondences and at the same time reduces significantly matching errors when compared to the original SIFT and ICP algorithms.

1 Introduction

Matching salient points on images is an essential component in many computer vision tasks, ranging form medical imaging, industrial inspection, compression, surveillance and recently much attention has received on uncalibrated 3D reconstruction. Despite the enormous efforts to build robust matching algorithms, there is no method than can be applied to any situation. Template matching techniques for example are commonly used for tracking multiple regions of the same object [1, 2]. Nevertheless, as current state of the art matching methods can not guaranty mismatches due to image noise, extensions to the original algorithms use robust statistic methods for target position prediction like the Kalman filter [3, 4], other extensions involve outlier rejection methods based on geometry constraints like epipolar geometry [5] and recently prediction of 3D camera motion has been considered to reduce the search space of feature correspondences [3, 6] and decrease mismatch errors.

This work is related to multiple features matching methods. The main contribution of the work is the integration of the discriminative properties encoded in the SIFT descriptor to the ICP algorithm to increase the number of matched features when processing images with repetitive patterns. A matching metric that incorporates appearance and predicted distance to the matching location on successive pairs of images is proposed. In addition the motion context box computed in a uniform sampling area of feature points is introduced to explicitly distinguish between local registration errors to improve the correct match pair assignment.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 502 – 513, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Work

Three important issues in feature matching methods are: the interest point detector algorithm, the local descriptor and the similarity metric that match descriptors of the reference and target images. Feature detector algorithms find image locations where large gradient variations occur in at least two principal directions. Harris [7] and KLT [1] feature detectors select regions where the eigenvalues of the second moment matrix are large. In the last few years feature detectors have been extended to increase its repeatability detection rate under scale, illumination, and small pose variation changes [8, 9, 10]. Once a salient point has been identified a local descriptor is built using photometric information. Recently Mikolajczyk and Schmid [15] have compared various state of the art salient point descriptors, including steerable filters [11], shape context [12], moment invariants [8], complex filters [13], SIFT [14] and cross-correlation. SIFT descriptor shown its superiority with respect to the other feature descriptors when images are subject to local transformations like scale, rotation and partial 3D pose changes. Finally, match assignment is done by computing a similarity metric between descriptors. Commonly used similarity metrics includes sum of square differences, sum of absolute differences, normalized correlation, and Mahalanobis distance metrics.

2.1 Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT) developed by Lowe [14] has 4 major stages:

1. **Scale-space extrema detection:** The first stage searches over scale space using a Difference of Gaussian function to identify potential interest points.
2. **Keypoint localization:** The location and scale of each candidate point is determined and keypoints are selected based on measures of stability.
3. **Orientation assignment:** One or more orientations are assigned to each Keypoint based on local image gradients.
4. **Keypoint descriptor:** A descriptor is generated for each keypoint from local image gradients information at the scale found in stage 2.

The SIFT feature algorithm is based upon finding locations within the scale space of an image which can be reliably extracted. The first stage finds scale-space extrema located in the Difference of Gaussians (DoG) function, which can be computed from the difference of two nearby scaled images separated by a multiplicative factor *k*. DoG scale space is sampled by blurring an image with successively larger Gaussian filters and subtracting each blurred image from the adjacent (more blurred) image

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1)$$

where $L(x,y,\sigma)$ is the scale space of an image, built by convolving the image $I(x,y)$ with the Gaussian kernel $G(x,y,\sigma)$. Points in the *DoG* function which are local extrema in their own scale and one scale above and below are extracted as keypoints.

For a rotation invariant descriptor, the local image gradients are used to determine the main orientation for each region. For each image sample $L(x,y)$, the gradient magnitude $m(x,y)$ and orientation $\theta(x, y)$ is computed using pixel differences:

$$m(x,y)^2 = (L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y+1))^2 \tag{2}$$

$$\theta(x,y) = \tan^{-1}\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)} \tag{3}$$

The resulting feature descriptor are *128* elements vectors with a total support window of *16 x16* scaled pixels. For a more detailed explanation of the SIFT see [14].

2.2 Iterative Closest Point

The ICP algorithm is another approach for solving the correspondence problem, although originally introduced to register 3D data sets by Besl and McKay [16] and Chen and Medioni [17] it is also used with 2D data sets to register images mainly on medical applications. The ICP algorithm iteratively performs two operations until convergence: the data matching and the transformation estimation to align the data sets.

The ICP algorithm takes two data sets as input representing salient points of a reference image I_1 and a target image I_2. Let P and Q be two input data sets containing N_p and N_q feature points respectively, that is, $P = \{p_i\}_{i=1}^{N_p}$ and $Q = \{q_i\}_{i=1}^{N_q}$. The goal is to compute the parameters of the transformation matrix T that best aligns the transformed points $P_k=T(P)$, to the point set Q. For 2D Euclidean transformation the parameters of T are the Rotation angle R and the Translation vector $t=(t_x,t_y)$.

A summary of the ICP algorithm is as follow:

1 Initialize: $k=0$; $P_k=P$;
2 Match closest points: For each point in P_k, finds its closest point in Q. A list $C = \{c_i\}_{i=1}^{N_p}$ is obtained where $C \subset Q$, and c_i es the closest point to p_i.

$$\arg\min_{c_i} \| c_i - Rp_i - t \| \tag{4}$$

3 Compute the transformation parameters of T with a mean square estimator, where the objective function to be minimized is:

$$f(R,t) = \frac{1}{N_p}\sum_{i=1}^{N_p} \| c_j - Rp_i - t \|^2 \tag{5}$$

4 Apply the transformation: $P_{k+1}=RP+t$
5 If stopping condition is met: Terminate
else $k=k+1$ and repeat steps 2-5

We use standard stopping conditions: the maximum allowable number of iteration K_{max}, the change in mean square registration error falls bellow a preset threshold $\tau>0$

and when the change in transformation parameters has a small variation between successive iterations.

Notice that the transformation parameters are computed from the original data set P whereas the closest points in the matching stage are computed using the transformed points P_k. Closed form solutions for computing the 2D Euclidean transformation exist based on SVD [17] and unit quaternions [18].

Two important problems in the ICP algorithm are:

It needs a good initial computation of feature correspondences especially in cases where image deformation is present due to viewpoint changes. As a result, given two data sets, the ICP algorithm converges to different local minima depending on the initial correspondences assigned between the data sets. To deal with this problem some methods have been proposed using extra information like colour [19, 20] in which the colour information is incorporated to the shape information in the correspondence computation or using combination of multiple attributes [21,22], which are combined with the Euclidean distance in searching for the closest point during the matching process.

Another drawback of the ICP algorithm is that it is sensitive to the presence of false match correspondences or "outliers". Zhang [23] introduced a variant of the ICP, which rejects corresponding points if the distance between the pair is greater than a dynamically adapted threshold called D_{max}. The threshold is updated in each iteration using the mean μ and the standard deviation η of the computed distances between the corresponding points as follow:

```
if  μ< D        // The registration is very good
    Dmax =μ+3η
elseif μ< 3D    // The registration is still good
    Dmax =μ+2η
elseif  μ< 6D   // The registration is not too bad
    Dmax =μ+η
else            // The registration is bad
    Dmax =ξ
```

Where D is a constant defined by the user that indicates the expected mean distance of the corresponding points when the registration is good and ξ is a maximum tolerance distance value when the registration is bad.

Following these ideas we propose to integrate the SIFT descriptor to the ICP algorithm and local static measures to improve the robustness of feature matching stage. Recent reviews and comparative studies about the ICP variant methods are available in [24, 25] focused on alternatives that improve the convergence speed.

3 Iterative Closest SIFT

In this section we show how to integrate SIFT and ICP methods to overcome one of the intrinsic limitation of each algorithm. Adding spatial information in the matching stage of SIFT and providing a good initial set of correspondences to initialize ICP

algorithm. Our motivation is to increase the number of feature correspondences on images containing repetitive patterns and to eliminate false match pairs.

3.1 Finding Initial Matching Pairs

Since ICP is an iterative descent algorithm, it requires a good initial estimate in order to converge to a local minimum. In the original ICP algorithm it is assumed that the data sets P and Q are partially aligned or a rough estimation of the transformation parameters is known. If this assumption is satisfied then, during the first matching process an important number of feature correspondences can be found and the algorithm can converge to a local minimum.

In the proposed algorithm a first estimate of the correspondences is computed based on the SIFT local descriptors. This allows to find a good amount of match pairs even when the initial partially alignment constrain is no satisfied.

The SIFT descriptors are 128 dimensional vector that describes the appearance of the local region. The search strategy is to search through the entire list of SIFT features descriptors $S_p = \{s_{pi}\}_{i=1}^{N_p}$ for ones that are similar in the set $S_q = \{s_{qi}\}_{i=1}^{N_q}$. Then, given two SIFT feature descriptors s_{pi} and s_{qj}, one from each data sets P and Q, the similarity function is defined as the Euclidean distance metric:

$$d_S = \| s_{pi} - s_{qi} \| = \sqrt{\sum_{k=1}^{128} (s_{pi,k} - s_{qi,k})^2} \tag{6}$$

Then, following the approach of [14] we use the distance ratio to the second best match to discard outliers and take this matching metric as the final similarity measure in the initialization stage. Given d_{s1} and d_{s2} the first and second best matches distances for a feature descriptor in the set S_p, a match is accepted if:

$$d_{S1} < 0.8 d_{S2} \tag{7}$$

Notice that the distance ratio has a low value when the best match performs significantly better that the second best match, while a high value of the distance ratio is obtained when the feature point has at least one strong competitor in terms of the descriptor appearance.

3.2 Matching SIFT Features Adding a Weighted Distance Factor

The match closest point equation (4) of the original ICP algorithm is adapted to consider both appearance similarity and geometrical proximity properties between feature points by:

$$\arg\min_{c_i} \| c_i - Rp_i - t \| / (\delta + \beta) + \omega \| s_{pi} - s_{qi} \| \tag{8}$$

The coefficient ω is a factor that control the weight of SIFT based appearance in matching stage. The parameter δ manages the degree of interaction between features according to the distance a small δ value limits matching candidates to a reduced neighborhood while a bigger value allows matching more distant features. β is a small

value to avoid division by zero. Then, following [23] where the maximum tolerable distance to accept or reject matches can be computed automatically. We set the parameter δ varying accordingly to the mean μ and standard deviation η computed in each iteration of the IC-SIFT algorithm.

$$\delta = \mu + 2\eta \tag{9}$$

We have observe that a good tradeoff to weight appearance and distance information is obtained setting the coefficient ω=0.5.

3.3 Differencing Registration Error

Traditional ICP algorithms do not distinguish about different registration errors for different point matches. This fact cause that many feature points can be rejected because their corresponding registration error is larger than the global registration error computed from all the pairing matches in previous iterations. This problem is especially evident working with real images since features of closer objects will have larger displacements than features corresponding to far away objects that will be related with small motion displacements.

We propose to add a local motion descriptor to overcome this limitation of the original method. The suggested motion context uses a similar data structure introduced in [29] to speed up the searching of correspondences but it differs in the information it holds. The motion context is an *MxN* rectangular grid of boxes that encloses the feature points from the reference image. Each motion context box B_{ij} will store the characteristic displacement vector v_{Bij}, the mean registration error μ_{Bij} and the standard deviation η_{Bij} of the N_{Bij} inlaying features in the corresponding box.

$$v_{B_{ij}} = \frac{\sum_{i=1}^{N_{B_{ij}}} c_i - (Rp_i + t)}{\left\| \sum_{i=1}^{N_{B_{ij}}} c_i - (Rp_i + t) \right\|} \tag{10}$$

The stored values will be used to find matches with a clear distinction of local registration error estimation. Using local registration errors will be proven under experimental test to be especially useful in boundary regions in the next section. Let (X_{min}, Y_{min}) the minimum and (X_{max}, Y_{max}) the maximum coordinates of the feature points in *P*. As we divide uniformly the bounding rectangular region that contains all the feature points, the number of Horizontal *H* and Vertical *V* subdivisions is defined by:

$$H = \frac{(X_{\max} - X_{\min})}{\Delta x}, V = \frac{(Y_{\max} - Y_{\min})}{\Delta y} \tag{11}$$

where Δx and Δy are the vertical and horizontal size of subdivision. Then, given the coordinates (x,y) of feature point p_i its corresponding B_{ij} indices are obtained as follow:

$$i = \left\lfloor \frac{x - X_{\min}}{\Delta x} \right\rfloor, j = \left\lfloor \frac{y - Y_{\min}}{\Delta y} \right\rfloor, \ if \ \ x \le X_{\max}, y \le Y_{\max}$$
$$i = H, j = V, \ if \ \ x > X_{\max}, y > Y_{\max} \tag{12}$$

To remove the effect of outliers in the computation of local mean and standard deviation, only the subset of features whose registration error is lower than three times the global mean registration error are considered to obtain the values of each motion context box. During the initialization phase of the IC-SIFT algorithm the motion context is estimated using only the feature correspondences extracted by appearance information from the SIFT descriptor. Then, at the matching stage for each feature point p_i its closest motion context box B_{ij} is estimated and if the corresponding motion context box is in the boundary of the image the transformed p_{ki} is set according to.

$$p_{k_i} = Rp_i + t + 0.5v_{B_{ij}} \tag{13}$$

Finally, for each feature point p_i its closest motion context box B_{ij} is estimated and the local estimated μ_{Bij} and η_{Bij} are used to accept or reject matching pairs based on local thresholds dynamically updated defined as follows.

```
if   μ_Bij < D          // The registration is very good
     Dmax_Bij = μ_Bij + 3η_Bij
elseif μ_Bij < 3D       // The registration is still good
     Dmax_Bij = μ_Bij + 2η_Bij
elseif  μ_Bij < 6D      // The registration is not too bad
     Dmax_Bij = μ_Bij + η_Bij
else                    // The registration is bad
     Dmax_Bij = ξ
```

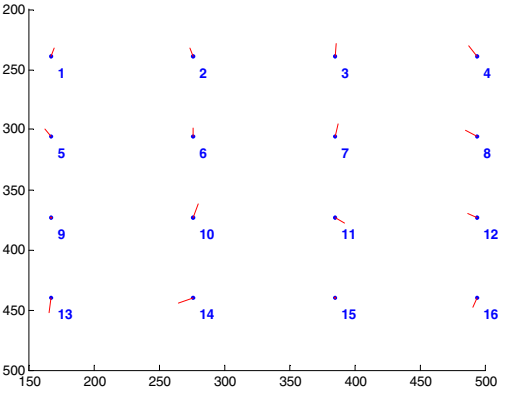


Fig. 1. Example of the 4x4 motion context box describing characteristic local feature motion

4 Experimental Results

This section shows the comparative results of the proposed IC-SIFT Algorithm against the original SIFT and ICP algorithms. In all the experiments we evaluate the number of correct match features on image pairs containing repetitive patterns. The performance of each matching algorithm is measured considering the number of correct matching features and the number of false positives matches i. e. incorrect matches between non corresponding feature points. In addition the RMS registration error is measured. We begin by using the RANSAC-fundamental matrix computation procedure [17]. Then, the estimated epipolar lines are used to classify a match as incorrect when the distance from the corresponding epipolar line is more than one pixel. But, since this approach has problems identifying false positives we have carried on a second manual classification process to obtain a more representative measurement between correct and incorrect matches. The verification process was done by two different users on each individual feature point. Figure 2 shows the real data sets employed to evaluate the proposed algorithm.

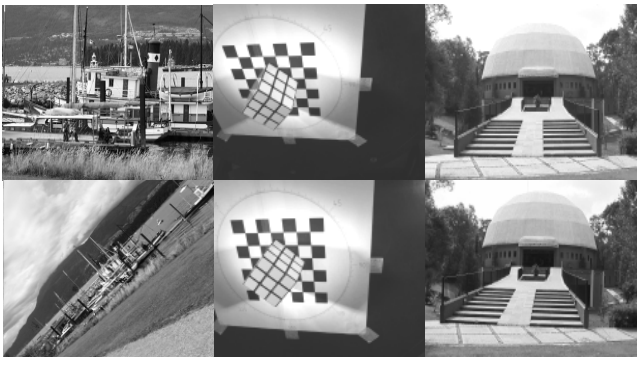


Fig. 2. Original test image pairs. Images are subject to rotation, scale and viewpoint change transformations.

In the first experiment we use the first and sixth image of the boat image sequence from [9]. Notice that this image pair has a large rotation and scale change. In this experiment the IC-SIFT algorithm clearly outperform the original algorithms since the transformation parameters can be reliable obtained.

The second experiment shows the performance of the new method in an image pair of the cube stereo pair with camera motion from top to bottom. Observe that there are many areas in the checkerboard pattern and in the cube with similar patterns. In addition, there are illumination changes on the left and right of the images since two spot lights were used to illuminate the scene.

200 SIFT key point descriptors were initially detected on the first image and looked for its matching pairs in 500 key points detected on the right image.

Table 2 summarizes the results obtained by using the proposed IC-SIFT, the original SIFT and ICP algorithms in the cube image pair.

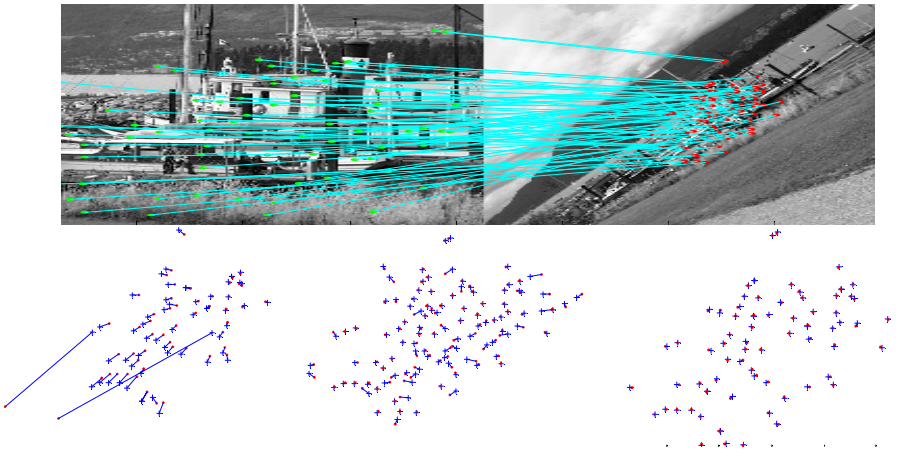


Fig. 3. Top feature correspondences found by the proposed IC-SIFT algorithm. Bottom match pairs found using SIFT, ICP and IC-SIFT algorithms.

Table 1. Matching results for boat image pair

Matching Method	Initial Features	Correct Matches	False Positive Matches	RMS Error
IC-SIFT	200	85	0	1.87
ICP	200	73	23	2.53
SIFT	200	55	6	3.92

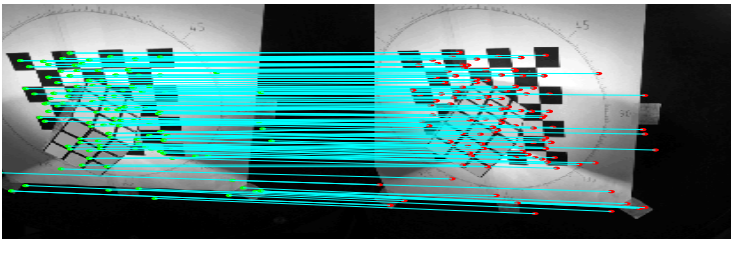


Fig. 4. Matches found by proposed IC-SIFT Algorithm in the cube image pair

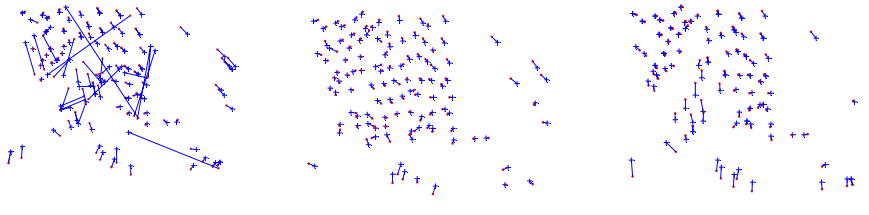


Fig. 5. Match pairs found using SIFT, ICP and IC-SIFT algorithms

Table 2. Matching results for cube image pair

Matching Method	Initial Features	Correct Matches	False Positive Matches	RMS Error
IC-SIFT	200	110	3	1.91245
ICP	200	96	26	2.5177
SIFT	200	92	15	4.0836



Fig. 6. Matches found by the IC-SIFT Algorithm in the library image pair

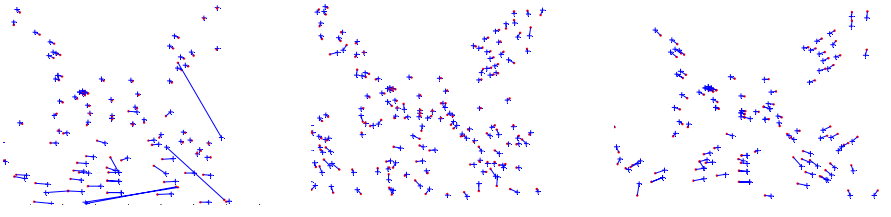


Fig. 7. Matches found using SIFT, ICP and IC-SIFT algorithms in the library image pair

Table 4. Matching results for library image pair

Matching Method	Initial Features	Correct Matches	False Positive Matches	RMS Error
IC-SIFT	200	118	7	2.8417
ICP	200	99	43	3.2397
SIFT	200	82	20	4.8937

Notice that since this image pair has repetitive patterns using the SIFT matching strategy, yields poor results and the correspondences are found in relatively large regions where ambiguities between features can be saved to some extend. On the other hand the IC-SIFT outperforms to the original algorithms with an improvement on both the number of correct matched points and with a significant reduction of false positives matches.

In the last experiment, we investigate the robustness of the new matching algorithm on a natural image pair (Proprietary library), see figure 5.

Notice that there is a view point change corresponding to 30 degrees. Furthermore the camera follows a translational motion from left to right. 200 interest points where initially detected in the image on the left and matched to 500 putative key points on the right image.

We observe from the results that the IC-SIFT algorithm always gives the lowest registration errors. This is due to the fact that it always gives the lowest false positive mach errors and the highest correct match scores.

5 Conclusions

A new feature matching algorithms has been proposed. The new algorithm integrates spatial proximity information to the SIFT descriptor and appearance similarity to the ICP algorithm to overcome problems of identifying correct matches on images with repetitive patterns and wide baseline separations between image pairs. The proposed IC-SIFT algorithm is applied to feature matching with excellent results.

The performance of the novel ICP-SIFT matching algorithm when compared with the original SIFT shows that the proposed algorithm finds 30% more matches and reduce significantly the number of false positive matches when processing real image pairs containing repetitive patterns. Future work will be done to eliminate even more outliers by integrating color information to the novel method.

References

1. Tomasi, C., Kanade, T.: Detection and tracking of point features. CMU Technical Report (1991) CMU-CS-91-132
2. Hager, G., Belhumeur, P.: Real-time tracking of image regions with changes in geometry and illumination. In proceedings of IEEE Conference on Computer vision and Pattern Recognition (1996) 403-410
3. Davison A.: Real-Time Simultaneous Localisation and Mapping with a Single Camera. ICCV (2003) 1403-1410
4. Soatto S., Frezza R., Perona P.: Motion estimation via dynamic vision. IEEE Transactions on Automatic Control, **41** (1996) 393-413
5. Zhang, Z., Deriche, R., Faugeras, O. Luong, Q.: A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Artificial Intelligence Journal **78** (1995) 87-119
6. Nistér, D.: Preemptive RANSAC for Live Structure and Motion Estimation. ICCV (2003) 199-206
7. Harris, C., Stephens, M.: A combined corner and edge detector. In M. M. Matthews, editor, Proc. Of the 4th ALVEY vision conference (1988) 147-151
8. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets. In Proceedings of the 7th European Conference on Computer Vision (2002) 414-431
9. Mikolajczyk, K., Schmid, C.: Scale and Affine Invariant Interest Point Detectors. International Journal of Computer Vision, **10** (2004) 63-86
10. Kadir, T., Zisserman, A., Brady, M.: An affine invariant salient region detector. ECCV, **1** (2004) 228-241

11. Freeman, W., Adelson, E.: The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, **13** (1991) 891-906
12. Belongie, S., Malik, J., P. J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, **24** (2002) 509-522
13. Van Gool, L., Moons, T., D. Ungureanu.: Affine/photometric invariants for planar intensity patterns. In Proceedings of the 4th European Conference on Computer Vision, Cambridge (1996) 642-651
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91-110
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. PAMI **27** (2005) 1615-1630
16. Besl, P., McKay, N.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 239–256
17. Chen, Y., Medioni, G.: Object Modeling by Registration of Multiple Range Images. Image and Vision Computing, **10** (1992) 145–155
18. Horn B.: Closed-form solution of absolute orientation using unit quaternions. Optical Society of America A, **4** (1987) 629–642
19. Godin, G., Rioux, M., Baribeau, R.: Three-dimensional registration using range and intensity information. Videometrics III, Proc. SPIE, **2350** (1994) 279-290
20. Johnson, A., Kang, S.: Registration and integration of textured 3-D data. In Conference on Recent Advances in 3-D Digitial Imaging and Modeling (1997) 234–241
21. Schuts, T., Jost, T. Hugli, H.: Multi-featured matching algorithm for free-form 3D surface registration. In IEEE International Conference on Pattern Recognition, (1998) 982–984
22. Godin, G., Laurendeau, D., Bergevin, R.: A method for the registration of attributed range images. IEEE International Conference on 3D Imaging and Modeling, (2001) 179-186
23. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vision **13** (1994) 119-152
24. Rusinkiewicz, S., Levoy, M.: Efficient Variants of the ICP Algorithm. Proceedings of IEEE 3DIM, (2001) 145-152
25. Gruen, A., Akca, D.: Least squares 3D surface and curve matching. ISPRS Journal of Photogrammetry & Remote Sensing, **59** (2005) 151-174
26. Chetverikov, D.: Fast neighborhood search in planar point sets. Pattern Recognition Letters **12**, (1991) 409-412

Invertible Polygonalization of 3D Planar Digital Curves and Application to Volume Data Reconstruction

Martine Dexet[1], David Cœurjolly[2], and Eric Andres[1]

[1] Laboratoire SIC - E.A. 4103, Université de Poitiers
Bât. SP2MI, bvd Marie et Pierre Curie, BP 30179
86962 Futuroscope Chasseneuil Cedex, France
{dexet, andres}@sic.univ-poitiers.fr
[2] Laboratoire LIRIS - CNRS UMR 5205, Universit Claude Bernard Lyon 1
43 Bd du 11 novembre 1918
Villeurbanne, France
dcoeurjo@liris.cnrs.fr

Abstract. In this paper, we describe a new algorithm to compute in linear time a 3D planar polygonal curve from a planar digital curve, that is a curve which belongs to a digital plane. Based on this algorithm, we propose a new method for converting the boundary of digital volumetric objects into polygonal meshes which aims at providing a topologically consistent and invertible reconstruction, i.e. the digitization of the obtained object is equal to the original digital data. Indeed, we do not want any information to be added or lost. In order to limit the number of generated polygonal faces, our approach is based on the use of digital geometry tools which allow the reconstruction of large pieces of planes.

1 Introduction

3D digital volume data can be very important which is a problem for fluent interactive visualization. To solve this problem, Marching Cubes (MC) based algorihms [1,2,3] and many variations of these methods [4,5] have been proposed these last twenty years. In some applications, we prefer to process directly on the digital model instead of the polyhedral representation because either the algorithms would not be efficient (*e.g.* in Constructive Solid Geometry) or the tools are not defined in the polyhedral model (*e.g.* mathematical morphology tools). To take advantages of both models, transformation schemes are of interest. In this paper we investigate mappings between the class of digital sets and the class of polyhedra. This mapping is an onto and not one-to-one mapping since several polyhedra may be digitized into a given digital object. Nonetheless, the existence of such a mapping allows us model changes without loss of information. In this context, given a digital object, the problem is to compute a polyhedral representation such that its digitization coincides with the input object. This scheme is called the *invertible reconstruction* of a digital set. In digital geometry, the invertible reconstruction is a classical problem [6,7,8]. Usually,

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 514–523, 2006.
© Springer-Verlag Berlin Heidelberg 2006

proposed reconstruction methods first consist of decomposing the digital object into several plane pieces using a *digital plane recognition algorithm* in order to obtain large plane segments. Then, based on this decomposition, a polyhedral representation is computed. This is done by performing for each recognized digital face an invertible polygonalization to obtain a Euclidean face. However, no proposed method provides an invertible and topologically consistent polyhedral reconstruction, that is without holes.

In this paper, we are interested in reconstructing 6-connected digital volumes. We first present a new efficient 3D planar digital curve polygonalization algorithm which computes in linear time a planar Euclidean polygonal face such that its digitization is equal to the original digital face (according to the Standard digitization model). Then, based on this algorithm, we propose a new volume data reconstruction method which converts a 3D digital object into a topologically correct polyhedron with respect to the invertibility property.

In Section 2, we first introduce the Standard analytical digitization model and focus on 3D Standard plane and line definitions. Then, we detail the Standard plane recognition method used in this work. In Section 3, we describe the 3D invertible digital curve polygonalization algorithm on which is based the digital volume data reconstruction method presented in Section 4. We conclude in Section 5.

2 Digital Geometry: Recalls and Tools

2.1 The Standard Analytical Model

The Standard model [9] is a digitization model, defined in any dimension and topologically consistent, which allows the 6-connected digitization of any linear object of $\mathbb{R}^3$. 3D Standard planes and lines (see Figures 1a and 1b) are analytically defined as follows:

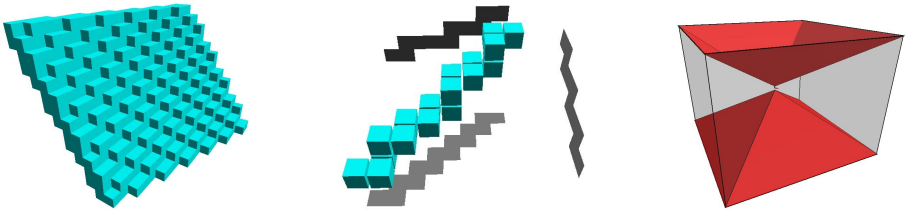
Definition 1 (Standard plane [9]). *The Standard plane with parameters* $(a, b, c, d) \in \mathbb{R}^4$ *is the set of points* $(x, y, z) \in \mathbb{Z}^3$ *verifying*

$$-\frac{|a| + |b| + |c|}{2} \leq ax + by + cz + d < \frac{|a| + |b| + |c|}{2}$$

where $a > 0$, *or* $a = 0$ *and* $b > 0$, *or* $a = b = 0$ *and* $c > 0$.

A *3D Standard line* is a 6-connected digital line whose three orthotropic projections are 2D Standard lines (see [9] for details).

The Standard digitization of a 3D Euclidean primitive (plane or line) also consists of all *voxels* (unit-size cubes) centered on *digital points* (points with integer-valued coordinates) which are cut by the primitive, except when this primitive crosses a voxel vertex. In this case, several voxels adjacent to this vertex do not belong to the Standard digitization. This is due to the fact that one inequality in Definition 1 is strict.



(a) (b) (c)

Fig. 1. (a) A Standard plane, (b) A 3D Standard line and its three 2D projections, (c) Dual of the voxel $(0, 0, 0)$

2.2 Standard Plane Recognition

The aim is to decide if a given voxel set belongs or not to a Standard plane. Various digital plane recognition algorithms have been proposed in the last decades [10,11], but only some of them provide the set of all Euclidean planes, or *solution set*, whose digitization contains the given voxels [12,13]. In this work, we use a Standard plane recognition algorithm [13] based on the computation of a polyhedron called the *generalized preimage* of the voxels. This preimage is computed in a parameter space and each point which lies inside this polyhedron corresponds to a Euclidean plane which crosses the given voxels. More precisely, relations between the 3D Euclidean and parameter spaces are defined as follows: To a point with coordinates $(a, b, c) \in \mathbb{R}^3$ in the parameter space $\mathcal{P}^3 = (O\alpha\beta\gamma)$ is associated the Euclidean plane with equation $z = ax + by + c$. In the same way, to a Euclidean point with coordinates $(x, y, z) \in \mathbb{R}^3$ is associated the plane with equation $\gamma = -x\alpha - y\beta + z$ in the parameter space. Let p be a point in the Euclidean or parameter space. In the following, we will denote by $Dual(p)$ the associated plane in the other space. Finally, let l be a 3D straight line in the Euclidean or parameter space, and let p_1 and p_2 be two points lying on l. Then, l maps to another 3D straight line $L = Dual(p_1) \cap Dual(p_2)$. Let V be a voxel. Then, $Dual(V) = \bigcup_{p \in V} Dual(p)$, also called the *dual* of V, can be geometrically defined as follows:

Proposition 1 (Dual of a voxel [13]). *Let V be a voxel with center $c = (x_c, y_c, z_c)$. Let V_+ (resp. V_-) be the set of vertices $v_i = (x_i, y_i, z_i)$ of V verifying $z_i \geq z_c$ (resp. $z_i \leq z_c$). Then,*

$$Dual(V) = \Big(\bigcup_{v \in V_-} Dual(v)^+ \Big) \cap \Big(\bigcup_{v \in V_+} Dual(v)^- \Big)$$

where

$$O^+ \ (resp.\ O^-) = \bigcup_{p=(x_p,y_p,z_p) \in O} \left\{ p' = (x_{p'}, y_{p'}, z_{p'}) \,\middle|\, \begin{array}{c} x_p = x_{p'} \\ y_p = y_{p'} \\ z_p \leq (resp.\ \geq)\ z_{p'} \end{array} \right\}$$

From Proposition 1, it derives that the dual of a voxel V (see Figure 1c) is the open parameter space area delineated by eight particular planes, each plane being the dual of a vertex of V. Moreover, the dual of each point in $Dual(V)$ is a plane which cuts V.

Definition 2 (Generalized preimage [13]). *Let $V = (V_1, \ldots, V_n)$ be a set of n voxels, and let $Dual(V_i)$ be the dual of V_i in $\mathcal{P}^3$, $1 \leq i \leq n$. The generalized preimage $\mathbb{G}_P$ of V is defined by:*

$$\mathbb{G}_P(V) = \bigcap_{i=1}^{n} Dual(V_i)$$

The recognition algorithm only consists of computing the generalized preimage of the voxels. If the preimage is empty, the voxel set does not belong to a Standard plane, otherwise it does. The complexity of this algorithm is $O(n^2)$ in the worst case, where n denotes the number of voxels.

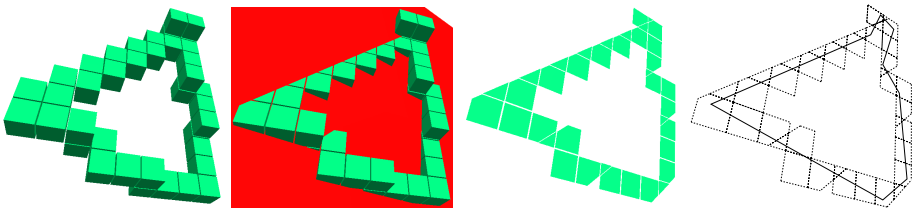
3 3D Planar Standard Curve Polygonalization

In this section, we tackle the following problem: Given a planar Standard curve C (see Figure 2a), that is, a 6-connected digital curve which belongs to a Standard plane, and given a Euclidean plane P which cuts all voxels of C, how to compute a planar polygonal curve which lies on P and such that its Standard digitization is equal to C? In [8], the authors propose a way of polygonalizing such a curve. The complexity of the proposed algorithm is $\mathcal{O}(n\ log\ n)$, where n denotes the number of voxels of the curve. In this paper, we describe a new polygonalization algorithm which solves this problem in linear time.

This algorithm works basically as follows. First of all, we suppose that the plane P has already been determined (see Figure 2b), for instance by choosing a plane in the solution set provided by the computation of the generalized preimage of C (see Section 2.2). Our polygonalization algorithm is then composed of two steps: First, the intersection between P and C is computed in order to obtain a planar curve composed of faces with various shapes (see Figure 2c). We call such a curve an *irregular curve*. Then, a planar polygonal curve which lies on the plane and which crosses all faces of the irregular curve is computed (see Figure 2d).

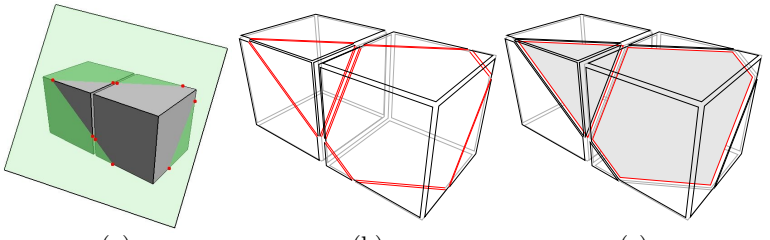
3.1 Intersection Operation Between a 3D Digital Curve and a Plane

In order to compute the intersection between C and P, we apply the following method. First, all vertices of C are sorted according to their position with respect to P. Three different vertex classes denoted $\oplus$, $\ominus$ and $\circleddash$ are thus obtained depending on the vertex positions. From these classes, all edges of C which are cut by P are determined (one endpoint belongs to $\oplus$ and the other one belongs to $\ominus$), and intersections between these edges and P are computed in order to create new vertices (see Figure 3a). Then, if two vertices have been created on a



(a) (b) (c) (d)

Fig. 2. Example of curve polygonalisation: (a) A digital curve, (b) A plane which cuts the voxels, (c) Intersection between the plane and the curve, (d) One possible result



(a) (b) (c)

Fig. 3. Illustration of the intersection algorithm: (a) Intersection operation between the plane and the edges, (b) Creation of new edges, (c) Creation of new faces

same voxel face, a new edge is created between these two vertices (see Figure 3b). In the same way, if new edges have been created on a same voxel, a new convex face is created from these edges inside the voxel (see Figure 3c). Finally, links are established between the new created faces according to the connections existing between the voxels of C. Since this algorithm treats only once every cell of the curve, its complexity is in $\mathcal{O}(k)$, where k denotes the number of cells (voxel edges, voxel faces and voxels) of C. However, since k and the number of voxels of C are linearly dependent, the complexity of this algorithm is $\mathcal{O}(n)$, where n denotes the number of voxels.

3.2 Irregular Curve Polygonalization

The aim is to compute from an irregular curve C' (see Figure 2c) a planar polygonal curve which crosses all faces of C' (see Figure 2d). In order to solve this problem, we extend the algorithm proposed in [14] which allows the polygonalization of curves composed of cells belonging to irregular isothetic grids.

The principle of the polygonalization algorithm is the following:

First, a face F is chosen in C', and a Euclidean point P_0 (see Figure 4a) is chosen inside F (for instance its barycenter). Note that if C' is an open curve (i.e. there exists two faces which are connected to only one other face), we choose one of the two extremity faces. If C' is closed, F can be chosen arbitrarily, or according to a heuristic. Moreover, a polygonalization direction has to be chosen.

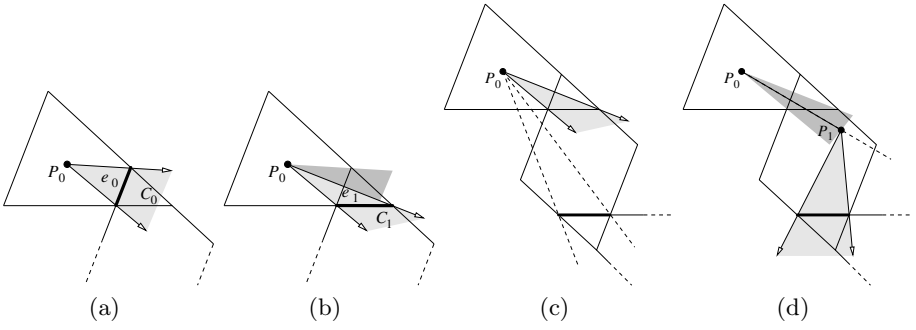


Fig. 4. Illustration of the irregular curve polygonalization algorithm: (a) First computed cone, (b) Visibility cone reduction according to the next face, (c) Example of empty visibility cone, (d) Choice of a new point and next cone computation

Then, the cone C_0 with center P_0 and defined by the two endpoints of the edge e_0 incident to F and the following face is computed. This *visibility cone* corresponds to all Euclidean lines passing through P_0 and crossing e_0.

A new cone C_1 (see Figure 4b) with center P_0 and defined by the two endpoints of the edge e_1 incident to the two next faces is computed. The visibility cone is then reduced according to C_1, that is the new visibility cone is equal to the intersection between C_0 and C_1. This process is iterated until the visibility cone becomes empty or until the last face of C' is reached.

If the visibility cone becomes empty, this means that there exists no straight line passing through P_0 and crossing all considered edges of C' (see Figure 4c). In this case, a line is chosen in the previous visibility cone and a new point P_1 lying on this line is chosen inside the last face reached by the visibility cone. The complete process is then iterated from P_1 until the last face of C' is reached (see Figure 4d). Note that if C' is closed, the last computed visibility cone has to reach the first chosen point P_0 in order to obtain a closed polygonal line.

4 3D Digital Analytical Reconstruction

Several attempts have been made in the digital geometry community to convert 3D digital objects into polygonal meshes [7,8]. However proposed algorithms either suffer from topological inconsistency or do not provide invertible results. In this section, we propose a 3D digital analytical reconstruction algorithm which solves these problems. Indeed, it provides an invertible and topologically correct polyhedral object from volume data.

4.1 Principle

Our algorithm works with the boundary B of a 6-connected digital volume (see Figures 5a and 5b). We assume that each voxel vertex which lies on B is a digital

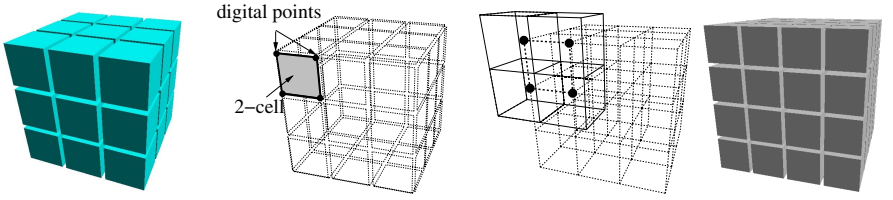


Fig. 5. (a) Input data, (b) Boundary of (a), (c) Digital volume equivalent to (a)

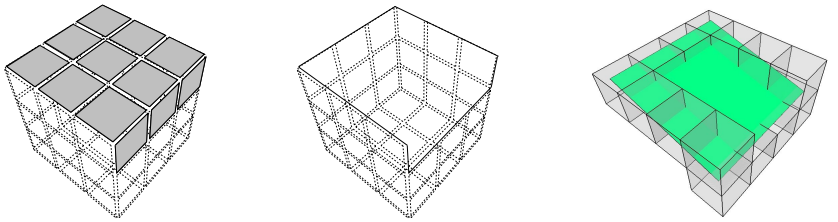
point. Four digital points incident to a same voxel face form a so-called *2-cell*. Our goal is to reconstruct a polygonal mesh so that its Standard digitization is equal to the digital points of B. In other words, if we consider each digital point of B as the center of a new voxel, the reconstructed object has to cross each voxel of this new 6-connected hollow volume (see Figure 5c).

The reconstruction algorithm works basically as follows: First of all, a 2-cell S is chosen in B and a first digital Standard plane piece is recognized starting from S using the 3D recognition algorithm described in Section 2.2 (see Figure 6a). When the recognition process ends, a Euclidean plane P is chosen in the computed solution set and the recognized plane border C, which is a 3D 6-connected digital curve, is extracted (see Figure 6b). The face polygonalization process (see Section 3) is then applied on C in order to obtain the contour of a Euclidean face which lies on P. By construction, its Standard digitization is equal to the recognized digital plane piece (see Figure 6c). A new unprocessed 2-cell is then chosen and the whole processes start over. These two processes are iterated until all 2-cells are treated.

However, the plane recognition process is constrained. Indeed, during this step, the next 2-cell c to be recognized might be adjacent to already reconstructed digital faces. The voxels of c are then crossed by one or more faces and thus belong to the Standard digitization of an already reconstructed face vertex or edge. In order to perform a topologically correct reconstruction, our goal is to obtain a polygonal face which passes through these vertices and edges. To ensure this property, *vertex and edge constraints* have to be taken into account during the recognition process. This means that a 2-cell is recognized if and only if its constraints do not lead to an empty solution set.

4.2 Constrained Plane Recognition

The first digital face is recognized without constraints. For those that follow, the recognition process can be constrained. Hence, if the next 2-cell to be recognized is constrained, the solution set provided by the recognition algorithm has to be updated according to each voxel constraint. The update of the solution set due to a vertex (resp. edge) constraint is done by doing the intersection between the generalized preimage and the dual plane (resp. line) of the constraint (see Section 2.2). If the result is empty, the recognition process stops. If not, a plane is chosen in the intersection.



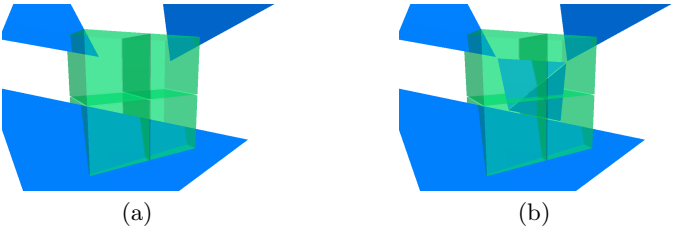
(a) (b) (c)

Fig. 6. Illustration of the reconstruction method: (a) Recognized 2-cells, (b) Digital curve border extraction (bold line), (c) Example of face polygonalization

4.3 Face Polygonalization

When the recognition process ends, the solution set is either empty or not. If not, a plane P has been chosen in the set and the border of the recognized 2-cells extracted. The corresponding 6-connected curve is then reconstructed according to P in order to obtain a facet which is on P. The extracted border can be adjacent to already reconstructed faces. Hence, we may not have to reconstruct the entire curve but only curve pieces, each curve being constrained at its extremities by a vertex or an edge constraint. Let C be such a curve. If C is closed and unconstrained, we need before applying the polygonalization process to fix a Euclidean point lying on the chosen plane, the digitization of which is equal to one of the curve digital points. Instead, if an extremity of C is constrained by an edge, a Euclidean point is fixed on the segment resulting of the intersection between the voxel extremity and the edge.

If the solution set is empty, this means that the first 2-cell to be recognized was too constrained so that no plane which passes through all the constraints exists. More precisely, this case corresponds to a four constrained surfel, that is each voxel of the surfel is constrained by a vertex or an edge. The reconstruction of such a 2-cell S is done as follows: If the voxel is constrained by an edge, a point is chosen on the intersection between the edge and the voxel. Then, two facets are added between the four vertex constraints (see Figure 7).

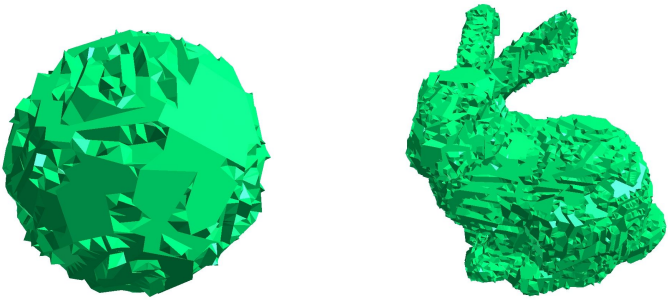


(a) (b)

Fig. 7. Specific treatment: (a) A four constrained surfel. (b) Facets addition

Table 1. Results: Comparison between the classical MC method and the Digital Analytical Reconstruction method

Input data	Dimensions	# digital points	MC		DAR	
			# vertices	# faces	# vertices	# faces
Sphere	33^3	4824	5212	10420	3013	3472
Bunny	$80 \times 79 \times 62$	22056	23105	46204	14037	16456
Dragon	$206 \times 146 \times 94$	110938	114552	229096	73649	87182



(a) (b)

Fig. 8. Digital analytical reconstruction examples: (a) A sphere, (b) A bunny

4.4 Results

Some results provided by our methods are shown in Table 1 and Figure 8. We can see that there is a consequent reduction in number of vertices and faces compared to the MC method while keeping the invertibility of the reconstruction property. Contrary to the MC method, our reconstruction algorithm is not deterministic. Various Heuristics can be applied to improve or adapt the method depending on the expected final result. For instance, depending on the starting 2-cell choice and recognition order, there can be either several big polygons linked by many tiny ones or smaller polygons with a more uniform size. Heuristics can also take into account geometrical or topological features such as sharp edges or corners, small cavities, tunnels or holes. The reconstruction algorithm is presented here in a very direct form without any ameliorations as it is not the goal of the paper.

5 Conclusion

In this paper, we have proposed a new 3D planar digital curve polygonalization algorithm which computes in linear time a planar Euclidean polygonal curve such that its Standard digitization is equal to the original digital curve. Based on this algorithm, we have proposed a way of performing an invertible and topologically consistent volumetric data surface reconstruction. A logicial challenging future work would be to perform exhaustive comparisons between this algorithm and

MC based algorithms. Moreover, various heuristics have to be taken into account in order to improve the obtained reconstructed objects.

References

1. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: SIGGRAPH. Volume 21 of Computer Graphics (ACM)., Anaheim, USA (1987) 163–169
2. Lewiner, T., Lopes, H., Vieira, A.W., Tavares, G.: Efficient implementation of marching cubes'cases with topological guarantees. JGT **8** (2003) 1–15
3. Nielson, G.M.: Dual marching cubes. In: IEEE Visualization, IEEE Computer Society (2004) 489–496
4. Montani, C., Scateni, R., Scopigno, R.: Discretized marching cubes. In: Proceedings of the Conference on Visualization, Los Alamitos, CA, USA (1994) 281–287
5. Lachaud, J.O., Montanvert, A.: Continuous analogs of digital boundaries: A topological approach to iso-surfaces. Graph. Mod. and Im. Proc. **62** (2000) 129–164
6. Franon, J., Papier, L.: Polyhedrization of the boundary of a voxel object. In: DGCI. Volume 1568 of LNCS. (1999) 425–434
7. Klette, R., Sun, H.J.: Digital planar segment based polyhedrization for surface area estimation. In: IWVF Proceedings. Volume 2059 of LNCS. (2001) 356–366
8. Sivignon, I., Dupont, F., Chassery, J.M.: Reversible polygonalization of a 3D planar discrete curve: Application on discrete surfaces. In: DGCI. Volume 3429 of LNCS. (2005) 347–358
9. Andres, E.: Discrete linear objects in dimension n: the Standard model. Graphical Models **65** (2003) 92–111
10. Buzer, L.: A linear incremental algorithm for Naive and Standard digital lines and planes recognition. Graphical models **65** (2003) 61–76
11. Gerard, Y., Debled-Rennesson, I., Zimmermann, P.: An elementary digital plane recognition algorithm. Discrete Applied Mathematics **151** (2005) 169–183
12. Vittone, J., Chassery, J.M.: Recognition of digital Naive planes and polyhedrization. In: DGCI. Volume 1953. (2000) 296–307
13. Dexet, M., Andres, E.: A generalized preimage for the Standard and Supercover digital hyperplane recognition. In: DGCI, Szeged, Hungary (2006) To appear.
14. Cœurjolly, D., Zerarga, L.: Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids. Computer and Graphics **30** (2006) 46–53

Lateral and Depth Calibration of PMD-Distance Sensors

Marvin Lindner and Andreas Kolb

Computer Graphics Group, University of Siegen, Germany

Abstract. A growing number of modern applications such as position determination, object recognition and collision prevention depend on accurate scene analysis. The estimation of an object's distance relative to an observers position by image analysis or laser scan techniques is thereby still the most time-consuming and expensive part.

A lower-priced and much faster alternative is the distance measurement with modulated, coherent infrared light based on the Photo Mixing Detector (PMD) technique. As this approach is a rather new and unexplored method, proper calibration techniques have not been widely investigated yet. This paper describes an accurate distance calibration approach for PMD-based distance sensoring.

1 Introduction

The determination of an object's distance relative to a sensor is a common field of research in Computer Vision. During the last centuries, techniques have been developed whose basic principles are still used for modern systems in a wide scope, such as laser triangulation or stereo vision.

Nevertheless, there is no low-priced off-the-shelf system available, which provides full-range, high resolution distance information in real-time even for static scenes. Laser scanning techniques, which merely sample a scene row by row with a single laser device are rather time-consuming and impracticable for dynamic scenes. Stereo vision camera systems on the other hand suffer from inaccuracy caused by mismatching correspondences in homogeneous areas.

The ZCam camera add-on provided by 3DV Systems [1] allows the determination of full range distance profiles of dynamic indoor scenes in real time. However, it uses highly complex shutter mechanism which makes it cost-intensive and unhandy.

A rather new and promising approach developed during the last years estimates the distance by time-of-flight measurements for modulated, incoherent light even for outdoor scenes based on the new Photo Mixing Detector (PMD) technology. The observed scene is illuminated by infrared light which is reflected by visible objects and gathered in an array of solid-state image sensors based on CMOS technology [2,3,4]. Unlike other system, the PMD-system is a very compact device which fulfills the above stated features desired for real-time distance acquisition.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 524–533, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The contribution of this paper is a calibration model for the very complex and lightly explored distance mapping process of the PMD. This includes lateral and distance calibration by deviation analysis.

A short overview about the PMD's functionality and known PMD-calibration techniques is given in Sec. 2. Our calibration model in general is described in Sec. 3. A detailed description of the lateral and the distance calibration is given afterwards in the Sec. 4 and 5. Finally, the results are discussed in Sec. 6 which leads to a short conclusion of the presented work.

2 Related Work

2.1 Photo Mixing Detector (PMD)

By sampling and correlating the incoming optical signal with the reference signal of the modulated, incoherent illumination directly on a pixel, the PMD is able to determine the signal's phase shift and thus the distance information by a time-of-flight approach [2,3,4].

Given a reference signal $g(t)$ and the optical signal $s(t)$ incident in a PMD pixel, the pixel samples the correlation function $c(\tau)$ for a given internal phase shift τ:

$$c(\tau) = s \otimes g = \lim_{T\to\infty} \int_{-T/2}^{T/2} s(t) \cdot g(t+\tau)\, dt.$$

For a sinusoidal signal, some trigonometric calculus yields

$$s(t) = \cos(\omega t), \qquad g(t) = k + a\cos(\omega t + \phi), \qquad c(\tau) = \tfrac{a}{2}\cos(\omega\tau + \phi) \tag{1}$$

where ω is the modulation frequency, a is the amplitude of the incident optical signal and ϕ is the phase offset relating to the object distance. The modulation frequency defines the distance unambiguousness. The demodulation of the correlation function is done using several samples of $c(\tau)$ obtained by four sequential PMD raw images $A_i = c(i \cdot \frac{\pi}{2})$:

$$\phi = \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right), \qquad a = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2}. \tag{2}$$

The manufacturing of a PMD chip is based on standard CMOS-processes which allows a very economic production of a device which is, due to an automatic suppression of background light, suitable for indoor as well as outdoor scenes.

Current devices provide a resolution of 48×64 or 160×120 px at 20 Hz, which is of high-resolution in the context of depth sensing but still of low-resolution in terms of image processing. A common modulation frequency is 20 MHz, resulting in an unambiguous distance range of 7.5 m. An approach to overcome the limited resolution of the PMD camera is its combination with a 2D-sensor [5].

2.2 Calibration

The calibration of common 2D cameras is a highly investigated subject mainly using point correspondences, vanishing lines or information from camera motion. Several approaches using planar calibration targets has been proposed, e.g. using homographies and dual-space geometries [6,7,8].

A first distance calibration approach in the context of a PMD was done by Kuhnert and Stommel [9]. Here, the phase-difference for the average of 5x5 center pixel has been related to the true distance inside a certain range by a simple line fitting for few distance samples only. After a correction of a given input image, a min/max depth map has been calculated by examine the neighbors of each pixel, which leads to a confidence interval for the true distance information.

3 PMD Calibration

Before we start to describe the individual calibration steps in more detail, a short overview about the camera calibration and the calibration model designed for the PMD camera is given.

In order to perform a full calibration of a given PMD camera, the calibration process is decomposed into two separate calibration steps:

1. a lateral calibration already known from classical 2D sensors and
2. a calibration of the distance measuring process.

One main reason for the distance distortion in the PMD camera is a systematic error due to the demodulation of the correlation function. Beside that, there are several factors like the IR-reflectivity and the orientation of the object that may lead to insufficient incident light to a PMD pixel and thus to an incorrect distance measurement. Additionally, the demodulation distance assumes distance homogeneity inside the solid angles corresponding to a PMD pixel. In reality different distances inside a solid angle appear and lead to superimposed reflected light reducing its amplitude and distorting a phase shift.

The only indicator about the accuracy of an individual distance endowed by the PMD camera is given by the amplitude a of the correlation function, which incorporates saturation, distance homogeneity, object reflectivity and object orientation (see Eq. 1).

The presented depth calibration model does not take full account of all known effects stated above. In the first instance, we rather want to concentrate on measurement deviations caused by manufacturing inaccuracies and the systematic error using a heuristic model.

4 Lateral Calibration

The lateral calibration is the classical Computer Vision approach for determining camera specific information such as radial lens distortion, focal length f and real image center (c_x, c_y). Given these parameters, the perspective projection $\mathbf{x}$

of a point $\mathbf{X}$ in space is defined by $\lambda\mathbf{x} = K\Pi_0\mathbf{X}$ [10,11,12]. K is the matrix of the camera depending intrinsic parameters and Π_0 represents the standard perspective projection.

$$K = K_s K_f = \begin{bmatrix} f/s_x =: f_x & 0 & c_x \\ 0 & f/s_y =: f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \; \Pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The lens distortion is commonly modeled by a polynomial of 4th degree [13]:

$$x_d = c_x + (x - c_x)(1 + a_1 r^2 + a_2 r^4 + 2t_1\tilde{y} + t_2(r^2/\tilde{x} + 2\tilde{x}))$$
$$y_d = c_y + (y - c_y)(1 + a_1 r^2 + a_2 r^4 + 2t_2\tilde{x} + t_1(r^2/\tilde{y} + 2\tilde{y}))$$

where $r^2 = \tilde{x}^2 + \tilde{y}^2$ and $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, 1)$ are the projected coordinates $\lambda\tilde{\mathbf{x}} = \Pi_0\mathbf{X}$.

The main goal of the PMD lateral calibration described in this section, is to investigate common calibration methods for the calibration of the low resolution PMD camera of at most 160×120 pixel. We therefore decided to analyze the behavior of an existing calibration module included in Intel's OpenCV library [14], which is based on techniques described by Bouguet [8]. This approach uses at least three different views of a planer checkerboard to estimate the intrinsic parameters as well as the radial distortion coefficients.

The lateral calibration has been tested by interactively passing new PMD images of a 4x7 checkerboard to the calibration module until the intrinsic parameters remained to be stable inside a certain range (see Fig 1). In order to suppress noise, five successive PMD grayscale images are averaged. Further improvement of the detection rate and the visual feedback is achieved by applying an initial histogram normalization to enhance the low contrast of the PMD grayscale image.

Before passing the pre-processed PMD grayscale to the pattern recognition stage, a scaling of the whole image is necessary. This is due to the fact, that the applied search window for the corner detection of the size 5×5 px covers too many pixels in the low resolution PMD image. Thus, the PMD-image is scaled to 480×360 px using bi-linear resampling. Higher order resampling, e.g. bi-cubic, leads to slightly more robust corner detection.

Overall, both the intrinsic parameters as well as the distortion coefficients were detected sufficiently robust in most of the experiments (see Tab. 1).

5 Distance Calibration

The following sections describe the distance calibration model itself along with the data analysis which is the basis for the calibration model.

As basis for the distance calibration, a series of reference measurements for known distances have been made. The PMD distance images were taken in the range of 0.75–7.5 m with a spacing of 10 cm. The 68 references distance images have been captured with a 48×64 PMD-sensor acquiring data of a planar, semi-reflective panel.

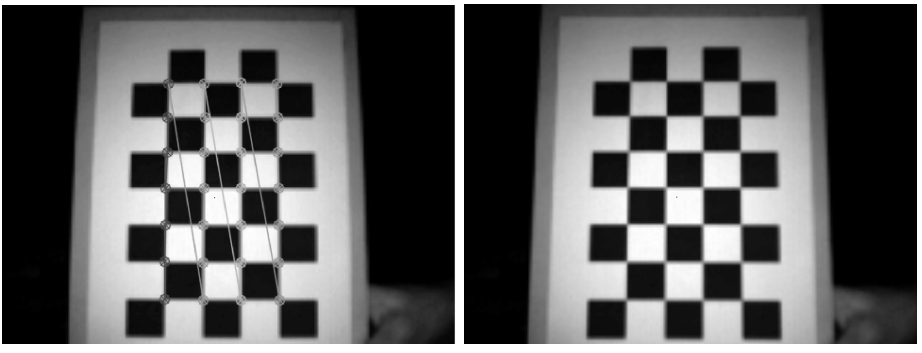


Fig. 1. Example of an PMD image before (left) and after a lateral calibration (right). Reference points has been marked by a line stripe.

Figure 2 shows the distance distortion, i.e. the difference between the measured and expected distance in each pixel for the depth images. The result exhibits an oscillating error with large variance in the close-up range. The latter can be explained by an constant exposure-time, which has been set up for for medium distances of 4-7 m.

To compensate the distance deviation the distance calibration is done in two distinguished steps (see Fig. 3):

1. a global distance adjustment for the entire image and
2. a local per pixel (pre-) adaption to obtain better results for the global adjustment and to compensate remaining deviation.

5.1 Global Distance Calibration

The main idea is to use a function of higher complexity for the global adjustment in combination with a simpler adjustment for the per-pixel calibration. The per-pixel calibration uses linear adjustment, thus being storage-efficient concerning the overall number of calibration parameters.

Given the periodicity if the distance error, one attempt would be to use sinusoidal base-function for the correction. Our approach uses uniform, cubic B-splines instead. B-splines exhibit a better local control and, even more important for online-calibration tasks, the evaluation of B-spline always requires a constant number of operations, in our case the evaluation of a cubic polynomial.

A segmentation of the distance images is applied in order to remove outlayers due to oversaturation. This has been done by examining the pixel's amplitude of the optical signal, which is a direct indicator for the reliability of the measured value (see Sec. 2.1).

In order to identify the optimal number of control points, an iterative least-square fitting for B-spline curves $b_{\text{glob}}(d) = \sum_{l=0}^{m} c_l \cdot B_l^3(d)$ is performed. In this

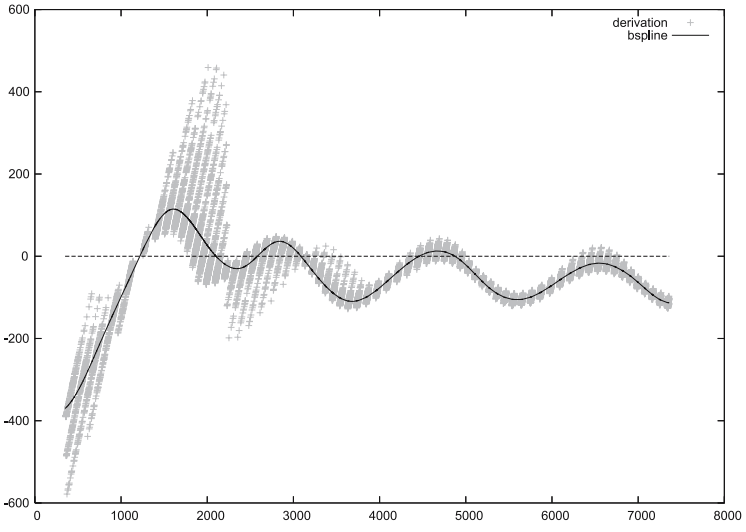


Fig. 2. Deviation between the measured and the expected distance as function of the measured distance for image depth information between 0.75-7.5 m and the fitted B-spline

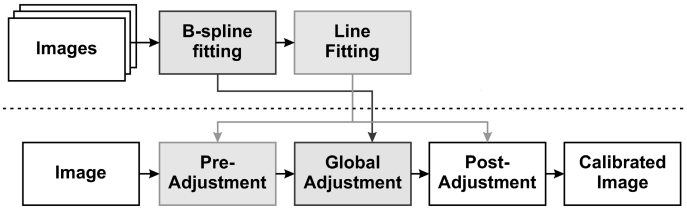


Fig. 3. The distance calibration process

approach, the number of control points m is successively increased, until the approximation error $\|A\mathbf{c} - \mathbf{b}\|^2$ with

$$A = \left[B_i^3(d_J)\right]_{\substack{i=0,\ldots,m \\ J=(x,y,k)}}, \quad \mathbf{c} = [c_i]_{i=0,\ldots,m}, \quad \mathbf{b} = [p_J]_{J=(x,y,k)}$$

is below a given threshold or m exceeds a maximum. Here $d_J = d(x, y, k)$ is the measured distance at pixel (x, y) of distance-image k and p_J is the difference to the real distance to the plane in image k. The resulting system of linear equations can be solved using a pseudo inverse approach: $\mathbf{c} = (A^T A)^{-1} A^T \mathbf{b}$.

The global adjustment of the distance $d(x, y, k)$ due to the determined fitted B-spline curve b_{glob} is simpl:

$$d_{glob}(x, y, k) = d(x, y, k) - b_{\text{glob}}(d(x, y, k)). \tag{3}$$

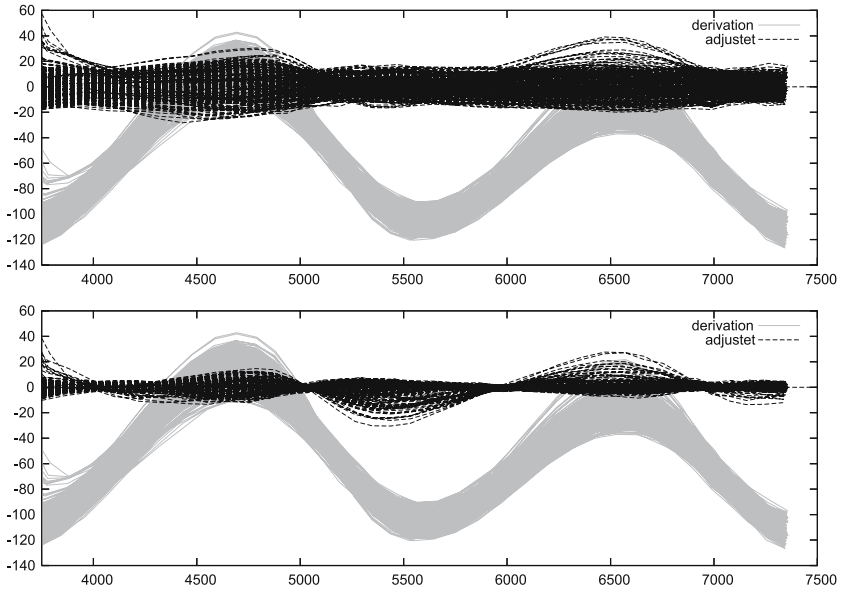


Fig. 4. Original (light gray) and adjusted deviation (dashed, dark gray) between the measured and the expected distance for measured distances between 3.75 - 7.5 meter. Top: global correction only, bottom: additional pre-adjustment.

Applying this global correction, we obtain an improved distance accuracy. Figure 4, top, shows the error functions for some pixels in the uncorrected and the globally corrected situation.

5.2 Per-pixel Distance Calibration

An even better result can be achieved by additionally taking individual pixel inaccuracies into account. Up to this point we have discussed the main idea of the global distance calibration process, but have not really considered which distance to adjusted:

1. the polar distance $d^p(x, y, k)$ given by the PMD camera, which reflects the natural time-of-flight point of view for central projection or
2. the according Cartesian coordinates $d^c(x, y, k)$ after the conversion with $\alpha_x = \arctan((x - c_x)/f_x)$ and $\beta_{xy} = \arctan((c_y - y)/(((x - c_x) \cdot \frac{f_y}{f_x})^2 + f_y^2)^{0.5})$:

$$d^c(x, y, k) = \cos \alpha_x \cdot \cos \beta_{xy} \cdot d^p(x, y, k)$$

The calibration process for both coordinates systems differ in the order of coordinate transformation and distance adjustment. First, the case of Cartesian coordinates is discussed. Here, the process of B-spline fitting can be simplified by using mean values w.r.t. the known plane distances. The average mean $d_{avg}(k)$

of all per-pixel distances $d(x, y, k)$ in distance image k can be used for B-spline fitting to reduce the number of sample points.

As the B-spline fitting relates to an average deviation between the distance $d(x, y, k)$ and the expected distance $d_{ref}(x, y, k)$,

$$b(d(x, y, k)) \approx d_{avg}(k) - d_{ref}(x, y, k), \tag{4}$$

an evaluation of the B-spline at a per-pixel distance leads to errors in the global adjustment. In order to reduce this error, a per-pixel pre-adjustment is applied, so that a pixel's distance closer matches the average distance d_{avg} over all distance images k. This is done by fitting a line $l_{x,y}$ for each pixel (x, y) minimizing

$$\sum_k \|d(x, y, k) - d_{avg}(d(x, y, k))\|^2.$$

In the case of Cartesian coordinates d^c_{avg} is simply given by

$$d^c_{avg}(k) = \frac{1}{n} \sum_{(x,y)} d^c(x, y, k)$$

where n is the number of pixels (x, y) taken into account.

Unfortunately, such an image averaging is not possible in the case of polar coordinates. Here, we use the dependence between $d^p_{avg}(k)$ and the B-spline $b_{\text{glob}}(d)$ as stated in Eq. 4

$$d^p_{avg}(k) \approx b_{glob}(d(x, y, k)) + d^p_{ref}(x, y, k)$$

Thus, the overall distance calibration is given by

$$d_c(x, y, k) = b_{glob}(d(x, y, k) - l_{x,y}(d(x, y, k)))$$

To eliminate remaining distance deviations a second line fitting analog to the pre-adjustment can be performed. This time for the differences between the global corrected distances and the reference plane.

The comparison between the calibration of both coordinate systems, together with the remaining results, are given in the Sec. 6.

6 Results

The calibration process of the PMD camera was decomposed into two separate calibration steps for a lateral and distance calibration.

In the case of the lateral calibration we have been able to show that the estimation of the PMD intrinsic parameters is possible and stable (see Tab. 1). To achieve this, the PMD images have to be pre-processed. Here a normalization and a up-scaling with bi-linear resampling suffices to stabilize and speed-up the pattern recognition process. Occasionally, outliers have to be compensated by passing another pattern view to the calibration module.

Table 1. Sample results of lateral calibration of a 120×160 PMD camera. The manufacturers' value for the PMD camera are: focal length 12 mm, 0.04 mm pixel dimension.

	Calibration Parameters		
Focal Length (f_x, f_y)	12.39, 12.36	12.29, 12.28	12.30, 12.29
Image Center (c_x, c_y)	77.49, 66.20	76.27, 68.75	78.93, 63.34
Radial Distortion (r_1, r_2)	-0.4869, 1.1313	-0.4700, 1.2842	-0.4824, 1.7011
Tangential Distortion (t_1, t_2)	0.0009, 0.0012	0.0000, 0.0048	0.0022, 0.0014

Table 2. Average variance of distance deviation in *mm* for all segmented pixel in the depth range of 3.75-7.5 m

Distance	**uncalibrated**	**global adjust.**	**+ pre-adjust.**	**+ post-adjust.**
Polar	64.1421	7.50838	3.20462	2.87305
Cartesian	64.1182	7.63023	3.11621	2.90705

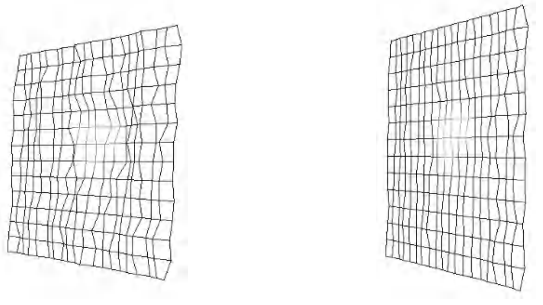


Fig. 5. Original distance profile of a plane wall (left) and the same image after an distance adjustment (right)

For the distance calibration we have been able to show, that polar and Cartesian adjustment lead to comparable results for a 48×64 PMD camera (see Tab. 2). In both cases we reached a per-pixel precision of 10 mm or better, whereas the averaged variance for each pixel has been about 3 mm (see Fig. 4).

The only difference between both approaches is the B-spline fitting, as already mentioned in Sec. 5. By using Cartesian coordinates, it is possible to calculate an average mean distance for each image in order to reduce the number of sample points without changing the precision, leading to a slight speed-up.

Due to the per-pixel line fitting, it turns out that a post-adjustment can be neglected as it leads to no further improvements. Therefore a global adjustment with pre-processing alone should be sufficient for an accurate distance calibration. A final example for a 3D distance adjustment is shown in Fig. 5.

7 Conclusions

This paper proposes a first approach to calibration of PMD distance sensors. In this context we are able to show a method for estimating the camera's intrinsic

parameter. Furthermore we have been able to calibrate the distance data of a 64×48 PMD camera with high accuracy. This approach directly carries over to PMD-sensors with higher resolution.

Further studies will consider how to combine high resolution 2D images and low resolution PMD images to get more detailed information for calibration and distance refinement, i.e. homogeneous surfaces or object contours. Furthermore, some work should be invested in a physically based illumination model for the PMD camera, in order to improve the characterization of the distance correction.

Acknowledgments

This work is partly supported by the German Research Foundation (DFG), KO-2960/5-1. Furthermore we would like to thank PMDTechnologies Germany.

References

1. 3DV Systems: (2006) http://www.3dvsystems.com.
2. Xu, Z., Schwarte, R., Heinol, H., Buxbaum, B., Ringbeck, T.: Smart pixel – photonic mixer device (PMD). In: Proc. Int. Conf. on Mechatron. & Machine Vision. (1998) 259–264
3. Lange, R.: 3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology. PhD thesis, University of Siegen (2000)
4. Kraft, H., Frey, J., Moeller, T., Albrecht, M., Grothof, M., Schink, B., Hess, H., Buxbaum, B.: 3D-camera of high 3D-frame rate, depth-resolution and background light elimination based on improved PMD (photonic mixer device)-technologies. In: OPTO. (2004)
5. Prasad, T., Hartmann, K., Weihs, W., Ghobadi, S., Sluiter, A.: First steps in enhancing 3D vision technique using 2D/3D sensors. In Chum, O.Franc, V., ed.: 11th Computer Vision Winter Workshop 2006. (2006) 82–86
6. Tsai, R.Y.: An efficient and accurate camera calibration technique for 3d-machine vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1986) 364–374
7. Zhang, Z.: A flexible new technique for camera calibration. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. (2000) 133–1334
8. Bouguet, J.Y.: Visual methods for three-dimensional modeling. PhD thesis (1999)
9. Kuhnert, K.D., Stommel, M.: Fusion of stereo-camera and pmd-camera data for real-time suited precise 3D environment reconstruction. In: Proc. Int. Conf. on Intelligent Robots and Systems. (2006) submitted for publication.
10. Ma, Y., Stoatto, S., Kosecka, J., Sastry, S.: An invitation to 3D vision. Springer (2004)
11. Faugeras, O.: Three-dimensional Computer Vision. The MIT Press (1993)
12. Forsyth, Ponce: Computer Vision - A Modern Approach. Alan Apt (2003)
13. Brown, D.: Decentering distortion of lenses. **32** (1966)
14. OpenCV: (2006) http://www.intel.com/technology/computing/opencv/index.htm.
15. Luan, X.: Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology. PhD thesis, Department of Electrical Engineering and Computer Science (2001)
16. Schwarte, R., Zhang, Z., Buxbaum, B.: Neue 3D-Bildsensoren für das Technische 3D-Sehen. VDE Kongress "'Ambient Intelligence"' (2004)

Autonomous Vehicle Video Aided Navigation – Coupling INS and Video Approaches

Chris Baker[1], Chris Debrunner[1], Sean Gooding[2], William Hoff[2], and William Severson[1]

[1] PercepTek, Inc. Littleton, CO
[2] Colorado School of Mines, Golden CO

Abstract. As autonomous vehicle systems become more prevalent, their navigation capabilities become increasingly critical. Currently most systems rely on a combined GPS/INS solution for vehicle pose computation, while some systems use a video-based approach. One problem with a GPS/INS approach is the possible loss of GPS data, especially in urban environments. Using only INS in this case causes significant drift in the computed pose. The video-based approach is not always reliable due to its heavy dependence on image texture. Our approach to autonomous vehicle navigation exploits the best of both of these by coupling an outlier-robust video-based solution with INS when GPS is unavailable. This allows accurate computation of the system's current pose in these situations. In this paper we describe our system design and provide an analysis of its performance, using simulated data with a range of different noise levels.

1 Introduction

Global Positioning System (GPS) outages are a serious problem to any autonomous vehicle system. Many autonomous navigation systems today rely on the consistent availability of GPS to compute an ongoing estimate of vehicle pose (position and orientation). GPS dropouts are not uncommon, typically occurring next to buildings or occluding land formations. GPS can also easily be jammed or spoofed. To accommodate GPS dropouts, a robust autonomous navigation system must additionally exploit an inertial navigation system (INS) approach using accelerometers and gyroscopes, and it must be tolerant of varying amounts of INS drift. Angular drift varies with INS quality and associated expense, and lower cost solutions typically rely on corrections from GPS to provide the best performance. Alternatively, video data can be used to compute vehicle pose in the absence of GPS. A video-based approach used alone is problematic, however, due its reliance on image texture. The presence and amount of image texture is context dependent and cannot be characterized in general.

The video-based pose approach that we use performs well on data containing many outliers due to our use of a statistical sampling analysis of the data. While this approach is common in video-based 2-frame pose estimation, it has not been used before as a method for outlier filtering for a Kalman based pose tracking system. Video-based solutions are not computed for every frame of data and hence our system handles the multi-rate approach providing an estimated pose at the highest sensor rate.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 534 – 543, 2006.
© Springer-Verlag Berlin Heidelberg 2006

This problem is in general an example of an approach to the simultaneous localization and mapping (SLAM) problem, as it is sometimes referred to in the computer vision community [1-3] which simultaneously estimates both the camera motion as well as the 3D locations (structure) of the tracked features – typically using only imagery. The addition of inertial measurements for improved accuracy is not new in computer vision [4-18]. Most of these approaches merge inertial measurements in a Kalman-based filter. Many approaches use a minimization technique on the image plane error to estimate the pose of the camera and combine this with the inertial measurements while some incorporate the features directly in the Kalman. Our approach differs from these in how we identify features to insert into the Kalman filter. By first performing outlier-robust pose estimation, we guarantee that the features added in the Kalman are inliers and initialize the 3D location of the feature points within the Kalman.

The autonomous navigation system described in this paper has three general areas of consideration: 1) the use of video sensor data, 2) computation of video-based vehicle pose, and 3) integration of video-based and INS-based vehicle pose. The presentation of the system in this paper is organized in terms of these three areas. Section 2.1 presents the simulated data generation capability that is used to provide data as input to the system. Video-based pose computation is described in Section 2.2. Integration of video-based pose and INS-based pose using a Kalman filter is described in Section 2.3. Section 3 presents analysis and results comparing the integrated system with an INS-only and a video-only approach and demonstrates that the combined system performs better than either system independently.

2 System Description

We have developed a Video Aided Navigation (VAN) system that provides an ongoing estimate of vehicle pose. In the general system, GPS, INS, and video data sources are used. Dropouts of GPS data or periods of non-useful video data are expected and accommodated. In this paper, we focus specifically on the point where GPS data becomes unavailable or unreliable. We are currently focusing our attention on an airborne platform scenario, specifically an unmanned air vehicle (UAV).

The system diagram is shown in Fig. 1. An imaging sensor is mounted on a UAV platform directed toward the ground. Features are extracted from the initial image in the video stream (as in [19]) and tracked as the images are collected using a correlation-based feature tracker. These feature tracks are used to compute the vehicle pose using either an essential matrix or a homography matrix estimation. The set of feature tracks will tend to include outlier features arising for various reasons (see Section 2.1.1). The image-based pose estimation module removes these outliers by performing a RANSAC-based sampling analysis which uses a robust error measure for outlier rejection (see Section 2.2 as well as [3, 20]). The Kalman filter maintains an estimate of the vehicle pose by combining the vision-based measurements with the INS measurements from the platform after initializing with the GPS corrected INS system pose.

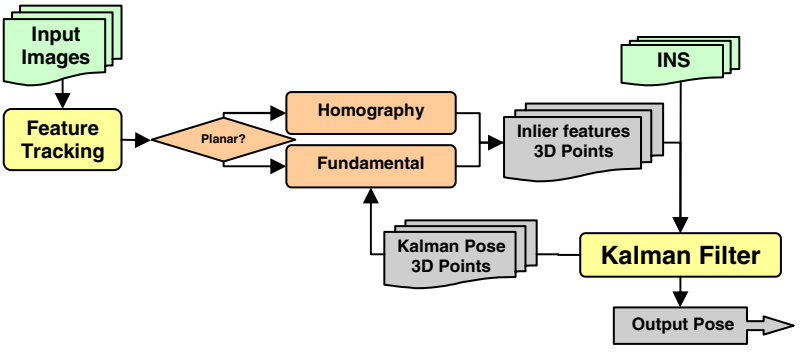


Fig. 1. Video Aided Navigation System Diagram

2.1 Full System Model

In order to systematically vary data inputs and noise levels, we have created a system model simulation package. The entire simulation consists of the following pieces.

- Ground plane of points
- Aerial vehicle (UAV)
- Camera system with fully defined intrinsic parameters
- Camera attached to the UAV platform
- One or more movable objects on the ground plane
- Ground truth transformations between all system objects
- INS system connected to the UAV
- GPS system connected to the UAV
- Complete noise models for each sensor in the simulation

The ground plane consists of a grid of 3D representing features extracted and tracked in a video stream. The terrain is modeled using sine wave functions. The 3D position of a given point on the ground plane dictates where it projects into an oriented camera of a given field-of-view. A trajectory specifies a sequence of platform poses, and each video frame consists of the projection of the collection of ground features that are visible. In order to accurately simulate what we would expect to see in real data, noise models have been developed for each simulated sensor.

2.1.1 Simulated Feature Tracks

The objective in feature extraction and tracking is to first find a large number of unique (trackable) features and track them over as many subsequent frames as possible. The vision community has spent a great deal of effort over its history developing robust feature extraction techniques. One of the more successful of these approaches makes use of the eigenvalues of the covariance of the local image gradient computed over a neighborhood at each image point [19, 21, 22]. Feature points are selected at locations where the smaller eigenvalue is greater than a threshold. This selects points that have significant intensity variation along both image dimensions.

Our data generation system creates track information that simulates a feature detector and tracker's output by adding noise to the true projected image positions of the 3D ground points. For a given simulation the ground features are partitioned into a set

of inliers and outliers. Inliers are feature points with small position errors that are tracked well and that correspond to a real fixed 3D feature. Outlier points have gross errors and model features not associated with a fixed 3D ground point (such as a moving object) or features that are not reliably tracked. Separate image plane noise models are applied to inliers and outliers. The following list outlines various causes of outliers typically seen in visual imagery, each of which we can simulate either by adding Gaussian image plane noise, or by adding moving objects to the simulation.

- Bouncing: Trackers can find high similarity in nearby locations and track points back and forth between these two locations.
- False Feature: Features drift due to an apparent feature at a depth discontinuity.
- Moving Objects: Arises from a moving object being in the field of view.
- Appearance Change: Drifting features as the appearance changes due to varying viewpoint.

2.1.2 Simulated INS

Along with the visual input, we rely on input from the INS system onboard the vehicle. We assume that the INS consists of a tri-axial gyroscopic sensor and a tri-axial accelerometer. The simulated sensor outputs are computed from first and second derivatives of the ground truth vehicle poses. Our noise model for a gyroscope output, consists of the true vehicle rotation rate, a constant offset, a moving or walking bias, and a wide band sensor noise such as that described in [23]. The wideband sensor noise is assumed to be zero mean Gaussian and is therefore specified by the sample covariance. The gyroscope's moving bias is modeled as a first order Markov process. The noise that drives the bias is zero mean Gaussian with sample covariance of one. In order for us to use this approach to model the gyroscope, an iterative solution to the process has to be used. We use the approach described by Brown in [24]. The resolution of the synthetic noise can be increased by either increasing the number of samples or by taking averages of completely separate random seeds.

2.2 Image-Based 2-Frame Pose Solution

There has been much previous work (summarized well in [25]) on computing camera pose from corresponding image points in two frames. When the internal calibration parameters of the camera are known, the motion can be captured in a matrix knows as the essential matrix. To solve for the essential matrix from points in two frames, we use the standard 7-point linear solution method described in [25] and the 4-point method for the estimation of planar homographies also given in [25]. Another notable approach is the 5-point method for the direct computation of the essential matrix [26]. The seven and four-point algorithms require a linear system solution to find the essential matrix or planar homography, which produces a solution for the given correspondences. Up to two motion and structure solutions can be extracted from the homography using the method described in Section 5.3.3 of [27]. One can also recover pose solutions for a rotation-only motion from a homography, which is not possible from the essential matrix. All of these algorithms can be applied to more than the minimum number of points, which can reduce the noise-sensitivity of the algorithm. If more than the minimum number of points is used, the solution spaces are just taken as the minimum norm subspaces of the desired dimension.

One of the difficult problems in this video-based pose estimation is that outliers (feature points with gross errors) can bias a motion solution sufficiently to make it useless, as is true with any least-squares-based solution. To avoid this deleterious effect of outliers, our approach uses the MSAC (M-estimator Sample Consensus [28] algorithm (a variant of RANSAC [20] which uses a robust error measure) to find a solution based on a reliable subset of the image feature points. Given a set of image point trajectories and a pair of image frames, MSAC randomly selects a minimal set of points and solves for the camera projection matrices and 3D structure describing the image feature motion over the two frames. We currently use either the 7-point fundamental matrix solution, which is converted into an essential matrix when the internal camera parameters are known, or the 4-point homography solution. These motion estimation algorithms are run on many (in our implementation 400-500) random samples of seven or four points (depending on the algorithm chosen) and the best solution is selected and returned. The choice of algorithm is based on the rank properties of the data matrix as described in [25] Section 10.9.2.

The criterion for the selection of the best solution for the two frame problem is measured in two stages. In the first stage the solution quality is characterized in terms of the symmetric epipolar distance [25], which is computed from the residual error between the image points and the epipolar lines in both images. The contribution to this error of a point is $\rho\left(e^2\right)$ where

$$\rho\left(e^2\right)=\begin{cases} e^2 & e^2 < T_h^{\,2} \\ T_h^{\,2} & otherwise \end{cases} \tag{1}$$

and e is the distance of the point's image position to the epipolar line (averaged over both images), and T_h is a threshold based on the expected feature point error. The set of points with $e^2 < T_h^{\,2}$ is the set of inliers for a particular solution. The quality of a particular solution is then based on its first stage residual error, which is computed as the sum of the $\rho\left(e^2\right)$ over all points. The solutions are then further tested in order of increasing first stage residual error by re-computing the solution over all inliers and, in the case of the seven point algorithm, finding the essential matrix nearest to the computed fundamental matrix. Given the 3D locations of the points, we compute the second stage error as the geometric error [25], which is the deviation between the reprojections of the 3D points and the original image feature locations. If no solutions are found with more than the desired number of inliers, the algorithm returns with a failure. Otherwise, the first solution (lowest error in the first stage of screening) with the required number of inliers is returned.

Because of MSAC's iterative nature, the computation time becomes an issue. In order to maintain faster running time, we do not run the MSAC algorithm on every frame, but on a subset of the frames. Detailed descriptions of our method for selecting frames can be found in [3] which enforces the constraint that a minimum set of corresponding features are maintained. This 2-frame essential matrix estimation provides two functions. First, it provides a 3D structure and motion solution that is used to initialize the 3D points in the Kalman. Second, it provides a filtering mechanism for removing the outliers from the feature tracks before they are fed into the Kalman filter.

2.3 Sensor Integration Using a Kalman Filter

Because the function relating a feature point position to an image observation is non-linear, we have evaluated an Extended Kalman Filter (EKF) to simultaneously estimate the motion of the camera and the locations of 3D points in the scene. As input, the Kalman filter uses feature point locations from the tracker, as well as INS data (i.e., raw gyroscope and accelerometer readings). The state vector x to be estimated consists of the following elements:

- Vehicle pose (3 translational and 3 rotational degrees of freedom)
- Translational and angular velocities (six additional degrees of freedom)
- Translational accelerations (3 translational degrees of freedom)
- The (x, y, z) locations of each of the N feature points being tracked

Therefore, the full state vector has a total of $15+3N$ elements. By combining these into a single state vector, we can represent cross-correlation between different axes of the motion, the cross-correlation of the uncertainty between different points, and the cross-correlation between the motion and the structure.

We have found that the EKF convergence is sensitive to the initial state. We have a good initial estimate of the pose of the camera at the beginning of the VAN process from the vehicle's GPS corrected INS pose estimation. However, we also need a good estimate of the 3D structure as feature points are added to the state. In addition, since the EKF assumes a Gaussian state distribution and uses a linearized model for the update step, outliers in measurement data can strongly distort the state estimate. This is why we use the 2-frame solution to provide initial estimates of the feature point locations in 3D and to eliminate outliers from the points provided to the Kalman filter.

We assume that the sensors are asynchronous and have independent noise, so each sensor can be incorporated into the state estimate using a separate measurement update equation. The filter will perform the time update projecting the state from the previous update time to the current update when gyroscope data, accelerometer data, or camera data (in the form of filtered image plane features) is available. The measurement update step will follow to update the filter's state based on the new measurement input. This is a recursive process and it will run whenever the measurement input data is available. The time and measurement update equations are given in [29].

Table 1. Gyroscope Specifications

Rate Gyro	Attribute	Units	Specs
Tactical (CASE I)	Random walk	°/sec/√Hz	0.0017
	Bias Time Constant	Sec	100
	Bias Variation	°/hr	0.35
Automotive (CASE II)	Random walk	°/sec/√Hz	0.05
	Bias Time Constant	sec	300
	Bias Variation	°/hr	180
Consumer (CASE III)	Random walk	°/sec/√Hz	0.05
	Bias Time Constant	sec	300
	Bias Variation	°/hr	360

New features are added to the state vector as they are observed, and old features that are no longer visible are removed. Thus, the state vector and its covariance matrix

can dynamically expand and contract depending on the number of trackable features. The 3D feature locations are initialized in the state vector based on the 2-frame solution. The covariance ellipsoid for the feature location is elongated along the projection line of the feature to capture the larger uncertainty of the feature depth.

3 Results and Analysis

Here we compare three different quality sensor systems – consumer, automotive, and tactical grad - as shown in Table 1 and summarized in [23]. The camera qualities corresponding to the different typical camera systems with resolutions of 2048x2048, 1280x1024, and 640x480 for the tactical, automotive, and consumer grade respectively. We assume in each case that the camera has a lens with a field of view held constant at 45°, and that our sub-pixel feature matching algorithm can match features to $^1/_{10}$ of a pixel. For each of the runs, the inlier ratio is 85%. Since outliers are derived from tracking problems (as described in 0), not camera quality, the outliers will exhibit different amounts of pixel error for the different camera systems. For these experiments, the outlier noise level has been set to 28.47 pixels, 22.78 pixels, and 11.12 pixels for the tactical, automotive, and consumer grade camera systems respectively. For each of these sensor systems we show the computed pose from the 2-frame video-only solution, the INS-only solution, and our combined VAN system. In this experiment we show that the combined VAN approach produces superior results than either of the two systems running independently.

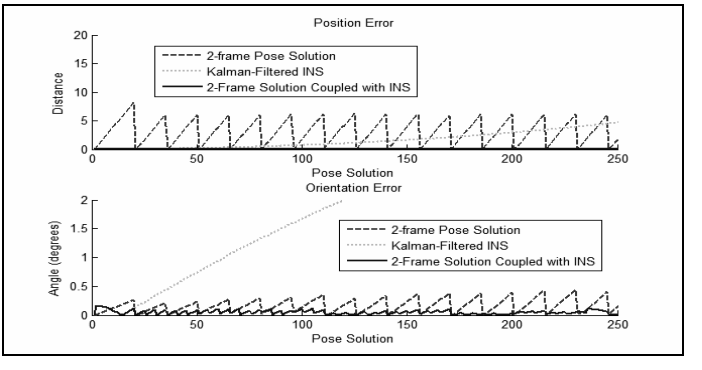


Fig. 2. Pose solutions for comparison of 2-frame video-only solution, INS-based solution, and our VAN system. This plot shows typical results for errors corresponding to a tactical grade (CASE I) INS and imaging system. The distances given are in meters.

3.1 CASE I: Low Noise – Tactical Grade Camera and INS Sensor System

Three pose estimation approaches are tested; the 2-frame video-only pose solution, the INS-based pose solution, and our VAN solution. For the low noise case, the INS data was set to the tactical grade sensor settings given in Table 1 and the camera is set as described in the previous section. Notice in Fig. 2 the video-only solution presents a saw-tooth wave form. This is due to the fact that a solution is computed on only every 3rd frame, thus causing the intermittent frames to have increasing error as the

true pose moves further from the last MSAC solution. Notice also the drift in the INS-only derived pose. This is typical of INS-only systems.

3.2 CASE II: Medium Noise – Automotive Grade INS

The medium noise case corresponds to the values in Table 1 for an automotive grade INS sensor and corresponding imaging system. Notice Fig. 3 shows similar results to the CASE I setup. The combined VAN algorithm performs much better than either of the individual systems. The INS-based system running alone is only slightly worse in this case when compared to the previous, likely due to the higher INS system noise.

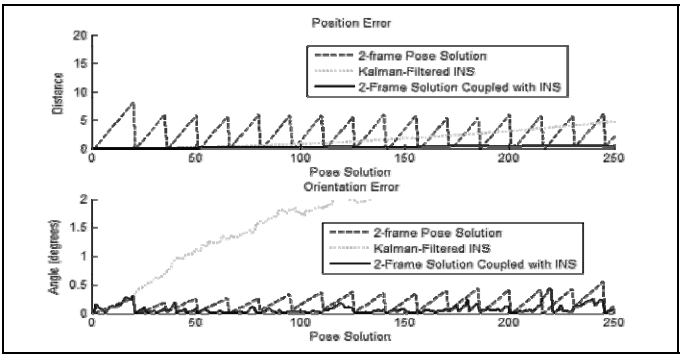


Fig. 3. Pose solutions for comparison of the 2-frame video-only pose solution, the INS-based pose solution, and our VAN system. This plot shows typical results for errors corresponding to an automotive grade (CASE II) INS and imaging system. The distances given are in meters.

3.3 CASE III: High Noise – Consumer Grade INS

For the high noise case, the noise values used on the INS data correspond to the values in Table 1 for the consumer grade INS system and corresponding imaging system.

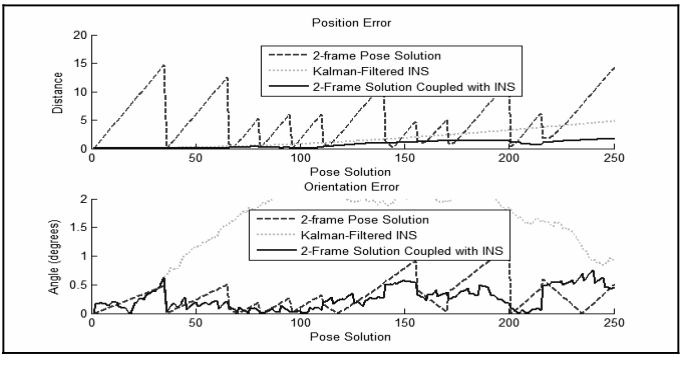


Fig. 4. Pose solutions for comparison of the 2-frame video-only pose solution, the INS-based pose solution and our VAN system. This plot show typical results for errors corresponding to a consumer (CASE III) grade INS and imaging system. The distances provided are in meters.

Notice in Fig. 4 the trends are as expected with higher noise in the combined VAN approach, but still much better than either of the systems individually. Notice in this case, video-only occasionally failed to find a solution. This is likely due to the fact that at these higher noise levels, many of the inliers begin to look like outliers, and if there are not enough inliers, no solution is returned.

4 Discussion and Conclusions

In this work we show that autonomous navigation systems consisting of INS/GPS systems are not adequate due to INS drift when GPS signals are lost. Video-only solutions will not perform robustly when image texture is inadequate for feature tracking or flow based methods. We have proposed and demonstrated a combined VAN approach which merges both INS and vision systems to exploits the benefits of each using a new outlier filtering technique. Our system is still in early development and further testing on real data sequences will be necessary for full validation.

References

1. Debrunner, C.H. and N. Ahuja, Segmentation and factorization-based motion and structure estimation for long image sequences. IEEE Trans Pattern Anal Mach Intell, 1998. **20**(2): p. 206-211.
2. Davison, A.J. *Real-time simultaneous localisation and mapping with a single camera*. in *International Conference on Computer Vision*. 2003. Nice, France.
3. Baker, C., C. Debrunner, and M. Whitehorn. *3D model generation using unconstrained motion of a hand-held video camera*. in *The International Society for Optical Engineering*. 2006. San Jose, CA: Springer.
4. Alenya, G., E. Martnez, and C. Torras, *Fusing visual and inertial sensing to recover robot egomotion.* Journal of Robotic Systems, 2004. **21**: p. 23-32.
5. Diel, D.D., P. DeBitetto, and S. Teller. *Epipolar Constraints for Vision-Aided Inertial Navigation*. in *IEEE Workshop on Motion and Video Computing (WAVC/MOTION'05).* 2005: IEEE Computer Society.
6. Deil, D.D., *Stochastic Constraints for Vision-Aided Inertial Navigation*, in *Mechanical Engineering*. 2005, Massachusetts Institute of Technology. p. 110.
7. Chai, L., *3-D Motion and Structure Estimation Using Inertial Sensors and Computer Vision for Augmented Reality*, in *Engineering Division*. 2000, Colorado School of Mines: Golden, CO. p. 110.
8. Domke, J. and Y. Aloimonos, *Integration of Visual and Inertial Information for Egomotion: a Stochastic Approach*. 2006, Dept. of Computer Science, University of Maryland: College Park, MD.
9. Huster, A. and S.M. Rock. *Relative Position Sensing by Fusing Monocular Vision and Inertial Rate Sensors*. in *International Conference on Advanced Robotics*. 2003. Coimbra, Portugal: Proceedings of ICAR 2003.
10. Makadia, A. and K. Daniilidis. *Correspondenceless Ego-Motion Estimation*. in *IEEE International Conference on Robotics and Automation*. 2005.
11. Nguyen, K., *Inertial Data Fusion using Kalman Filter Methods for Augmented Reality*, in *Engineering*. 1998, Colorado School of Mines: Golden.

12. Qian, G., R. Chellappa, and Q. Zheng, *Robust structure from motion estimation using inertial data.* J. Opt. Soc. Am., 2001. **18**(12): p. 2982-2997.
13. Rehbinder, H. and B.K. Ghosh. *Mulit-rate fusion of visual and inertial data*. in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 2001.
14. Rihbinder, H. and B. Ghosh, *Pose Estimation Using Line-Based Dynamic Vision and Inertial Sensors.* IEEE Transactions on Automatic Control, 2003. **48**(2): p. 186 - 199.
15. Roumeliotis, S.I., A.E. Johnson, and J.F. Motgomery. *Augmenting inertial navigation with image-based motion estimation*. in *IEEE International Conference on Robitics and Automation*. 2002.
16. Strelow, D. and S. Singh. *Online Motion Estimation from Image and Inertial Measurements*. in *Workshop on Integration of Vision and Inertial Sensors (INERVIS)*. 2002.
17. You, S. and U. Neumann. *Integrated Inertial and Vision Tracking for Augmented Reality Registration*. in *Virtual Reality Annual International Symposium*. 1999. Houston**,**TX: IEEE.
18. You, S., U. Neumann, and R. Azuma. *Hybrid inertial and vision tracking for augmented reality registration*. in *IEEE Virtual Reality '99*. 1999: IEEE.
19. Harris, C. and M. Stephens, *A combined corner and edge detector.* Alvey Vision Conference, 1988.
20. Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography.* Communications of the Association of Computing Machinery, 1981. **24**: p. 381-395.
21. Lucas, B.D. and T. Kanade, *An iterative image registration technique with an application to stereo vision.* International Joint Conference on Artificial Intelligence, 1981.
22. Shi, J. and C. Tomasi. *Good Features to Track*. in *IEEE Conference on Computer Visioin and Pattern Recognition (CVPR94) Seattle*. 1994.
23. Flenniken, W., *Modeling Inertial Measurement Units and Analyzing the Effect of their Errors in Navigation Applications*. 2005, Auburn University: Auburn, Al.
24. Brown, R.G. and P.Y.C. Hwang, *Introduction to random signals and applied Kalman filtering*. 2nd ed. 1997, New York: J. Wiley. 502.
25. Hartley, R. and A. Zisserman, *Multiple View Geometry in Computer Vision*. 2001, Cambridge: Cambridge University Press.
26. Nister, D., *An efficient solution to the five-point relative pose problem.* Computer Vision and Pattern Recognition, 2003.
27. Ma, Y. and S. Soatto, *An invitation to 3-d vision: from images to geometric models*. 2004, New York: Springer.
28. Torr, P.H., *MLESAC: A new robust estimator with application to estimating image geometry.* Computer Vision and Image Understanding, 2000. **78**: p. 138-156.
29. Brown, R. and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. 3rd ed. 1992, New York: Wiley.

A Method of Improving Cloud Predictions for Real-Time Weather Forecasting and Visualization

Ramesh Vellore, Darko Koračin, and Melanie Wetzel

Desert Research Institute, Reno, Nevada
Ramesh.Vellore@.dri.edu

Abstract. Indirect or passive observations using satellite remote sensing in the visible, infrared and microwave spectra provide global coverage of the thermal states of the cloud tops or the ground surface. The commonly employed temperature profile matching techniques using satellite data and numerical weather prediction models are only relatively successful in estimating the cloud top height (CTH) for optically dense middle and high clouds (cloud tops at heights generally greater than two kilometers). Therefore, accurate predictions of low-level CTH present a formidable challenge to the forecasting and nowcasting community. In this study, we present an approach to estimating low-level CTH by combining the above-cloud information extracted from the satellite imagery and the below-cloud information obtained from weather station measurements. Assumed ranges of brightness temperature and CTH are used to process the cloudy pixels for visualization and classification purposes. Our study indicates that the CTH evaluated using satellite data confirms the presence of low-level clouds in the range 400-1000 m. Accurate estimates of the boundary layer CTH can provide better low-level cloud products (*e.g.*, fog or clouds formed by fog lifting) for improved weather forecasting and applications in the research community.

1 Introduction

Satellites are equipped with passive sensors that measure the energy emitted from the earth's surface or a cloud. It has become possible to provide better quantification of cloud properties with the help of the increased number of spectral channels and higher resolution imagers on the Geostationary Operational Environmental Satellite (GOES; Menzel *et al.* 1998; Greenwald and Christopher 2000). Reliable estimates of diagnostic cloud products from the space imagery, in particular cloud-top height (CTH), are useful to numerical weather prediction (NWP), atmospheric studies, and climate modeling. The operational method of estimating the CTH using satellite data is still rudimentary. The commonly used method is the temperature matching method (Takacs, *et al.* 2004). This method combines brightness temperatures across the 11 μm window channel and the temperature profile obtained from a NWP model to infer the best estimate of the cloud top altitude. This method works considerably well for optically dense

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 544–553, 2006.
© Springer-Verlag Berlin Heidelberg 2006

deep clouds above 5 km, but fails to detect low-level clouds due to their less pronounced optical properties and the presence of low-level inversions below 2-3 km (Minnis *et al.* 2005). Many studies such as Vellore *et al.* (2006) emphasized the need to accurately estimate cloud top height and improve the NWP model initial and boundary conditions leading to improvement in operational weather predictions of temperature and moisture in the lower atmosphere. This study provides an approach to estimating CTH using the brightness temperature information extracted from infrared channels, and ground-truth measurements for a nocturnal case study over northern and central Nevada. This paper presents the results of a first step in a process of detecting low-level cloud information, and can lead to further methodology enhancements for diurnal, seasonal, and climate predictions of low-level clouds over complex terrain.

2 Method for Cloud Classification

GOES is operated by the National Oceanic and Atmospheric Administration and carries a visible-and infrared channel-Imager instrument detecting different wavelengths of radiation energy (visible channel 1: 0.62 μm, shortwave infrared channel 2: 3.9 μm, longwave channel 3: 6.7 μm, longwave channel 4: 10.7 μm, and longwave channel 5: 12 μm). The Imager provides cloud distribution and cloud-top temperature (CTT) at a pixel resolution of 4 km for the infrared channels. The pixel resoution of the visible imagery is 1 km. A flowchart for nighttime low-level cloud classification is shown in Fig. 1.

The infrared satellite imagery on 29 June 2006 at 1100 UTC and 1200 UTC are considered for cloud classification in this study. The domain coverage is about 400 $\times$ 400 km^2, spanning between 116.5-120.5^{o}W and 37.5-41.0^{o}N (see Fig. 2). During this time, the atmosphere was stably stratified in the lowest 2 km throughout the entire valley of northern and central Nevada, and the sky was overcast with low altitude clouds. The cloud-base height data from the ceilometer (an instrument that uses a light beam to record the altitude of the lowest cloud layer) was obtained from the surface weather station (B17) located at the Naval Air Station, Fallon, Nevada (39.32^{o}N, 118.22^{o}W, elevation = 1290 m).

The preliminary identification of cloudy pixels is based on the surface air temperature at B17 and the channel 4 (11 μm) brightness temperature. Low-level clouds usually are characterized by CTT above 281.15 K (8^{o}C), and this condition segregates the low-level clouds from the clouds above. The difference (contrast) between the window channel (11 μm) and the shortwave IR spectral channel brightness temperature (3.9 μm) is an approximate exponential function based on cloud drop sizes and column-integrated cloud water amounts. Smaller (larger) contrast indicates optically thin (thick) clouds. Contrast values of less than -2 K (units in Kelvin) are identified as stratus clouds (Lee *et al.* 1997); if the CTH is less than 300 m and the surface relative humidity (SRH) is more than 95%, it is identified as fog. Optically thick low-level clouds (stratocumulus) are identified by a contrast between -2 and -0.1 K. The contrast ranges were typically between -5 and 2 K at 1100 and 1200 UTC on 29 June 2006. There

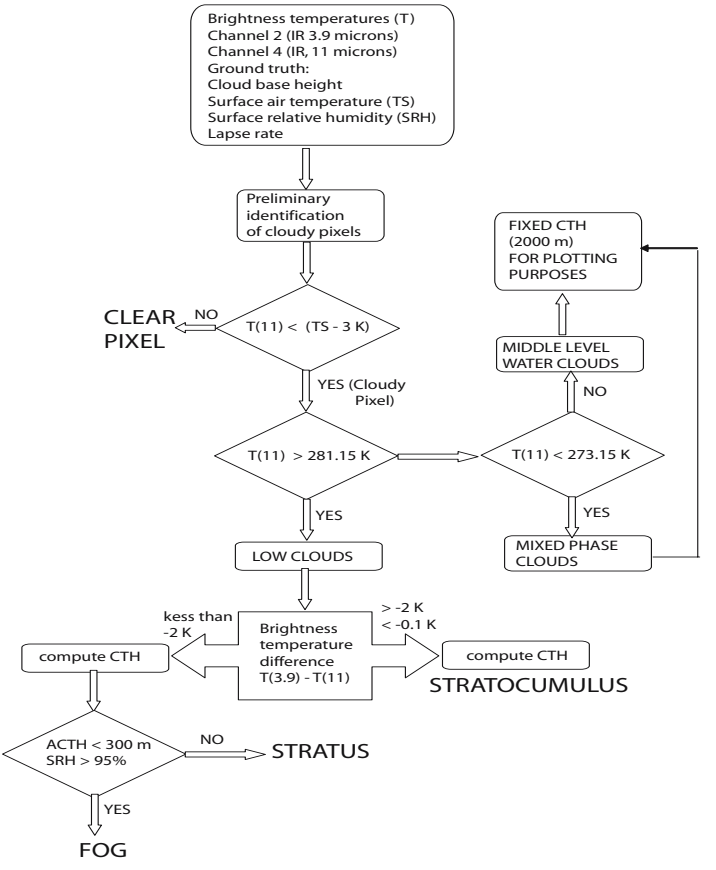


Fig. 1. Flowchart for nighttime low-level cloud identification and classification

were only a few pixels indicating cloud tops radiating at temperatures below the freezing level. The clouds with CTT less than 288.15 K and below the freezing level are classified as middle-level water clouds. It should be noted that the middle-level clouds could be multi-layered with low-level clouds present below. If the difference between satellite-derived brightness temperatures and the air temperature at the surface is within 3 K, the pixels are identified as clear air pixels. Further checks can be performed on the clear air pixels based on the terrain type, either desert or vegetated surfaces. Since the domain was predominantly filled with cloudy pixels, this check was not performed.

The variation of temperature with height (*i.e.*, the temperature lapse rate, Γ) in the atmosphere controls the potential for convective activity in the atmosphere. Adiabatic conditions of temperature decreasing at a certain rate with height are usually assumed for the environmental lapse rate. The computation procedure consists of finding Γ using the data at the Fallon B17 station (Table 1) which is representative for the entire domain of interest, and it is used to compute the CTH for all the satellite cloudy pixels. As the low-level clouds are of

primary interest in this study, whose thickness are less than a few hundred meters, appropriate estimates of cloud base temperature (T_{CB}), which is greater than the satellite-derived CTT, are rather essential. Small variations in the surface air temperature (T_s) are rather sensitive in the quantitative estimation of T_{CB}. There was a lot of variability in the air temperature measurements at B17 due to gusty winds affecting the accuracy of the temperature sensor measurements near the surface during 1100-1200 UTC on 29 June 2006. Since satellites passively measure the temperature of the radiating surface (or the skin temperature of the body) and the ground-truth soil measurements are relatively unaffected by the wind gustiness near the surface, soil temperatures (T_{soil}) are used in this study to deduce T_s. On average, the nighttime T_{soil} at B17 was 9-10 K greater than the air temperature during the period of interest. The representative T_s is deduced from the T_{soil} measurements at B17 as follows:

$$T_s = T_{soil} - 9K \tag{1}$$

A difference of 9 K was consistent with another test which was performed using a 6-hour average of surface air temperature measurements prior to the period of gusty winds at B17. The steps of the computation procedure are:

1. Following the dry adiabatic lapse rate (Γ_{dry} = - 9.8 K km^{-1}) in the subcloud layer, the cloud base temperature (T_{CB}) at B17 is calculated as follows:

$$T_{CB} = T_s + (\Gamma_{dry} \times CBH) \tag{2}$$

   where CBH is the cloud base height measured by the ceilometer.
2. Following the wet adiabatic lapse rate (Γ_{wet} = -6.5 K km^{-1}) in the cloud layer, the CTH at B17 is determined using T_{CB} and the satellite-derived CTT as follows:

$$(CTH)_{B17} = \frac{(CTT)_{B17} - T_{CB}}{\Gamma_{wet}} \tag{3}$$

   where $(CTT)_{B17}$ is the cloud top temperature at B17 obtained from the colocated satellite pixel in the satellite imagery.
3. Using the CTT, CTH, and T_s values, the temperature lapse rate (Γ) between the cloud top and the surface is calculated as follows:

$$\Gamma = \frac{(CTT)_{B17} - T_s}{(CTH)_{B17}} \tag{4}$$

4. Following Minnis *et al.* (1992) and Koračin *et al.* (2003), the CTH of the cloudy pixels for the entire domain of interest is therefore calculated as follows:

$$CTH = \frac{CTT - T_s}{\Gamma} \tag{5}$$

A similar procedure can be used to classify low-cloud types and obtain CTH over the ocean. It is more feasible to assume the lapse rate of adiabatic conditions over the ocean, as the surface temperature variability is not high for the periods of interest. Also, properties such as moisture content in the boundary layer most of the time is uniformly well mixed. Koračin *et al.* (2003) showed that adiabatic assumptions for low-level CTH computation over the ocean agree well with airborne observations in the U. S. West Coast. The lower-atmospheric properties over the land change rapidly in a few hours and pose great difficulty to commonly used adiabatic assumptions for the computation of CTH.

A multi-channel classification procedure can be performed by including the visible cloud reflectance during the daytime (Wetzel *et al.* 2001). Estimates of the cloud layer liquid water path can be made during the daytime by fitting the difference between thermal and near infrared brightness temperatures and visible reflectances for representative clouds of variable optical path, but this is not possible during night time when visible reflectance is not available. Therefore, a broader contrast range was assumed in this study for fog and stratus cloud types.

3 Discussion

A total of 95481 (309 × 309) pixels was considered for classification and CTH computation from a single satellite image (Fig. 2). Among these pixels, about 70% are classified in the fog and low-level cloud categories (0% fog, 8% stratus, 62% stratocumulus) with CTH ranging between 400-1000 m. This result is in general agreement with the Naval Research Laboratory (http://www.nrlmry.navy.mil) cloud classification. About 25% are classified in the middle-level cloud categories; 5% of the pixels are classified as no clouds, and the brightness temperatures in these pixels are comparable with the surface temperature at B17. The maximum (minimum) cloud top among the low-level cloud pixels was at 1050 m (350 m). Among the low-level clouds, a large percentage lie in the CTH range 700-1000 m, and are categorized as optically thick. The percentages of low-cloud classified pixels in the various CTH bins are shown in Table 2.

This study summarizes a method and a strategy for evaluating the cloud top heights based on a combination of infrared reflectance values from the geostationary (GOES) satellites with the ground-truth measurements. It should be mentioned that CTH is not observed directly, except by LIDAR (LIght Detection And Ranging) observations which are usually only available for research purposes. Ceilometers view the cloud base from the ground, and the satellites give the cloud top view. Ceilometers give the CBH and not the cloud base temperature, whereas the satellites give the CTT and not the CTH. Due to these observational limitations, it is not currently possible to verify the accuracy of this method described in this study in a strict sense. However, the incorporation of the CTH information obtained from this method into NWP models for improving the cloud predictions and visualization of three-dimensional cloud objects can provide a robust evaluation of how modeled clouds reflect the reality of

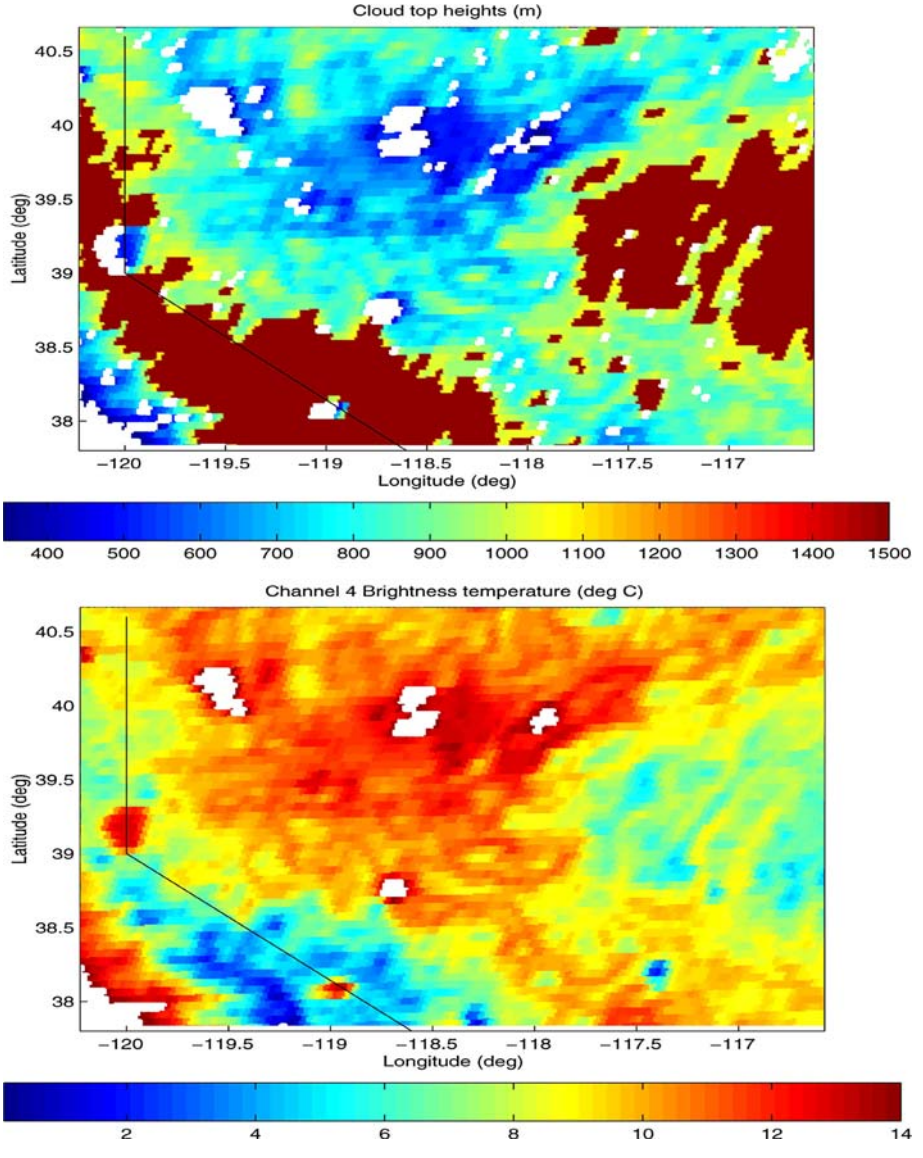


Fig. 2. Top panel: Cloud top heights at 1200 UTC on 29 June 2006. Bottom panel: Channel 4 (11 μm) brightness temperatures at 1200 UTC on 29 June 2006. Latitudes are in degrees north and longitudes are in degrees west (negative).

remotely sensed data as closely as possible. Also, the method used in this study utilizes ground-based measurements with the satellite data to provide better structure of the atmospheric boundary layer within the forecasting domain, or for any specific localized regions of interest.

In a subsequent study, the skill in estimating the low-level CTH can be further enhanced by optimally obtaining the lapse rate of the lower atmosphere for various times of the day and year from radiosonde observations of the atmospheric temperature profiles provided by the network of weather stations (for example, Reno at the western side and Elko at the eastern side of this domain shown in Fig. 2) using neural network aids.

Table 1. Data from Naval Air Station (B17), Fallon, Nevada on 29 June 2006 at 1200 UTC. Data source: Western Regional Climate Center (http://www.wrcc.dri.edu), Desert Research Institute, Reno, Nevada.

Variable	Value
Observed:	
Soil temperature	299.4 K
Surface pressure	875.2 hPa
Relative humidity	69%
CBH (ceilometer)	548 m
CTT (satellite)	283.3 K
Estimated:	
T_s	290.4 K
T_{CB}	285.1 K
Cloud thickness	269 m
Lapse rate (Γ)	-8.8 K km^{-1}
CTH	817 m

Table 2. Percentage of low-cloud cloud-top height bins for the entire domain on 29 June 2006 at 1100 and 1200 UTC

CTH bins (m)	1100 UTC	1200 UTC
400-500	2.0	3.9
500-600	6.1	8.0
600-700	12.8	15.1
700-800	19.2	19.3
800-900	26.7	24.3
900-1000	21.2	18.2

4 Method Application for Real-Time Forecasting and Visualization

The analysis towards improvement of cloud predictions in a real-time forecasting system follows a sequence of steps: (1) satellite data collection and quality control, (2) cloud classification, (3) data assimilation of cloud information into NWP models, (4) forecasting of cloud structure and evolution, and (5) visualization and assessment of cloud products. The main advantage of the developed

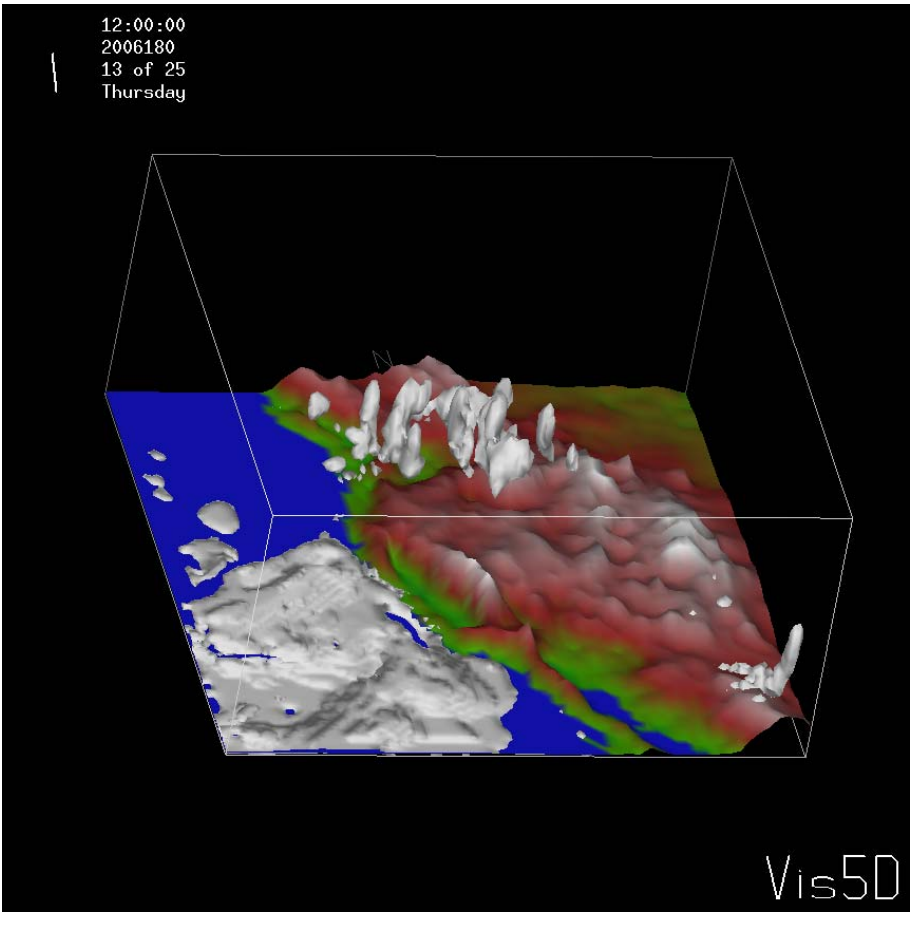


Fig. 3. Visualization of cloud structure in the real-time weather forecasting system for the western U.S. and eastern Pacific Ocean

method is to improve the inherently inaccurate initial and boundary conditions obtained from global and regional-scale models with coarse horizontal and vertical resolution that are required input to the high-resolution forecasting and nowcasting.

The real-time forecasting system using Mesoscale Model 5 (MM5; Grell *et al.* 1995) was developed at Desert Research Institute (DRI; http://www.adim-dev.dri.edu). It consists of five interactive domains with horizontal grid resolutions in the ratio 3:1 ranging from 27 km to 333 m focusing on western Nevada. The atmosphere is to be divided into 40 levels, out of which 25 vertical levels are designed to be below 3 km. The meteorological variables are predicted at every level. The variables that are of primary interest for visualization are water substance mixing ratios in the atmosphere.

The main objective of the real-time forecasting system is to support naval aircraft and rotorcraft operations at the Fallon Naval Air Station. One of the most critical parameters for the operational air safety is visibility, which is primarily affected by the presence of low-level clouds. Due to the complexity of the airborne operations, the forecasted cloud parameters need to be visualized in three-dimensional space and time. An example of the visualization of cloud water product in this forecasting system is shown in Fig. 3. The three-dimensional envelope bordered by the minimum cloud water mixing ratio of 0.1 g kg^{-1} is visualized on the outermost model domain that encompasses the western U.S. and part of the eastern Pacific. The figure shows the complex structure of the cloud over the ocean and land due to synoptic and topographic effects.

The method described in this study will provide support towards improvement of the cloud predictions. The developed method is being tested as a pre-processing module in a real-time forecasting and nowcasting system. The application of this method will be a valuable resource in obtaining more accurate forecasted cloud parameters and subsequent visualization to support aircraft operations in the considered area.

Acknowledgements

This study was supported by Office of Naval Research grant N00014-06-0028. The authors gratefully acknowledge the Western Regional Climate Center (WRCC) for obtaining the data for the Naval Air station Area at Fallon, Nevada and other stations over Nevada. The authors express special appreciation to Mr. Travis McCord of the Desert Research Institute (DRI) for technical preparation of the manuscript.

References

Grell, G. A., J. Dudhia, and D. R. Stauffer, 1995: A description of the fifth-generation Penn State/NCAR Mesoscale Model (MM5). *NCAR Tech. Note.* TN-398+STR, 122 p. [http://www.mmm.ucar.edu/mm5/doc1.html]

Koračin, D., J. Powers, M. Wetzel, S. Chai, and N. Adhikari, 2003: Improving prediction of the marine coastal clouds using satellite and aircraft data. *Proc. of 5th conference on coastal atmospheric and oceanic prediction and processes*, American Meteorological Society, 138-140.

Lee, T. F., F. J. Turk, and K. Richardson, 1997: Stratus and fog products using GOES-8-9 3.9 μm data. *Weather and Forecasting*, **12**, 664-677.

Menzel, W. P., and Coauthors, 1998: Application of GOES 8/9 soundings to weather forecasting and nowcasting. *Bulletin of American Meteorological Society*, **79**, 2059-2077.

Minnis, P., P. W. Heck, D. F. Young, C. W. Fairall, and J. B. Snider, 1992: Stratocumulus cloud properties derived from simultaneous satellite and island based instrumentation during FIRE. *Journal of Applied Meteorology*, **31**, 317-339.

Minnis, P., and Coauthors, 2005: Near real-time satellite cloud products for nowcasting applications, WWRP symposium on Nowcasting and very short-range forecasting, Toulouse, France 5-9 September 2005.

Greenwald, T. J., and S.A. Christopher, 2000: The GOES I-M Imagers: New tools for studying microphysical properties of boundary layer stratiform clouds. *Bulletin of American Meteorological Society*, **81**, 2607-2619.

Takacs, A., B. Brown, and J. Mahoney, 2004: Verification of oceanic weather diagnoses and forecasts for avaiation weather elements. *Preprints, 84th Annual Meeting of the American Meteorological Society*, Seattle, 11-15 January 2004.

Vellore, R., D. Koračin, M. Wetzel, S. Chai, and Q. Wang, 2006: Challenges in mesoscale prediction of nocturnal stratocumulus-topped marine boundary layer and implications for operational forecasting. Manuscript under revision, *Weather and Forecasting*.

Wetzel, M. A., and Coauthors, 2001: Evaluation of COAMPS forecasts using satellite retrievals and aircraft measurements. *Weather and Forecasting*, **16**, 588-599.

An Efficient Hardware Architecture for Full-Search Variable Block Size Motion Estimation in H.264/AVC

Seung-Man Pyen[1], Kyeong-Yuk Min[1], Jong-Wha Chong[1,*] and Satoshi Goto[2,**]

[1] Dept. of Electronic Engineering, Hanyang University, Seoul, Korea
[2] Graduate School of Information, Production and Systems at Waseda University, Kitakyushu, Japan
himani815@hotmail.com

Abstract. In this paper, we propose a high speed hardware architecture for the implementation of full-search variable block size motion estimation (VBSME) suitable for high quality video compression. In the high-quality video with large frame size and search range, the memory bandwidth is mainly responsible for throughput limitations and power consumption in VBSME. The proposed architecture is designed for reducing the memory bandwidth by adopting "meander"-like scan for a high overlapped data of the search area and using on-chip memory to reuse the overlapped data. We can reuse the previous candidate block of 94% to the current one and save about 23% memory access cycles in a search range of [-16, +15]. The architecture has been prototyped in Verilog HDL, simulated by ModelSim and synthesized by Synopsys Design Compiler with Samsung 0.18um standard cell library. Under a clock frequency of 51MHz, The simulation result shows that the architecture can achieve the real-time processing of 720x576 picture size at 30fps with the search range of [-16~+15].

1 Introduction

A video sequence usually contains a significant amount of temporal redundancy. The block matching algorithm (BMA) based on the motion estimation and compensation is widely used in many video coding standards such as H.26x, MPEG-1, -2, and -4 to remove temporal redundancy. The fixed block size block matching algorithm (FBSBMA) is to divide the current frame into several macroblocks, and to search for a best matched block within a search range in a reference frame. In case of FBSBMA, if a macroblock consists of two objects moving into different directions, the coding performance of the macroblock is worse than that of two objects moving into one direction. To compensate for the demerits of FBSBMA, variable block-size block matching algorithm (VBSBMA) is adopted in the advanced video coding standards.

---

[*] Member, IEEE.
[**] Fellow, IEEE.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 554 – 563, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In the H.264/advanced video coding (AVC), VBSBMA consists in partitioning a macroblock into 7 kinds of blocks including 4x4, 4x8, 8x4, 8x8, 8x16, 16x8 and 16x16 as it is shown in Fig. 1.

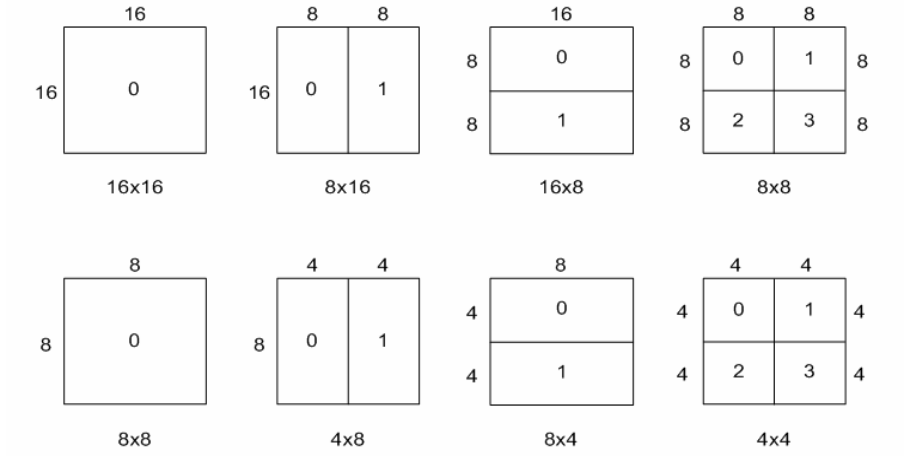


Fig. 1. The various block sizes in H.264/AVC

In this way, the coding performance is improved, because BMA performs the function of making a macroblock segmented into 7 different sizes. Although VBS-BMA achieves higher coding performance than that of FBS-BMA, it requires a high computation effort since 41 motion vectors of 7 different sizes should be computed for each macroblock. Therefore, many efficient hardware architectures such as systolic array [8], 1-D processing element (PE) array [6] and 2-D PE array [4] [7] [10] have been proposed for implementing VBS-BMA. The 1-D PE array is a simple structure, as it is easier to control and less gates than a 2-D PE array, but it is normal to search the sum of absolute difference (SAD) against only one row or a column of the macroblock at a time. On the other hand, the 2-D PE array is a complex structure as it is more difficult to control and has more gates than a 1-D PE array is and has, but it should compute the motion vector of a search point at a time. So 2-D array is suitable structure for high quality video. In the high-quality video with large frame size and search range, the memory bandwidth is a major bottleneck in the motion estimation architecture. To reduce memory bandwidth, the motion estimation architecture has to access each pixel just once and reuse the overlapped data. The architecture is designed for reducing the memory bandwidth by adopting “meander”-like scan in order to access to each pixel just once and by using on-chip memory to reuse the overlapped data. The proposed architecture can perform full-search VBSBMA and it can achieve all 41 motion vectors of the macroblock.

The rest of this paper is organized as follows:

In section II, the proposed architecture is described in detail. The experimental result and the conclusion are given in section III and IV, respectively.

2 The Proposed Architecture

Fig. 2 shows the block diagram of the proposed architecture, which consists of an on-chip memory, a ME control unit, a 16x16 PE array, a SAD adder-tree and a comparator unit.

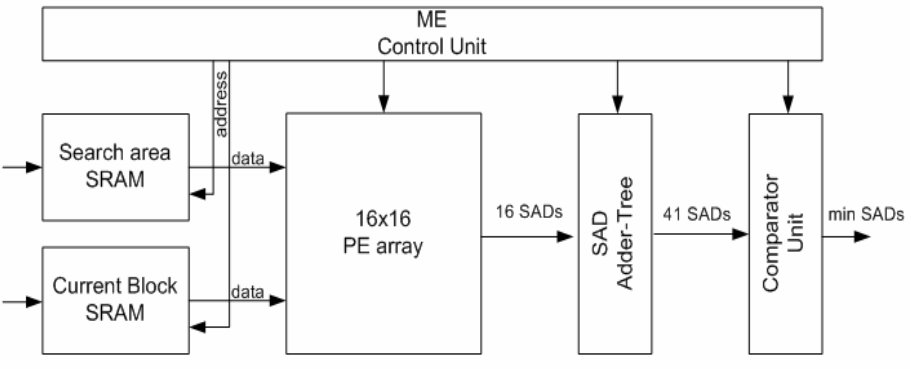


Fig. 2. Schematic overview of the proposed architecture

To reuse the overlapped data, the search area static random access memory (SRAM) is the SRAM to store the pixel data of the search area and the current block SRAM is the SRAM to store the pixel data of the current macrobock. The ME control unit generates the address signals of the search area SRAM and the current macroblock SRAM, and generates the control signals to control other blocks. 16x16 PE array computes sixteen 4x4 SADs of the 16x16 candidate block. The Adder-tree is employed to compute the SADs of all the 41 subblocks from the SADs of the 4x4 subblocks. Comparator unit should find out the best matched MV with the minimum SAD.

2.1 The Data Scheduling for Data-Reuse

In the full-search BMA, each current macroblock should be compared with all the candidate blocks in search area in order to find the best matched macroblock. So the full-search BMA require a high memory bandwidth. In order to reduce memory bandwidth, we adopted the "meander"-like scan (Fig.3) of the search area that obtains the high overlapped data of the search area.

If the search point moves into the horizontal direction, the candidate blocks are many overlapped again and again in comparison with the previous candidate block. 16x18 bit pixels data of the present candidate block can be only changed in comparison with the previous candidate block. In the case of Raster order scan, 16x16 pixels data of the candidate block can be changed when the search point is changed to the next line. As 16x16 pixels data of the candidate block is changed when the search point is changed to the next line, 16x16 PE array needs more 15 cycles than "meander"-like scan in order to load the pixel data of the candidate block. In order to

reduce 15 cycles each of PE needs search data buffer to store the pixel data of the candidate block [14].

The proposed architecture adopted a "meander"-like scan of the search area that gives an actual support to simple control and efficient memory size. The "meander"-like scan format has the three directions of the data-flows. In the odd line of the search area, the direction of the data-flow moves into the right. In the even line of the search area, the direction of the data-flow moves into the left. If the computations for the rows of the search area are finished, the direction of the data-flow moves into the bottom.

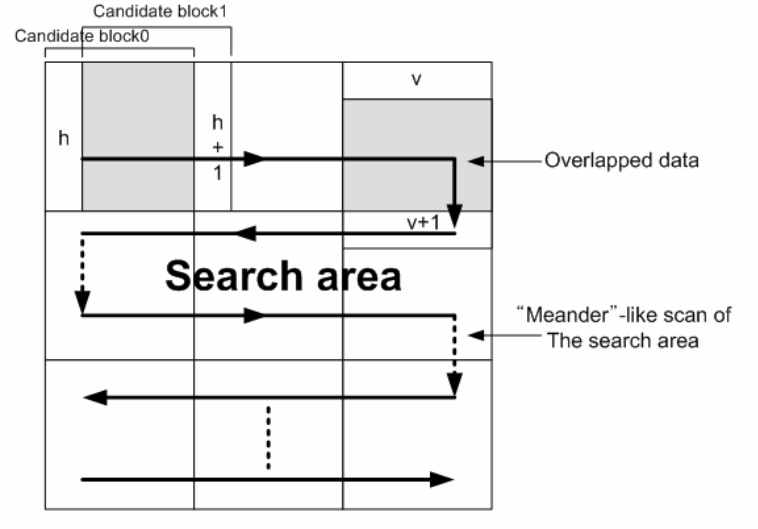


Fig. 3. The "meander"-like scan of the search area

2.2 The PE Architecture

Fig.4 shows the architecture of the PE.

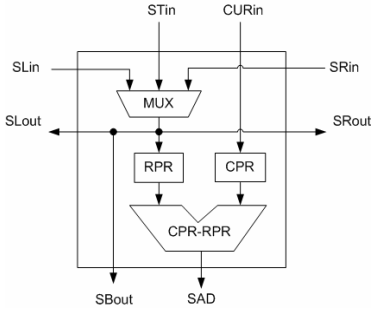


Fig. 4. Processing element

The PE consists of a current pixel register (CPR), a reference pixel register (RPR), a 3 input multiplexer, and a computing unit. The CPR stores the pixel data of the current macroblock, and the RPR stores the pixel data of candidate block. The 3 input multiplexer in the PE selects the direction of the data-flow. The computing unit computes the absolute difference between the CPR and the RPR at every cycle.

2.3 The Process Unit (PU) Architecture

Fig.5 shows the architecture of the PU.

The PU consists of a 4x4 PE and 5 adder-trees. Sixteen SAD, which are the outputs of a 4x4 PE array, can be added to get a 4x4 SAD by using the 5 adder-trees.

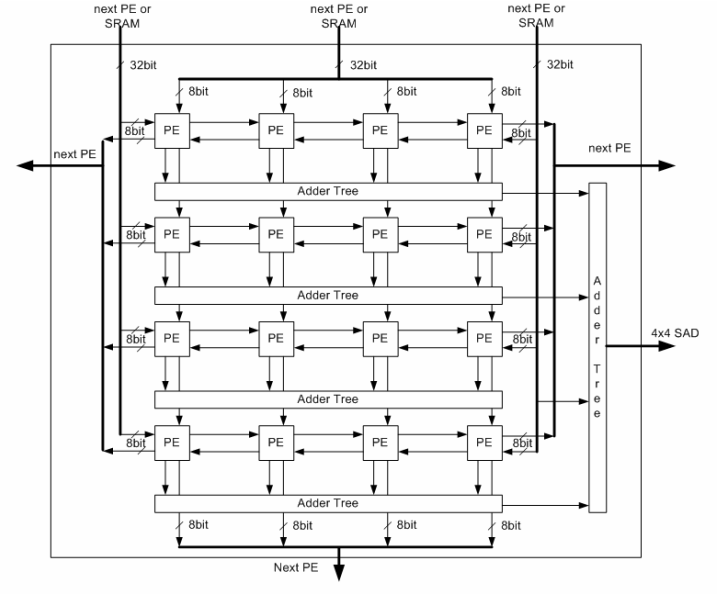


Fig. 5. Processing unit

2.4 The Whole Architecture

The proposed architecture is illustrated in Fig.6.

The architecture is composed of 16 processing units (PU), a SAD adder-tree and a comparator unit. Two SRAM modules are used to store the search area and the current macroblock. Each SRAM module is composed of 16 SRAM's in order to manipulate address generation. In the SAD adder-tree (Fig.7), the SADs of sixteen 4x4 subblocks are used to obtain the SADs of all the 41 subblocks of 7 different sizes.

The comparator unit that composed of 41 comparing elements (Fig.8) find out the minimum distortions as well as the corresponding motion vectors.

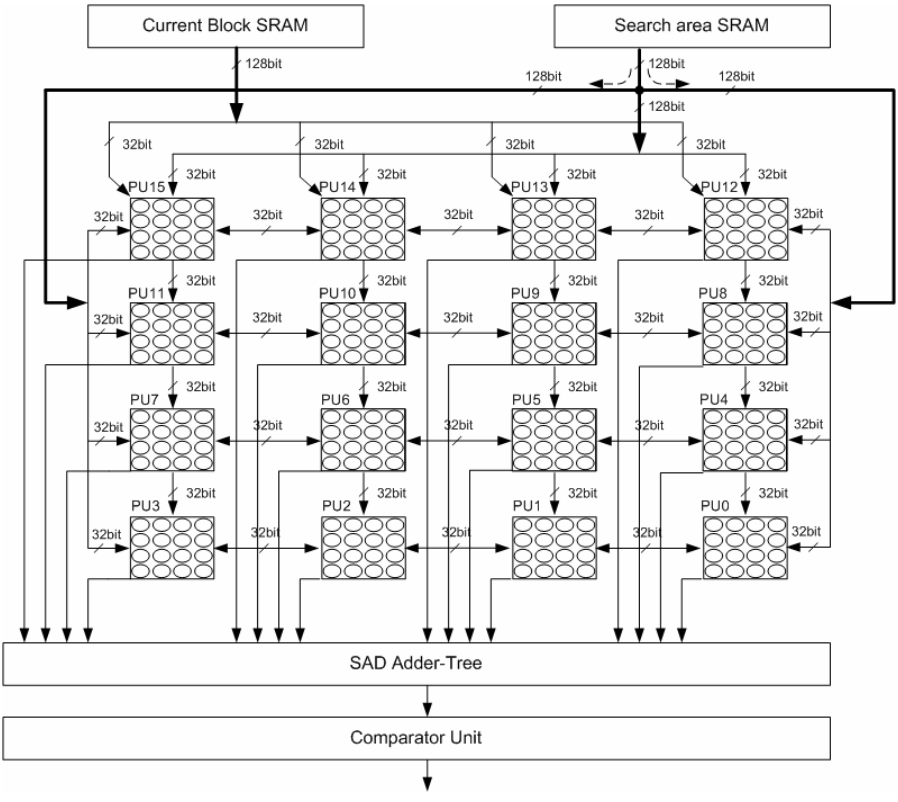


Fig. 6. The whole architecture

Comparing element is consisted of a comparator and two registers. R1 register stores the minimum SAD for comparison and R2 register stores the motion vector.

Table 1 shows the data sequence in the PE array for the current macroblock and the search area. C(x,y) is the pixel data in the current macroblock and R(x,y) is the pixel data in the search area.

At the beginning of VBSBMA, 16x1 8bit pixels data of the current macroblock in the current block SRAM are inputted to the 16x16 PE array during the first 16 cycle. And 16x1 8bit pixels data of the candidate block in the search area are inputted to the 16x16 PE array, in the same cycle. 16x1 pixels data of the current macroblock is inputted in the 16x16 PE array during the 16 cycle, and 16x1 pixels data of the candidate block is inputted in the 16x16 PE array at every cycle. After the placement of the pixel data of the current macroblock and the pixel data of the candidate block during the 16 cycles, the sixteen 4x4 SADs of the candidate block can be computed cycle by cycle. The calculation of 41 SADs in the adder-tree needs 1 more cycle after computing sixteen 4x4 SADs. Therefore, for the search range of [-16 ~ + 15], the calculation of all motion vectors needs 16+1+32x32 = 1041 clock cycles.

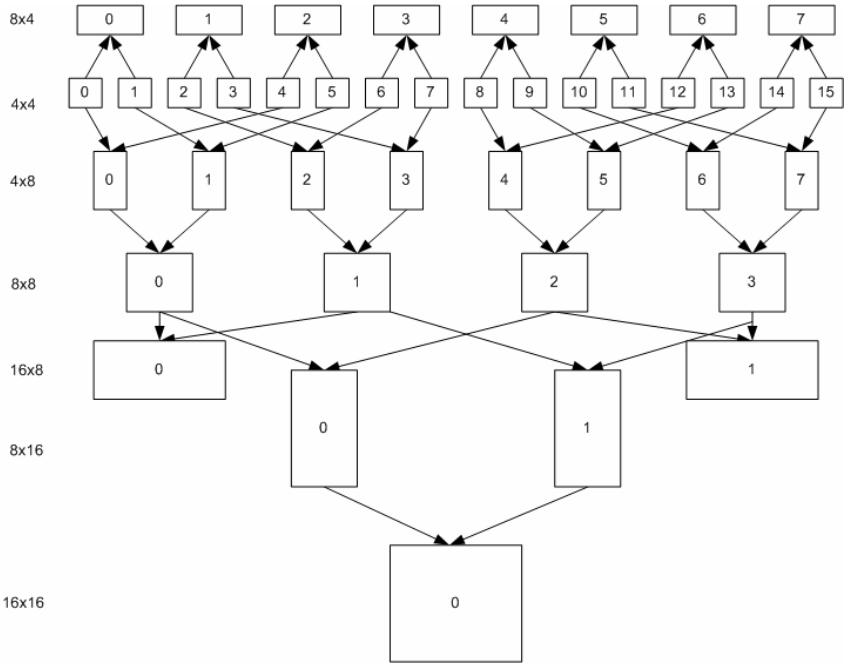


Fig. 7. The SAD adder-tree

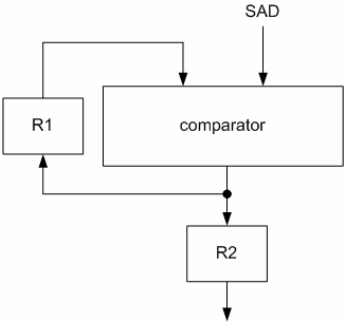


Fig. 8. The Comparing element

Table 1. Data-flow Schdule

CLK	1st column				2nd column				--	16th column			
--	PE0	PE1	--	PE15	PE16	PE17	--	PE31	--	PE240	PE241	--	PE255
0									-				C(0,0) R(-16,-16)
--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	C(0,0) R(-16,-16)	C(0,1) R(-16,-15)	--	C(0,15) R(-16,-1)	C(1,0) R(-15,-16)	C(1,1) R(-15,-15)	--	C(1,15) R(-15,-1)	--	C(15,0) R(-1,-16)	C(15,1) R(-1,-15)	--	C(15,15) R(-1,-1)
16	C(0,0) R(-15,-16)	C(0,1) R(-15,-15)	--	C(0,15) R(-15,-1)	C(1,0) R(-14,-16)	C(1,1) R(-14,-15)	--	C(1,15) R(-14,-1)	--	C(15,0) R(0,-1s6)	C(15,1) R(0,-15)	--	C(15,15) R(0,-1)

Table 1. (*continued*)

17	C(0,0) R(-14,-16)	C(0,1) R(-14,-15)	--	C(0,15) R(-14,-1)	C(1,0) R(-13,-16)	C(1,1) R(-13,-15)	--	C(1,15) R(-13,-1)	--	C(15,0) R(1,-16)	C(15,1) R(1,-15)	--	C(15,15) R(1,-1)
--	--	--	--	--	--	--	--	--	--	--	--	--	--
46	C(0,0) R(15,-16)	C(0,1) R(15,-15)	--	C(0,15) R(15,-1)	C(1,0) R(16,-16)	C(1,1) R(16,-15)	--	C(1,15) R(16,-1)	--	C(15,0) R((30,-16)	C(15,1) R((30,-15)	--	C(15,15) R((30,-1)
47	C(0,0) R(15,-15)	C(0,1) R(15,-14)	--	C(0,15) R(15, 0)	C(1,0) R(16,-15)	C(1,1) R(16,-14)	--	C(1,15) R(16,0)	--	C(15,0) R((30,-15)	C(15,1) R((30,-14)	--	C(15,15) R((30,0)
48	C(0,0) R(14,-15)	C(0,1) R(14,-14)	--	C(0,15) R(14, 0)	C(1,0) R(15,-15)	C(1,1) R(15,-14)	--	C(1,15) R(15,0)	--	C(15,0) R(29,-15)	C(15,1) R(29,-14)	--	C(15,15) R(29,0)
--	--	--	--	--	--	--	--	--	--	--	--	--	--
78	C(0,0) R(-16,-15)	C(0,1) R(-16,-14)	--	C(0,15) R(-16, 0)	C(1,0) R(-15,-15)	C(1,1) R(-15,-14)	--	C(1,15) R(-15,0)	--	C(15,0) R(-1,-15)	C(15,1) R(-1,-14)	--	C(15,15) R(-1,0)
79	C(0,0) R(-16,-14)	C(0,1) R(-16,-13)	--	C(0,15) R(-16, 1)	C(1,0) R(-15,-14)	C(1,1) R(-15,-13)	--	C(1,15) R(-15,1)	--	C(15,0) R(-1,-14)	C(15,1) R(-1,-13)	--	C(15,15) R(-1,1)
80	C(0,0) R(-15,-14)	C(0,1) R(-15,-13)	--	C(0,15) R(-15, 1)	C(1,0) R(-14,-14)	C(1,1) R(-14,-13)	--	C(1,15) R(-14,1)	--	C(15,0) R(0,-14)	C(15,1) R(0,-13)	--	C(15,15) R(0,1)
--	--	--	--	--	--	--	--	--	--	--	--	--	--
1039	C(0,0) R(-16,15)	C(0,1) R(-16,16)	--	C(0,15) R(-16, 30)	C(1,0) R(-15,15)	C(1,1) R(-15,16)	--	C(1,15) R(-15,30)	--	C(15,0) R(-1,15)	C(15,1) R(-1,16)	--	C(15,15) R(-1,30)

3 The Simulation and Synthesis Result

The architecture has been prototyped in Verilog HDL, simulated by ModelSim and synthesized by Synopsys Design Compiler with Samsung 0.18um standard cell library. The total gate count is about 128k. The gate of 16x16 PE array requires 90.7k and the rest gates are spent on the comparator unit to find out the minimum SAD, the adder-tree to obtain the SAD of all the 41 subblocks, and the control unit. For the search range of [-32~ +31], the total 58kb on-chip memory is required to store the search area and the current macroblock. Table 2 shows the performance of the proposed architecture.

Table 2. The Performance of the proposed architecture

Algorithm	Full Search
Number of PE	256
Searching range	32x32, 64x64
Gate count	128k
On-chip memory	58kbits
Process	Samsung 0.18um standard cell library
Block size	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16

Table 3 shows the comparison between the proposed architecture and other full-search VBSME architectures.

Table 3. Comparison of four hardware architecture for VBSME

algorithm	[6]	[7]	[5]	proposed
Number of PE	16	16x16	16x16	16x16
Searching range	32x32 16x16	64x64	64x64, 32x32	64x64 32x32
Gate count	108k	-	154k	128k
On-chip memory	-	96k bits	60k bits	58kbits
Process	0.13um	0.5um	0.28um	0.18um
Block size	7 kind of block sizes	16x16, 8x8, 4x4	7 kind of block sizes	7 kind of block sizes

From the Table 3 it can be observed that the architecture requires less gate count and on-chip memory.

4 Conclusion

We proposed a high speed hardware architecture of VBSME suitable for high quality video compression in this paper. The architecture consists of a 16x16 PE array, an adder tree and comparator unit to find out 41 motion vector and minimum SADs for the 7 different blocks. The proposed architecture can reduce the memory bandwidth by adopting a "meander"-like scan for a high overlapped data of the search area and using on-chip memory to reuse the overlapped data. Compared with the raster order scan of the search area, the "meander"-like scan can save about 23% memory access cycles in a search range of [-16, +15]. The architecture allows the real-time processing of 720x576 picture size at 30fps with a search range of [-16~+15], under a frequency of 51MHz.

References

1. De Vos, L., Schobinger, M.: VLSI Architecture for a Flexible Block Matching Processor, Vol. 5. NO. 5. IEEE Transactions on Circuits and Systems (1995)
2. Jen-Chieh, Tuan., Tian-Sheuan, Chang., Chein-Wei, Jen.: On the Data Reuse and Memory Bandwidth Analysis for Full-Search Block-Matching VLSI Architecture, Vol. 12. IEEE Transactions on Circuits and Systems for Video Technology (2002)
3. Komarek, T., Pirsch, P.: Array Architectures for Block Matching Algorithms, Vol. 36. IEEE Transactions on Circuits and Systems (1989)
4. Yu-Wen, Huang., Tu-Chih, Wang., Bing-Yu, Hsieh., Liang-Gee, Chen.: Hardware Architecture Design for Variable Block Size Motion Estimation in MPEG-4 AVC/JVT/ ITU-T H.264, Vol. 2. Proc. IEEE International Symposium on Circuits and Systems (2003)
5. Min-ho, Kim., In-gu, Hwang., Soo-Ik, Chae.: A Fast VLSI Architecture for Full-Search Variable Block Size Motion Estimation in MPEG-4 AVC/H.264, Vol. 1. Asia and South Pacific (2005)

6. Yap, S.Y., McCanny, J.V.: A VLSI Architecture for Advanced Video Coding Motion Estimation, Proc. IEEE International Conference on Application-Specific Systems, Architectures, and Processors (2003)
7. Kuhn, P.M., Weisgerber, A., Poppenwimmer, R., Stechele, W.: A Flexible VLSI Architecture for Variable Block Size Segment Matching with Luminance Correction, IEEE International conference on Application-Specific Systems, Architectures, and Processors (1997)
8. Chen, Y.K.. Kung, S.Y.: A systolic design methodology with application to full-search block-matching architectures, VLSI Signal Processing (1998)
9. De Vos, L., Stegherr, M.: Parameterizable VLSI Architectures for the Full-Search Block-Matching Algorithm, Vol. 36. IEEE Transactions on Circuits and Systems (1989)
10. Rahman, C.A., Badawy, W.: A Quarter Pel Full Search Block Motion Estimation Architecture for H.264/AVC, IEEE International Conference on Multimedia and Expo (2005)

A Domain Reduction Algorithm for Incremental Projective Reconstruction

Rafael Lemuz-López and Miguel Arias-Estrada

Instituto Nacional de Astrofísica Óptica y Electrónica
Luis Enrique Erro, 1 Tonantzintla, Puebla, México
{rlemuz, ariasm}@inaoep.mx

Abstract. In this paper we address the problem of recovering the three-dimensional shape of an object and the motion of the camera based on multiple feature correspondences from an image sequence. We present a new incremental projective factorization algorithm using a perspective camera model. The original projective factorization method produces robust results. However, the method can not be applied to real-time applications since it is based on a batch processing pipeline and the size of the data matrix grows with each additional frame. The proposed algorithm obtains an estimate of shape and motion for each additional frame adding a dimension reduction step. A subset of frames is selected analyzing the contribution of frames to the reconstruction quality. The main advantage of the novel algorithm is the reduction of the computational cost while keeping the robustness of the original method. Experiments with synthetic and real images illustrate the accuracy and performance of the new algorithm.

1 Introduction

The problem of shape from motion is to recover the structure of the scene and camera motion at the same time using feature point correspondences from multiple views. In the last years different methods to obtain 3D models from an image sequence have been developed [1, 2, 3, 4, 5, 6].

Typical image sequences for real applications of structure from motion methods are formed of 20-30 frames per second with hundred of feature point correspondences between successive frames. In addition, for certain periods of time the camera motion will be slow, adding negligible information to the reconstruction process but making the estimation of the structure of the scene and camera motion unaffordable. Then, selecting the subset of frames that are more suitable to obtain a reliable projective reconstruction can reduce the amount of information and therefore the processing time.

In this paper a new algorithm for projective reconstruction is proposed based on a factorization method. The original algorithm proposed by Mahamud et al in [1] is extended to achieve incremental projective reconstruction. The main contribution of the new algorithm is the development of an online domain reduction step. Analyzing the contribution of previous frames on the projective reconstruction process a subset of frames is selected. The proposed algorithm has

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 564–575, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the advantage of keeping the robust characteristic of the original algorithm and reducing the computing time with comparable qualitative results to the original iterative factorization method.

2 Related Work

Based on the projective camera model, projective reconstruction methods can be divided into four categories: epipolar geometry, recursive, factorization based methods and robust non linear methods. Epipolar geometry based methods [3, 4, 8] compute the epipolar geometry between successive pairs of images and then merge the partial results to compute the structure and motion of the scene.

Recursive methods have been investigated in [7, 8] based on nonlinear filtering that fuses estimates over time and recently in [15] using a preemptive scoring of the motion hypothesis to compute structure and motion from a calibrated projective camera. On the other hand, in the factorization methods [1, 2, 9, 10], the information of correspondences of all frames is treated simultaneously.

The optimal way to recover motion and structure from images is to use robust non-linear methods like bundle adjustment [11]. However, bundle adjustment does not give a direct solution; it is a refining process and requires a good starting point.

The original factorization method [9] factors a measurement matrix representing the image positions of feature points tracked over multiple frames then, using Singular Value Decomposition (SVD), two matrices are computed, which represent object shape and camera motion respectively. Three important problems that arise with factorization like methods are:

- SVD is a computationally expensive procedure.
- The size of the measurement matrix increases for each new frame processed.
- All correspondence points must be available for factorization.

To overcome these problems, various strategies have considered that the structure of the scene could remain the same over time, i.e., it should be possible to update shape without performing extensive computations. Morita and Kanade [10] improved the original factorization method, and developed a sequential version for the orthographic and para-perspective cases, that has almost the same accuracy but with lower computational cost.

Peter Sturm and Bill Triggs [2] extended the factorization method for shape recovery from projective cameras using estimated fundamental matrices and epipoles from the correspondence data. The algorithm recursively finds the unknown projective depths associated with the full perspective camera model from the epipolar geometry between each view. After that, the SVD factorization method is applied to find the shape and motion matrix. Mahamud et al in [1], proposed a bilinear iterative algorithm by adding new constrains on the error function minimized by the Sturm-Triggs method. In the Mahamud et al method, the initial projective depth values are obtained using the Kanade orthographic method as initial estimation avoiding the need to compute projective depths from the epipolar geometry.

The problem of selecting the subset of frames more suitable to obtain a reliable estimation of structure and motion has been investigated for epipolar based projective reconstruction methods by Repko and Pollefeys in [3,16], counting the number of feature points tracked in successive pairs of images and analyzing the reprojection errors for pairs and triplets of views using epipolar geometry and homographies using the Geometric Robust Information Criterion (GRIC) proposed in [14]. Their keyframe criterion selects two or three views where the GRIC score of the epipolar model is lower than the score of the homography model. A similar idea was presented in [17] where three weighted terms are considered for the keyframe pairing selection.

$$S_g = w_1(1.0 - \frac{n_1}{n_2}) + w2\frac{1}{e_H^2} + w_3 e_F^2 \tag{1}$$

The first term is related to the number of feature points reconstructed in the previous keframe pair n_1 with the features that are also reconstructed in the actual keframe pair n_2. The second term weights the median reprojection error e_H^2 when a planar homography is fitted to the feature data. Finally, the last term includes the median reprojection error e_F^2 when the fundamental matrix is estimated with the feature correspondences. Keyframes are then chosen so that the weighted sum of the three terms is minimized.

3 Projective Factorization

The original projective factorization method described in [1] assumes that n feature points are observed by m perspective cameras. Then, a registered measurement matrix W with dimensions $3m \times n$ is formed, as follows:

$$W = \begin{bmatrix} \lambda_{11}x_{11} & \lambda_{12}x_{12} & \lambda_{1n}x_{1n} \\ \vdots & \vdots & \vdots \\ \lambda_{m1}x_{m1} & \lambda_{m2}x_{m2} & \lambda_{mn}x_{mn} \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_m \end{bmatrix} \begin{bmatrix} X_1 \cdots X_n \end{bmatrix} \tag{2}$$

Where $X_j = (x_j, y_j, z_j, 1)^T, (j = 1, \cdots, n)$ are the unknown homogeneous 3D point vectors, $P_i (i = 1, \cdots, m)$ are the unknown 3×4 image projections matrix associated with camera i, λ_{ij} are scale factors called projective depths and, $x_{ij} = (u_{ij}, v_{ij}, 1)^T$ are the measured homogeneous image point vectors respectively. Using matrix notation the image points are projected from a world point by the following equation:

$$\frac{1}{\lambda_{ij}} P_i X_j \tag{3}$$

Ideally, to solve the structure from motion problem one should minimize the mean -squared distance between the observed image points and the points positions predicted from the parameters λ_{ij}, P_i and X_j, i.e.:

$$E = \Sigma |x_{ij} - \frac{1}{\lambda_{ij}} P_i X_j|^2 \tag{4}$$

However, the corresponding problem is difficult since the error is highly non-linear in the unknowns λ_{ij}, P_i and X_j. Mahamud et al [1] proposed an iterative algorithm to minimize the algebraic error function:

$$E = \Sigma|\lambda_{ij}x_{ij} - P_iX_j|^2 \tag{5}$$

Their iterative algorithm minimizes E by alternating steps, where the motion and structure parameters are estimated from the measurement matrix, with steps where the projective depths are computed from the motion and structure estimates. Initially a rank-4 decomposition determines the best sets of projective depths for factorization. Two matrices are found: P ($3m \times 4$ dimensions) and X ($4 \times n$ dimensions), representing camera motion and object shape respectively. Then, by imposing a constraint on the columns of W to have unit norm, the projective depths are computed. An important property of the Mahamud et al algorithm is that it converges without regarding of the starting point [1].

To summarize the original projective factorization algorithm in a compact notation, two new vectors and one new matrix are defined:

$$\lambda_j = \begin{pmatrix} \lambda_{1j} \cdots \lambda_{mj} \end{pmatrix}^T \; j = (1, \cdots, n) \tag{6}$$

$$\Phi_j = \begin{bmatrix} x_{1j} & 0 & \cdots & 0 \\ 0 & x_{2j} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & x_{1j} \end{bmatrix} \tag{7}$$

$$d_j = \Phi_j\lambda_j = (\lambda_{1j}x_{1j}, \cdots, \lambda_{mj}x_{mj})^T \tag{8}$$

Then, the full original iterative projective factorization algorithm [1] is described as follow:

Algorithm 1. Original projective factorization algorithm

1. Set $\lambda_{ij} = 1$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$;
2. Compute the current scaled measurement matrix W by equation (1);
3. Normalize W; constraining the columns to $|W_j| = 1$, $j = 1, \ldots, n$.
4. Perform the rank-4 factorization on W by SVD, $W = USV^T$, to generate an estimate of projective matrix P and shape matrix X; $P = U_4$ and $X = SV_4^T$ where U_4, S_4 and V_4 are the sub matrices obtained from U, S and V using only the 4 first columns (the ones associated with the 4 largest eigenvalues) and S is a diagonal matrix with elements σ_5 known as the singular values of W.
5. For $j = 1, \ldots, n$ do
   (a) compute $A_j = \Phi_iU_4U_4^T$ and $B_j = \Phi_4\Phi_4^T$;
   (b) solve the generalized eigenvalue problem $A_j\lambda = \sigma_5B_j\lambda$ and set λ_j to be the generalized eigenvector associated with the largest eigenvalue ;
6. If λ_{ij} are approximately the same as the previous iteration, stop; else go to step 2;

In step 3, if W corresponds to a projection of real 3D points then, with the correct depth values, it must be a rank-4 matrix [5]. The fifth singular value σ_5 of the S diagonal matrix represents the distance using the Frobenius norm from the subspace of rank-4. Then, the σ_5 can be used as a measure of this distance and it should have zero limit if the iterative process is known to converge. Due to the effect of noise and misleading matches on the preprocessing stage of tracking algorithms, the σ_5 eigenvalue will not be zero but, after the iterative process of the projective reconstruction algorithm, its value will be approximately the same as in the previous iteration reaching a stable value near zero.

4 Proposed Incremental Projective Reconstruction Algorithm

The proposed algorithm for incremental projective reconstruction is similar to the Mahamud et al [1] approach but it assumes that only a predefined maximum number of r frames can be maintained in W. The proposed method differs from the original one when the $k_{th} \geq r$ frame is processed. A domain reduction stage is added to keep the processing time constant restricting the size of W to allocate only r frames. Furthermore, the initialization step allows the reuse of the pre-computed structure and motion estimations for each added frame. Then, from the $k_{th} \geq r$ frame, the structure and motion of the scene is incrementally computed for each additional frame in nearly constant processing time.

4.1 Domain Reduction by Inter-frame Selection

Our objective is to find the subset of r frames $\{W_r\}$ from W to obtain a projective reconstruction preserving the properties of the original method like reconstruction accuracy and robustness but, at the same time reducing significantly the processing time. In order to obtain this result we propose to add an online domain reduction stage. The row subset selection from W is accomplished imposing a constraint on the rows by using the σ_5 singular value.

Let us assume that whenever an image is added at time t, we keep a record of the last value assigned to $\sigma_5(t)$ after the last iteration of the projective reconstruction method. Intuitively, whenever a new frame is processed, at the end of the iterative projective reconstruction algorithm the $\sigma_5(t+1)$ eigenvalue gives an idea on the contribution of this new frame in improving the reconstruction quality according to the proximity of the $\sigma_5(t+1)$ singular value to the zero limit. When the maximum number of frames r that can be considered on the measurement matrix W has been reached, a decision must be taken about which frames of W will give the best projective reconstruction quality in terms of the approximation to the rank-4 subspace. It would be desirable to estimate in the k_{th} frame which combination of the k previous frames taken r at a time gives the lower σ_5 eigenvalue after computing the projective factorization for each combination of frames:

$$min\ \sigma_5^{C(k,r)} \tag{9}$$

But this test would take long time to be computed since there are $k!/(k-r)!r!$ possible combinations. Therefore, we propose to introduce a simplification. Only one row index of the data matrix (the one corresponding to the frame with less contribution to the reconstruction process) is located since it must be replaced.

Prior to the frame selection the proposed algorithm favors the selection of separated frames. Computing the discrete derivative $\sigma_5'(t)$.

$$\sigma_5'(t) = \sigma_5(t) - \sigma_5(t-1), t = 2, \ldots, k \tag{10}$$

Local minimums are located analyzing sign changes from negative to positive in $\sigma_5'(t)$.

$$localmins = \{t\ frames\} | \sigma_5'(t) > 0 \wedge \sigma_5'(t-1) < 0, t = 2, \ldots, k \tag{11}$$

Then, the σ_5 corresponding to local minimums are re-estimated by:

$$\sigma_5(localmins) = \sigma_5(localmins)\frac{1}{nf} \tag{12}$$

Where nf is the number of frames from the last local minimum.

Finally, the subset or r frames being part of W are selected according to the following expression:

$$\{t_r\} = min\ \sigma_5(t) \cup \{1, k\}, t = 2, \ldots, k-1 \tag{13}$$

Observe that by construction (13) we assure to consider the first an last frame of the image sequence since projective reconstruction benefits from wide separated views and because the interest is to have an estimate of structure and motion for the current frame. The rest of W includes the frames with highest contribution to yield close distance to the rank-4 subspace.

The decision about keeping or replacing rows from W is based on the analysis of σ_5. Thus, if the $\sigma_5(t)$ corresponding to the last frame is smaller than the one associated to the previous frames in W, the corresponding less contributing frame will be deleted from W.

Note that to decide frame deletion it is only needed to keep a record of the maximum σ_5 after r frames have been added in W. Once the frame of smaller contribution has been identified the three rows corresponding to that frame are deleted from W then, the current processed frame can be added in the last three rows of the measurement matrix W.

4.2 Incremental Projective Reconstruction Algorithm

Incorporation the new domain reduction in W described previously, the proposed incremental projective factorization algorithm is as follows:

Algorithm 2. Proposed incremental projective algorithm

1. Set $\lambda_{ij} = \lambda_{i-1j}$, for $i = r, \cdots, m$ and $j = 1, \cdots, n$;
2. compute $\sigma_5'(t) = \sigma_5(t) - \sigma_5(t-1)$;
3. Find the set of localmins in $\sigma_5(t)$ analyzing sign changes in $\sigma_5'(t)$;
4. Set $\sigma_5(localmins) = \sigma_5(localmins)\frac{1}{nf}$;
5. Find the new subset of r frames $\{t_r\} = min\ \sigma_5(t) \cup \{1, k\}, t = 2, \ldots, k-1$
6. Replace rows from W;
7. Do steps 3 to 6 of the original algorithm 1;

5 Experimental Results

Synthetic data was used to compare quantitatively the accuracy of the final 3D reconstruction of the original algorithm and the proposed incremental algorithm using 500 random generated points within a semi sphere of 100 units. Thirty views of these points were taken by a camera looking to the center of the sphere. The camera was located at a distance of 200 units from the origin of the sphere to introduce perspective effects. Figure 1 shows the synthetic data used for the experimental test.

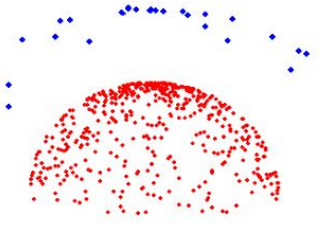


Fig. 1. The synthetic semi sphere points and camera locations used for comparing the original Batch Projective Factorization vs Incremental Projective Factorization methods

5.1 Incremental Projective Reconstruction Accuracy

To compare the original Batch Projective Factorization [1] (BPF) and the proposed Incremental Projective Factorization algorithm (IPF) the following computation are performed. A submatrix W_t is build containing the feature correspondences up to frame t to get the best estimation of the original algorithm. Then, the original algorithm is applied to the submatrix W_t. On the other hand the proposed algorithm is evaluated with a maximum predefined number of frames explicitly shown in the experiments. Several experiments were conducted adding uniform noise with varying standard deviation to both x and y image coordinates. Figure 2 plots the RMS reprojection error resulting of adding zero mean Gaussian noise with standard deviation 2 to simulate tracking drift and image noise. On the left, three and five frames are kept in W when using the

IPF algorithm and on the right a closer view for frames 10 to 30 when seven frames are kept in W. Notice that the RMS reprojection error (1 pixel variation) of projective reconstruction is similar for the original [1] and proposed method. However, the new method needs lower computational time and reduced memory resources.

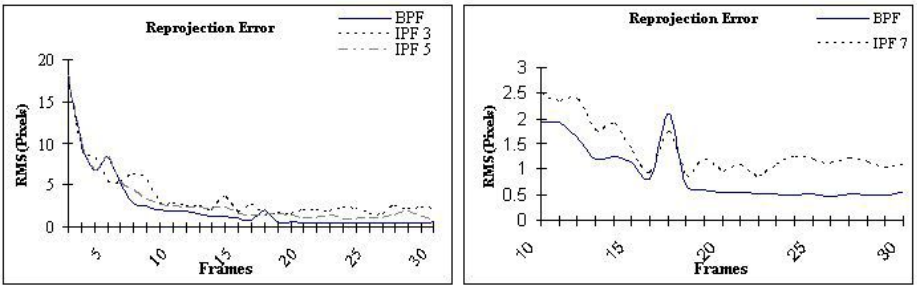


Fig. 2. The RMS reprojection error for the original BPF vs IPF algorithm using 3,5 and 7 frames

Figure 3 shows the variation of the σ_5 singular value during incremental projective reconstruction using the synthetic data of figure 1 when the maximum allowable frames in the measurement matrix are 7. Note that at frame 30 the intermediate frames selected for projective reconstruction are the frames 6, 9, 12, 20 and 24.

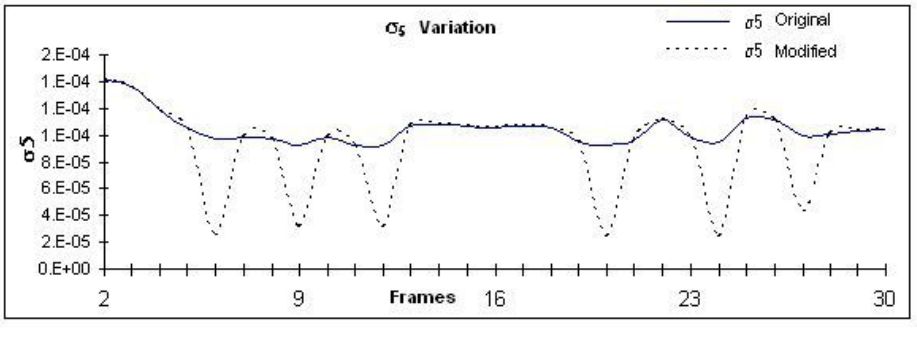


Fig. 3. The variation of the fifth singular value during projective reconstruction using the Incremental Projective Factorization method on frames 2 to 30

5.2 Processing Time

Figure 4 shows the comparison of processing time for the original and proposed method. The structure and motion is computed using 100 random generated

points projected along a sequence of 10 frames. Notice how the proposed incremental reconstruction algorithm requires constant processing time (0.32 ms) when the number of previous frames to be considered on the reconstruction pipeline has been reached (4 in the current example). The processing time was measured using an Intel Pentium IV 1.5 GHz processor with 256 MB RAM.

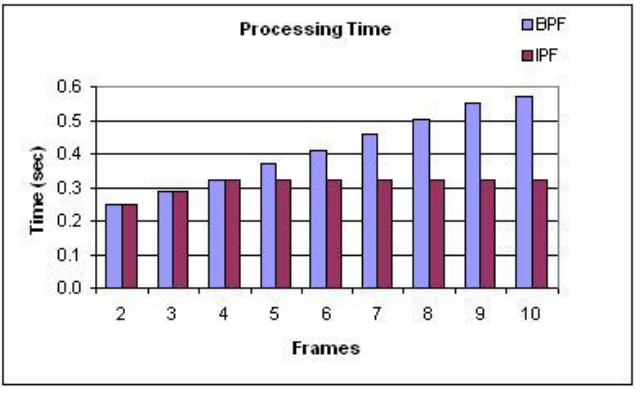


Fig. 4. The processing time for Incremental Projective Factorization Algorithm Vs Batch Projective factorization

The reduction of processing time can appear marginal for the last example but having in mind that real image sequences could have hundreds of frames the advantage of our approach is evident. For example 10 seconds of video is composed of 300 frames then, computing the 300th with a full measurement matrix would take minutes compared to less than a second when the proposed algorithm is used allowing only 10 frames in W during projective reconstruction.

5.3 Real Image Sequence Experiments

In this section we evaluate qualitatively the structure of the scene recovered by the proposed incremental projective factorization method on three real image sequences.

Considering the general scheme proposed in [3,4 5,6], a 3D self-calibrated reconstruction algorithm is developed. In particular, the automatic detection and tracking of points was carried out by the KLT tracker [12]. Then, the incremental projective reconstruction algorithm described in section 4 obtains an initial structure of the scene up to a projective transformation. Then, the quasi-linear self-calibrated reconstruction proposed in [13] was used to update from projective reconstruction to a metric one. Finally, a texture-mapping technique is used to show an interpolated 3D structure of the original scenes. In all cases the texture of the first image is arbitrary used to show realistic models.

Figure 6 shows that the proposed factorization method can efficiently recover the shape and motion from the corresponding image sequences. This is appreciated from the identifiable geometrical components recovered by our method.

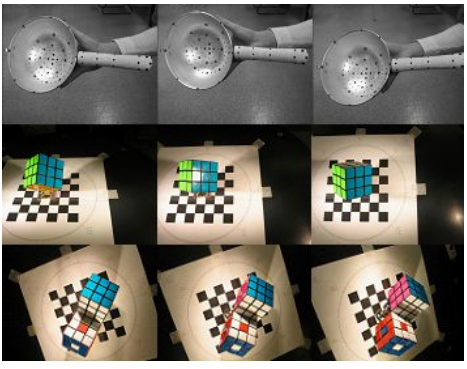


Fig. 5. Three frames of three real video sequences used in our experiments. Top, the pot sequence taken from [16] consisting of 10 frames, 520×390 of image size and 44 feature points. Middle, the cube sequence contains 30 images and 32 salient points. Bottom, two cube sequence 40 images and 52 salient points.

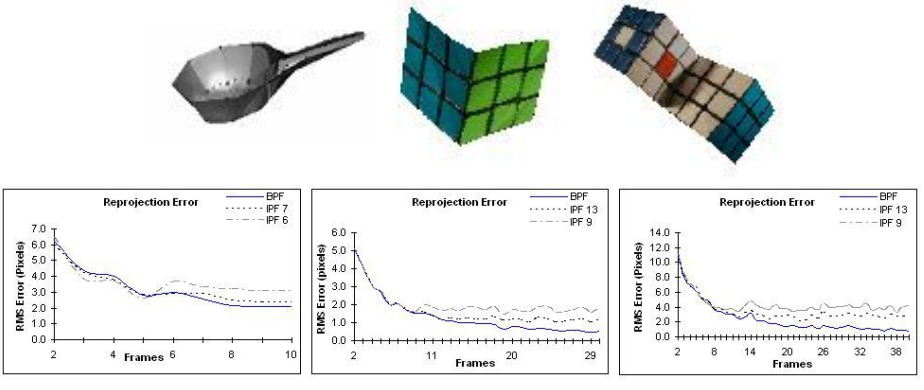


Fig. 6. Top, reconstructed 3D models from the video sequences. Bottom, measured reprojection error for the original BPF and the proposed IPF algorithms. Left: 6 and 7 frames are considered for the IPF algorithm. Right: 9 and 13 frames are automatically considered for the cube sequences using IPF algorithm.

6 Discussion

There are applications where the accuracy of the reconstructed scene can be reduced to some extend and where the most important requirement to be fulfilled is real time processing. Under this situation the original projective factorization algorithm has the inconvenient that, whenever a frame is added the size of the measurement matrix grows and then the processing time quickly becomes prohibitive for real time applications. Then, to recover the structure and motion adding information of a new frame the previous computations can not be used directly, instead the full algorithm can be computed again with a bigger

measurement matrix. Recently an alternative has been presented to align partial results [4] using only the last r frames. On the other hand the proposed online selection criterion automatically determines the frames than should be considered for partial reconstruction by using the fifth eigenvalue as a measure of contribution in the reconstruction quality. An important advantage of the proposed method is that, the size of the measurement matrix is kept constant using a simple but effective domain reduction technique by truncating the size of the measurement matrix considering only the last r frames that most contribute to the projective reconstruction. Notice that keeping the size of the measurement matrix constant, the processing time also remains constant after a latency period when the first r frames have been processed.

7 Conclusions

In this paper an algorithm for incremental projective reconstruction has been proposed. We have shown that adding an online selection criterion to keep or reject frames incremental projective factorization can achieve similar results compared to the original algorithm when a predefined number of frames are allowed to take part in the reconstruction process. An important advantage of the proposed method is that, after a latency period when the maximum allowable frames has been reached the processing time remains constant and similar results are obtained. Experimental results using synthetic and real scenes illustrated the accuracy and performance of the new factorization method.

References

1. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably-convergent iterative methods for projective structure from motion. CVPR **1** (2001) 1018–1025
2. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In Proceedings of European Conference on Computer Vision (Eccv'96) **1065** (1996) 709–720
3. Repko, J., Marc, P.: 3d models from extended uncalibrated video sequences: Addressing key-frame selection and projective drift. 3DIM (2005) 150–157
4. Martinec, D., Pajdla, T.: 3d reconstruction by fitting low-rank matrices with missing data. CVPR **1** (2005) 198–205
5. Hartley, R., Zisserman, A.: Multiple View Geometry in computer Vision. 1 edn. Cambiridge (2000)
6. Grossman, E., Santos-Victor, J.: Algebraic aspects of reconstruction of structured scenes from one or more views. In Proceedings of the BMVC **2** (2001) 633–642
7. Broida, T.J. Chandrashekhar, S., R., C.: Recursive estimation of 3d motion from a monocular image sequence. IEEE Transactions on Aerosp. Electron. Syst **26** (1990) 639–656
8. Soatto, S. Frezza, R., P., P.: Motion estimation via dynamic vision. IEEE Trans. Automat. Contr. **41** (1996) 393–413
9. Tomasi, C., T., K.: Shape and motion from image streams - a factorization method. Int'l J. of Computer Vision **9** (1992) 137–154

10. Morita, T., T., K.: A sequential factorization method for recovering shape and motion from image streams. Proceedings of ARPA Image Understanding Workshop **2** (1994) 1177–1188
11. Triggs, B. McLauchlan, P.H.R., A., F.: Bundle adjustment - a modern synthesis. In Proceedings of the International Workshop on Vision Algorithms: Theory and Practice (1999) 298–372
12. Tomasi, C., Kanade, T.: Detection and tracking of point features. CMU Technical Report (1991) CMU–CS–91–132
13. Pollefeys, M. Reinhard, K., Gool, L.: Self-calibration and metric reconstruction of varying and unknown intrinsic camera parameters. Int'l J. Computer Vision **1** (1999) 7–25
14. Torr, P. Fitzgibbon, A., A., Z.: Maintaining multiple motion model hypotheses through many views to recover matching and structure. ICCV **1** (1998) 485–491
15. Nistér, D.: Preemptive ransac for live structure and motion estimation. ICCV (2003) 199–206
16. M. Pollefeys, L. Van Gool, M.V.K.C.F.V., Tops, J.: Video-to-3d. In Proceedings of Photogrammetric Computer Vision 2002 (ISPRS Commission III Symposium), International Archive of Photogrammetry and Remote Sensing (2002) 247–252
17. S. Gibson, J. Cook, T.H.R.H., Oram, D.: Accurate camera calibration for off-line, video-based augmented reality. ISMAR (2002) 37–46

An Automated Procedure for Word Balloon Placement in Cinema Comics

Bong-Kyung Chun[1], Dong-Sung Ryu[1], Won-Il Hwang[3], and Hwan-Gue Cho[1,2]

[1] Dept. of Computer Engineering, Pusan National University, South Korea
[2] Research Institute of Computer Information and Communication, South Korea
[3] DIOTEK Co., Ltd., South Korea
{bkchun, dsryu}@pearl.cs.pusan.ac.kr, wihwang@diotek.co.kr, hgcho@pusan.ac.kr

Abstract. Comics has been studied for a long time, as they are the oldest genre of non-photorealistic rendering. Especially, it is a very interesting work to transform a video into a comics in an automatic manner. As a kind of sub-work for this, we introduce a method to automatically place word balloons in an image in this paper. It is not trivial to place the time-serial word balloons with keeping the rules of 2D comics text layout. Our algorithm first positions each word balloon relative to its respective actor while maintaining the reading order of dialogues by applying a cartographical labeling algorithm. The position of each balloon is then refined on the basis of our proposed measure that estimates the quality of balloon layout. We experiment with several commercial cinemas, which shows that the final results are good enough to compare with hand-drawn comics.

1 Introduction

Comics has been studied for a long time, as they are the oldest genre of non-photorealistic rendering. Two famous books, *"Understanding Comics"* and *"Reinventing Comics"* have successfully explained the basic structure of traditional comics and the power of comics as a communication tool [1,2].

It is especially a very interesting work to transform a video (generally a cinematic film) into a comic book in an automatic manner. It consists of several kinds of sub-work, i.e., cartooning an image, rendering the stylized effects for the moving objects and placing word balloons. Currently, there has been lots of work for these comics' own effects except the work of word balloons.

Word balloon is one of the most distinctive and readily recognizable elements of the comic medium. The location of each word balloon is very crucial to convey the contents of a video stream since the word balloon must express the main dialogue of a video stream by a textual form. So, it is not trivial to place the time-serial dialogues into word balloons with keeping the traditional rules of 2D comics text layout scheme.

In this paper, we propose a new procedure for an automated word balloon placement in an image. Our procedure consists of two steps. The first step is responsible for positioning each word balloon relative to its actor while maintaining the reading order of dialogue texts by applying a cartographical labeling algorithm [3,4]. The second step refines the location of each balloon on the basis of our measure function, which estimates the quality of word balloon layout, to show more natural or hand-drawn styles.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 576–585, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Work

Researchers in non-photorealistic rendering have noticed the interesting features of comics, and they have examined lots of work related to comics. For example, they have studied the methodology for cartooning an image and adding the comics' own stylized effects for expressing the movement of objects.

There is lots of work for stylization and abstraction of an image [5,6,7]. It converts an input image into a highly abstracted, spatial-temporally coherent image. These cartoon-like stylized images are used for comics or animation, since they are typically composed of large regions which are semantically meaningful and highly abstracted.

Some researchers also have studied the comics' own stylized effects for manifesting the motion of objects in a scene. For example, Hanl *et al* [8] studied how to deform objects using a squash-and-stretch technique and add speedlines to objects for rendering the cartoon-like motion. Others have intensively studied the depiction mechanism to show the dynamics of moving objects using principles of visual art and narration [9,10].

But there is no work similar to our work which totally focuses on the automatic layout of word balloons in a scene. Kurlander *et al* developed the internet chatting system *Comic Chat* with stylized comic effects [11]. They have mentioned the problem of word balloon layout. But it not only has the free-reserved area for word balloons but also overlooked the relationship between objects and balloons. They just considered the reading order of word balloons and solved it using a simple greedy methodology.

There are also lots of systems for generating comics such as *Comic Book Creator* [12]. But these systems don't offer some kinds of automatic functions. Totally manual work and the pre-made materials such as clip arts are required.

3 Main Problem

There is no absolute standard to decide what kind of word balloon layout is good because it is dependent upon human perception. But we have known that there are general rules which comic writers have followed for good layout of balloons for a long time. In this section, we describe the rules and define the main problem to solve.



(a) An original image (b) o_i and r_i in an image

Fig. 1. An example for the problem of word balloon layout

We first start with some definitions to constitute a scene (Figure 1). Let b_i be a word balloon, and r_i be an reference area of b_i. Typically, r_i can be the area of an actor's face. There are many automatic methods to detect an actor's face for r_i, but we assume that

r_i is given by a user in this paper. In general, one b_i is assigned for each r_i, but some special cases may allow more than one word balloon for each r_i. In addition, speech dialogue d_j will be included in each b_i. Note that the sequence of dialogues, $\{\ d_j\ \}$ is the time-serial object. That means d_j is the next utterance of d_{j-1}; for example, d_2 is the next utterance of d_1.

Comic writers tacitly have kept main rules to make readers understand contents exactly and better, when they place word balloons in a comic cut. Our balloon layout also follows these, and the three main rules are as follows:

1. $b_i \cap b_j = \varnothing$: Balloons should be disjointed from each other.
2. $b_i \cap r_j = \varnothing$: Every balloon b_i should not overlap every r_j.
3. b_i must be placed according to its $RO(b_i)$, to keep the reading order of dialogues.

Suppose that we have a set of balloons, $\{b_i\}$ and b_i is represented by three tuples as $b_i = \langle c_i(x_i, y_i), v_i, h_i \rangle$, where $c_i(x_i, y_i)$ is the center coordinates of b_i, and v_i and h_i are the length of vertical and horizontal axes of b_i . We define $RO(b_i)$, the Reading Order of $\{b_i\}$. Let $Zone(b_i)$ be a horizontal strip (rectangle) which encloses b_i at a minimum. If c_i is not in $Zone(b_j)$ and y_i of c_i is higher than y_j of c_j, we set $RO(b_i) < RO(b_j)$. If c_i is in $Zone(b_j)$, then $RO(b_i)$ is determined by comparing x_i and x_j. If $x_i < x_j$, then $RO(b_i) < RO(b_j)$. Figure 2 shows an example of balloon layout and its $RO(b_i)$ number.

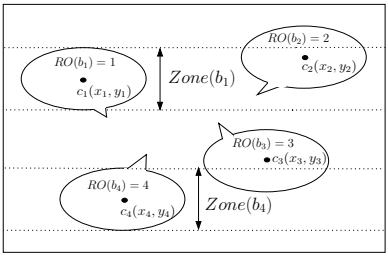


Fig. 2. An example of the Reading Order of a word balloon, $RO(b_i)$, in comics

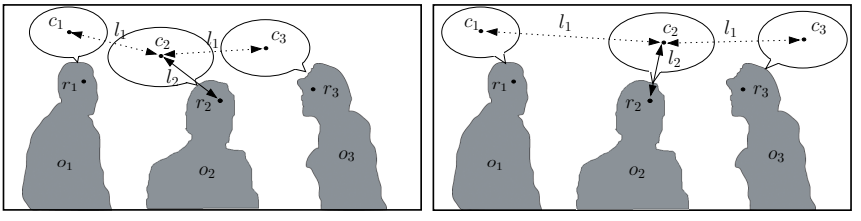
Next, we propose a measure function to evaluate the layout of word balloons. There are two additional rules for good balloon placement.

1. **Balloon-reference area relation:** b_i should be as closely placed to r_i as possible.
2. **Balloon-balloon relation:** b_i should be as far away from other b_js as possible.

Based on these, we define a measure function for balloon layout, $Qlayout(\{b_i\}) = k_1 \cdot L_1 - k_2 \cdot L_2$. L_1 is the sum of distances between all pairs of b_i which were assigned to different r_j. L_2 is the sum of the distances between b_i and r_i. k_1 and k_2 are the coefficients to control the balance between L_1 and L_2. Figure 3 shows two different results of balloon layout and its corresponding $Qlayout(\{b_i\})$ value. We can see that Figure 3 (b) is better than (a).

The final problem is how to get the optimal (or near-optimal) balloon layout. We call this problem the Optimal Balloon Layout (OBL) problem. OBL can formally be defined as follows:

Input: $\{r_i\}$, $\{d_j\}$: the user-given reference areas and the sequence of dialogues.
Output: b_i=$\langle c_i(x_i, y_i), v_i, h_i\rangle$, which maximizes $Qlayout(\{b_i\})$ satisfying $RO(b_i)$.



(a) $Qlayout(\{b_i\}) = 753.4$ (b) $Qlayout(\{b_i\}) = 1358.51$

Fig. 3. An example that has two different layout results, $Qlayout(\{b_i\})$, in the same cut. We notice that (b) is better than (a).

4 Automatic Balloon Placement Procedure

Point-feature label placement (PFLP) is the problem of placing text labels adjacent to point features on a map or diagram so as to maximize legibility. Given K point features, each point feature generally has a set of eight possible label positions as shown in Figure 4. Let the number of each potential positions in Figure 4 be the relative quality of that labeling. The PFLP can be thought of the optimization problem of finding a set of a potential position for each point feature to maximize/minimize the sum of its relative quality. The selected potential positions should not be overlapped one another.

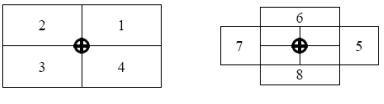


Fig. 4. A typical set of eight possible label positions for a point feature [3]

In this paper, we try to solve word balloon placement (WBP). The WBP also can be thought of a kind of the PFLP problem. As shown in Figure 5, r_k and its potential balloon positions can be consider as a point feature and its potential label positions, respectively. Based on this fact, the WBP problem is defined as follows:

- Given K reference points (r_k) and N_k possible positions for each r_k, each potential balloon position is represented by a variable $P_{k,i}$, $1 \leq k \leq K$, and $1 \leq i \leq N_k$.
- Each $P_{k,i}$ has value 0 or 1, indicating the absence or presence, respectively, of a balloon (b_k) in that position.

– One set of constraints expresses the requirement that a balloon should be placed at each r_k exactly once:

$$\sum_{i=1}^{N_k} P_{k,i} = 1, \quad \text{for } 1 \leq k \leq K$$

– Given Q pairwise overlaps between possible balloon positions, a second set of constraints expresses the requirement that no two balloons overlap:
$P_{r_q,s_q} + P_{r'_q,s'_q} \leq 1$ for each potential overlap, $1 \leq q \leq Q$.
– The objective function is to maximize $Qlayout(\{b_i\})$ value. The bigger the objective function is, the better the layout result is.

Figure 5 (a) shows an input image and r_k. The step of constructing the $P_{k,i}$ for each r_k is shown in Figure 5 (b), and the final result of the WBP to maximize the $Qlayout(\{b_k\})$ value is shown in Figure 5 (c).

In this example, we assume that the balloon of r_k is b_k, and the $RO(b_k)$ is same as k; that means $RO(b_k) < RO(b_{k+1})$. Any algorithm for the PFLP cannot solve the WBP problem while satisfying the reading order constraint if the position of r_ks are not given according to $RO(b_k)$s as shown in Figure 5. However, most scenes from a pre-made movie do not keep the $RO(b_k)$. So, we need to modify the PFLP algorithm to suit the word balloon placement problem.

In this paper, we propose a new heuristic algorithm for word balloon placement based on an extended point-feature labeling placement. For this, we define the extended potential balloon area ($EP_{k,i}$) as shown in Figure 7 (a) because the reading order is the most important constraint in the WBP unlike the PELP. $EP_{k,i}$ allows to place b_k while keeping the $RO(b_k)$. b_k is not fixed at one position, but it can be located along the $EP_{k,i}$ area. Each box in Figure 7 (a) represents $EP_{k,i}$.

Our method consists of two steps. The first step is responsible for positioning each b_k relative to its r_k while maintaining the $RO(b_k)$ by applying a cartographical labeling algorithm. The second step refines the location of each b_k on the basis of $Qlayout(\{b_k\})$ that estimates the quality of layout to show more natural or hand-drawn styles.

Our heuristic procedure works as follows: First, we construct $EP_{k,i}$ for each given r_k, and then place the b_k of each r_k in the order of $RO(b_k)$. We said that the balloon of r_k is b_k, and $RO(b_k)$ is same as k. So, we start to determine the position of the balloon of r_1. We do not consider the reading order when determine $EP_{1,i}$ of the r_1 because it has the lowest $RO(b_1)$. After determining $EP_{1,i}$, we place b_1 as near r_1 as possible like Figure 7 (b). The dashed boxes in Figure 7 (b) denote b_k.

We should consider the reading order of b_k when determining b_{k+1}. b_k has its own bounded zone $BZone(b_k)$ where the next balloon b_{k+1} must be placed for satisfying the reading order (see Figure 6). In Figure 7 (b), the dashed line indicates $BZone(b_1)$. Therefore, b_2 should be placed below the dashed line. When determining the position of b_2, we first divide $EP_{2,i}$ into active $EP_{2,i}$ and non-active $EP_{2,i}$ based on $BZone(b_1)$. Among active $EP_{2,i}$, we place b_2 as near r_2 as possible and as far b_1 as possible. This step is shown in Figure 7 (b) and the shadowed boxes represent non-active $EP_{2,i}$. We repeat this procedure for all b_k. Figure 7 (c) shows the result of first step of our method.

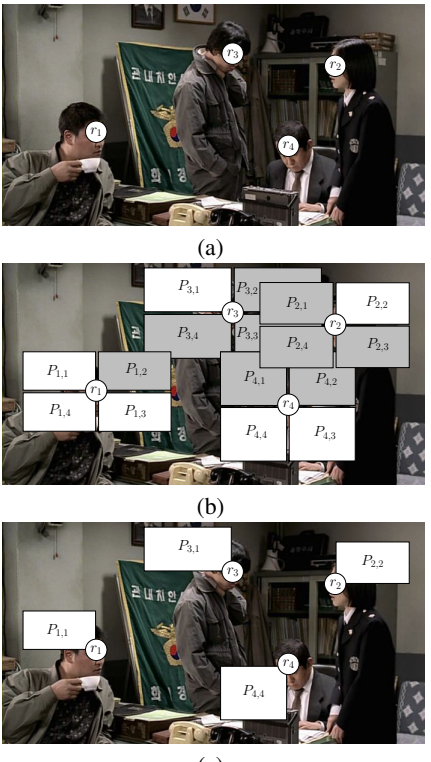


Fig. 5. The process step of word balloon placement by using a simple labeling algorithm. (a) Set r_k for each object. (b) Construct the potential position of balloons $P_{k,i}$ for each r_k. (c) The final result of word balloon layout.

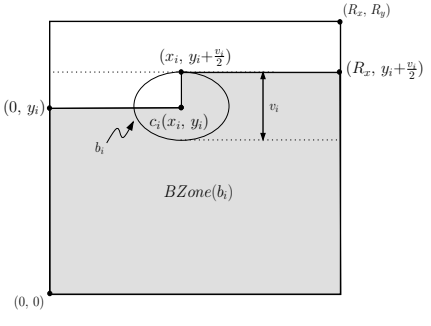


Fig. 6. Bounded Zone $BZone(b_i)$, where b_{i+1} must be placed after b_i

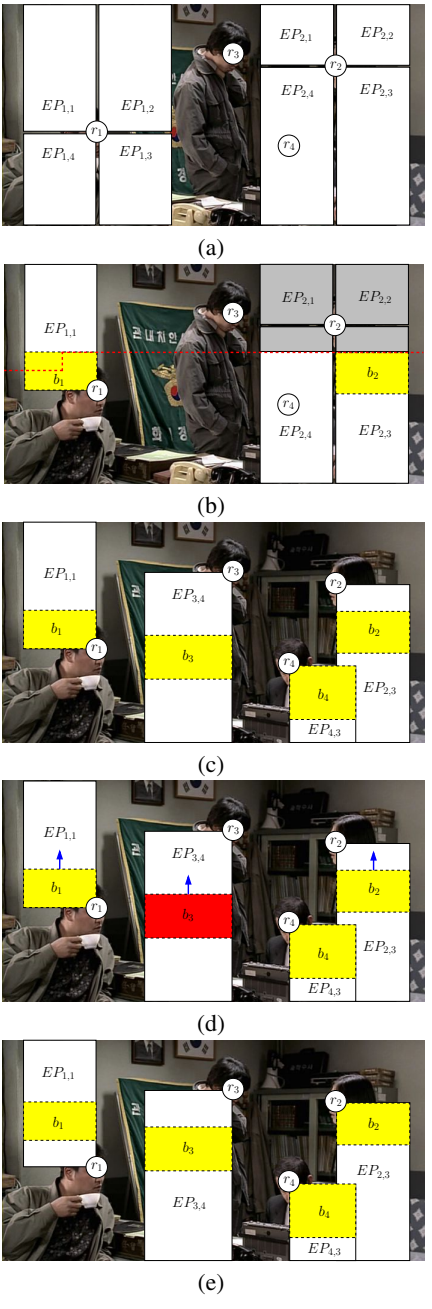


Fig. 7. The process step of our method. (a) Construct the expanded potential balloon area $EP_{k,i}$. (b) Place b_k within $EP_{k,i}$. (c) The result of the first step of our method. (d) The process of refining b_k based on $Qlayout(\{b_k\})$. (e) The final result of our method.

We notice that the distance between r_k and b_k is more important metric that influences on the quality of word balloon layout than the distance between balloons. Although both $\sum d(b_k, r_k)$ in Figure 8 (a) and $\sum d(b_k, r_k)$ in Figure 8 (b) are same, the layout result of Figure 8 (b) is better than (a). The longer $d(b_k, r_k)$ is, the more the quality of the layout is influenced. It can be expressed as the exponential form; $d(b_k, r_k)^2$.

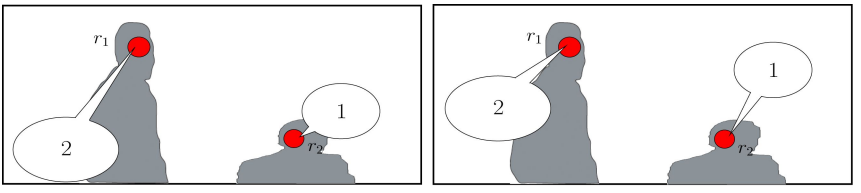


Fig. 8. The importance of $d(b_k, r_k)$. (b) is better than (a) in spite of the same $\sum d(b_k, r_k)$.

Algorithm 1. Word Balloon Placement
Input: (1) $\{b_k\}$; word balloons
(2) $\{r_k\}$; a set of reference areas
(3) K; the number of r_k
Output: one feasible placement for $\{b_i\}$

BZone(b_0) = Rectangle(R_x, R_y);
Construct $EP_{k,i}$ for each r_k;
FOR (k=1; k$\leq$ K; k++)
  FOR (i=1; i$\leq$ N_k; i++)
    IF $EP_{k,i} \in BZone(b_{i-1})$ **THEN** break;
  END FOR
  Determine the position of b_k within $EP_{k,i}$ as near r_k as possible;
END FOR
DO
  PreQlayout = $Qlayout(\{b_k\})$;
  Find the b_m with $max(\{d(b_k, r_k)\})$;
  Move the b_m to its r_m by ΔD;
  Refine other b_ks for keeping $RO(b_k)$;
WHILE ($Qlayout(\{b_k\}) >$ PreQlayout)

In the first step of our method (until Figure 7 (c)), we determined the position of all b_k while satisfying the $RO(b_k)$. It might be said that the work for word balloon placement is done. However, the quality of word balloon layout usually might be not good due to the reading order constraint. Therefore, we need to perform the procedure of refining the location of each b_k on the basis of $Qlayout(\{b_k\})$ to show more natural or hand-drawn styles.

$d(b_k, r_k)$ should be refined for the better value of the $Qlayout(\{b_k\})$. We find the b_m with $max(\{d(b_k, r_k)\})$ and move b_m to its r_m by ΔD. Other b_ks that have lower $RO(b_k)$ than $RO(b_m)$ also should be moved to r_k for maintaining the reading order. We repeat this procedure until maximizing $QLayout(\{b_k\})$. Figure 7 (d) shows this procedure. The b_3 denotes the b_m with $max(\{d(b_k, r_k)\})$ and the arrows represent the moving offset ΔD. The final layout result of our method is shown in Figure 7 (e).

The pseudo code of our heuristic algorithm for word balloon placement is given in Algorithm 1. If the number of balloon of r_k is more than one, we can solve the problem as just generating the duplicated reference areas for other assigned balloons.



(a) An example from *Comic Chat* [11] (b) An example generated by our algorithm

Fig. 9. Comparison with *Comic Chat* and our algorithm

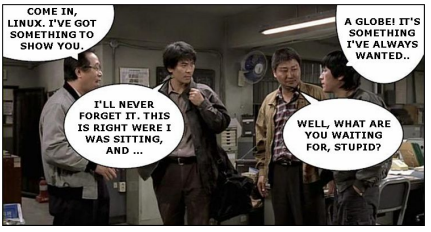


(a) A result before the procedure of refining b_k (b) A result after the procedure of refining b_k

Fig. 10. Results before/after the procedure of refining the position of b_k. By applying this procedure, we can get the better layout result.

5 Experiment Results

The layout results of *Comic Chat* [11], and our algorithm are shown in Figure 9. The balloons in Figure 9 (a) keep the reading order well. But it is difficult to read because the distance between each balloon and its owner is so far. The reason is that the relationship between each balloon and its owner is not considered when positioning balloons. Figure 9 (b) shows an example generated by our work. Our method considers not only the reading order but also the relationship between each balloon and its owner.



(a) A result from the movie *Memories of Murder*



(b) A result from the movie *Before Sunrise*



(c) A result from the movie Mr. Hitch



(d) A result from the movie *Mr. Hitch*

Fig. 11. Word balloon layout results by using our method

Figure 10 shows the results before/after performing the second procedure of our method which is responsible for refining the position of b_k to maximize $Qlayout(\{b_k\})$. We notice that (b) is better than (a). By performing this procedure, we can get the better layout result as most b_ks move nearer r_k.

At last, we present several results of the balloon layout in Figure 11. These images are from famous movies such as *Memories of Murder* and *Mr. Hitch*. We can see that our method generates the good layout results considering the reading order and $Qlayout(\{b_k\})$. All results are good enough to compare with hand-drawn comics.

6 Conclusion

In this paper, we have proposed an automated method for word balloon layout on the basis of an extended point-feature label placement algorithm. For this, we first defined several rules for the good word balloon layout, and then formulated an optimization word balloon layout problem. Compared with previous works, our method generates better layout results because of considering not only the reading order but also the relationship between each balloon and its owner.

Our proposed method has some limitations and drawbacks. First, we assumed that the reference areas of word balloons are defined by users. Second, it is a very important problem of determining the optimal size and shape of word balloons. However, we just assume that balloons are generated from the pre-defined balloon scheme in this work. Therefore, we need to develop a more automated method considering these problems.

We will announce a more automated version of CORVIS in near future, and also publish some comic books from several kinds of famous cinemas.

Acknowledgement

This work is supported in part by the Foundation of Ministry of Information Communication of Korea (under project no.A1100-0602-0055) in 2006. The authors would like to thank Dr. Koo in ETRI for his consistent support to the project.

References

1. McCloud, S.: Understanding comics: The invisible art. Kitchen Sink Press (1999)
2. McCloud, S.: Reinventing comics: How imagination and technology are revolutionizing an art form. Perennial (2000)
3. Christensen, J., Marks, J., Shieber, S.: An empirical study of algorithms for point-feature label placement. ACM Transactions on Graphics **14** (1995) 203–232
4. Edmondson, S., Christensen, J., Marks, J., Shieber, S.: A general cartographic labeling algorithm. Cartographica **33** (1997) 13–23
5. Haeberli, P.: Paint by numbers: abstract image representations. In: Proceedings of SIGGRAPH 1990. (1990) 207–214
6. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. ACM Transactions on Graphics **21** (2002) 769–776
7. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Transactions on Graphics **23** (2004) 574–583
8. Hanl, C., Haller, M., Diephuis, J.: Non-photorealistic rendering techniques for motion in computer games. Computers in Entertainment **2** (2004) 11
9. Masuch, M., Schlechtweg, S., Schulz, R.: Speedlines: Depicting motion in motionless pictures. In: Proceedings of SIGGRAPH 1999. Computer Graphics Proceedings, Annual Conference Series, ACM, ACM Press / ACM SIGGRAPH (1999) 277
10. Hwang, W.I., Chun, B.K., Ryu, D.S., Cho, H.G.: Cinema comics: Cartoon generation from video stream. In: Proceedings of GRAPP 2006. (2006) 299–304
11. Kurlander, D., Skelly, T., Salesin, D.: Comic chat. In: Proceedings of SIGGRAPH 1996, ACM Press (1996) 225–236
12. Games, P.: Comic book creator (2005)

Segmentation of Three Dimensional Cell Culture Models from a Single Focal Plane*

Hang Chang[1,2] and Bahram Parvin[1]

[1] Lawrence Berkeley National Laboratory, Berkeley, CA 94720
[2] Institute of Automation, Chinese Academy of Sciences, Beijing, China

Abstract. Three dimensional cell culture models offer new opportunities for development of computational techniques for segmentation and localization. These assays have a unique signature of a clump of cells that correspond to a functioning colony. Often the nuclear compartment is labeled and then imaged with fluorescent microscopy to provide context for protein localization. These colonies are first delineated from background using the level set method. Within each colony, nuclear regions are then bounded by their center of mass through radial voting, and a local neighborhood for each nucleus is established through Voronoi tessellation. Finally, the level set method is applied again within each Voronoi region to delineate the nuclear compartment. The paper concludes with the application of the proposed method to a dataset of experimental data demonstrating a stable solution when iterative radial voting and level set methods are used synergistically.

1 Introduction

Current protocol for most biological imaging assays is limited to monolayer cell culture models; however, in the body, cells exist in more complex three-dimensional arrangements, in intimate association with each other and components of their microenvironment. These arrangements are critical to the function and maintenance of the differentiated state. The primary rationale for extending a subset of these protocols to 3D cell culture models is that they provide much more faithful replicates of cell behavior *in vivo* than is possible using 2D substrata. While the information these cultures can provide is undoubtedly more valuable, the experiments are much harder to set up, and require more advanced quantitative tools for phenotypic characterization. Furthermore, efficient and robust computational requirements for these experiments have been a rate limiting issue due to a more complex phenotypic signature. A sample of

* The Research was supported by National Aeronautics and Space Administration Grant no. T6275W, NASA Specialized Center for Research in Radiation Health Effects, the low dose radiation research program and the medical imaging program, Office of Biological Effects Research, U.S. Department of Energy, Grant No. DE-FG03-01ER63240. PubID is LBNL-61401.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 586–595, 2006.
© Springer-Verlag Berlin Heidelberg 2006

these 3D colonies, at one focal plane, are shown in Figure 1. For a certain class of studies, three dimensional cell culture models fill a gap between monolayer and *in vivo* models [1]. Although the former may be appropriate as an initial step toward discovery and certain aspects of biological studies, the later is more expensive and time-consuming, and as a result cannot scale for high-throughput studies targeting different end points. However, 3D assays introduce significant computational challenges: (i) subcellular compartments often overlap, (ii) staining/labeling may not be uniform, and (iii) that the scale of labeled compartment may have a large variation.

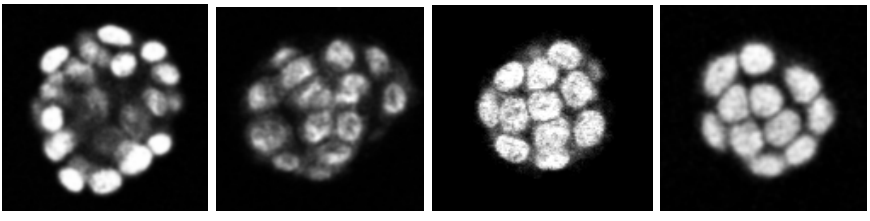


Fig. 1. A slice of a three dimensional cell culture assay indicating variation in size and intensity and overlapping compartments

The first step in most biological imaging assays is to label a subcellular compartment (e.g., nuclear) and to provide context for quantifying protein localization. Labeling the subcellular compartment corresponds to segmentation. Research in the segmentation of subcellular structures spans from traditional ad hoc methods of modeling intensity distribution to geometric techniques [2,3] and surface evolution methods. A key observation is that nuclear regions are often convex and form positive curvature maxima when they overlap each other. This feature was used earlier in 2D segmentation of nuclear regions [3]. However, this method is only applicable to monolayer cell model systems where background is clearly delineated. Our approach is based on constraining the solution to provide seeds corresponding to the nuclear regions and then breaking up local regions based on additional intensity and geometric constraints. The seeding is based on radial voting, where gradient information is projected inward to infer a local center of mass. Once these seeds are established, Voronoi tessellation provides the local neighborhood where each nucleus resides. This local neighborhood is further partitioned based on its intensity distribution using the level set methods.

Organization of the paper is as follows. Section 2 summarized previous methods. Section 3 provides a detailed description of the proposed method. Section 4 provides examples and results on the application of the proposed methods on real data. Section 5 concludes the paper.

2 Previous Research

The difficulties in localization of subcellular compartments are often due to variations in scale, noise, and topology. Other complexities originate from missing data and perceptual boundaries between neighboring compartments. Previous methods for extraction of nuclear compartments have relied on either global thresholding or adaptive (localized) thresholding followed by the watershed method for separating adjacent regions. Other methods have relied on model-based techniques leveraging inherent geometric properties for subcellular compartments [2, 3]. These methods rely on the notion that the nuclear regions are often convex and, when adjacent compartments overlap, they form folds corresponding to curvature maxima, which can be used for grouping and partitioning. Still other techniques have used level set implementation of the active contour models for segmentation and delineation [4].

While geometric methods offer robust model-based constraints for bounding the solution to the segmentation problem, level set methods offer numerically stable solutions to the active contour problems. By limiting and bounding the active contour model to a small nuclear region and its immediate background, a more stable solution is produced, which ensures that (1) overlapping compartments are not merged together, (2) potential leakage in the curve evolution is localized to a small neighborhood, and (3) an improved segmentation can be generated as a result of localized statistics of the region bounding one nucleus.

3 Approach

Specific steps in delineation of nuclei in a mammosphere system are shown in Figure 2. With the 2D image as an input, the colonies are first segmented by applying the active contour model. Then, within each colony, the location of the centroid of each nucleus is estimated through iterative radial voting. With the estimated seeds for nuclei, each nucleus is then bounded in a voronoi tessellation, which is further refined by active contour evolution.

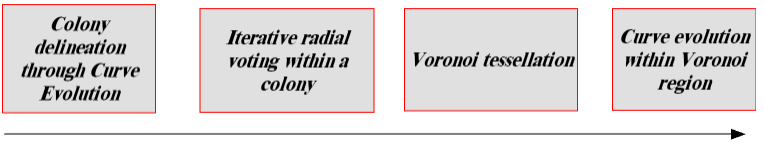


Fig. 2. Steps in delineation of nuclear regions in a mammosphere structure

3.1 Colony Segmentation with the Active Contour Model

The active contour model has been used extensively in the medical and biological imaging communities, and many variations of it are currently supported through

ITK from Kitware, Inc. The level set formulation of the active contour model offers a natural and numerically robust implementation of curve evolution equations within the Eulerian formulation. The ITK image library offers several 2D and 3D implementations of curve evolution within the level set framework. The most basic model includes a constant external motion coupled with geometric smoothing. This model essentially behaves like a region-growing method, subject to the continuity of evolving contours while simultaneously handling topological changes. Another model within the ITK library, known as geodesic active contours [5], aims to unify parametric and level set geometric models. This approach formulates the level set concept within an optimization framework. In addition, the ITK library includes a number of level set formulations of active contours that incorporate prior shape models, which can facilitate segmentation of known patterns. Our implementation corresponds to the Mumford-Shah function as modeled by Chan and Vese [6] to segment objects whose boundaries are not necessarily defined by a gradient, thus reducing the leakage experienced by some gradient-based curve evolution functions. The deriving energy functions are governed by forces that are computed from the interior and exterior of evolving boundaries. A brief overview follows. The energy functional $F(c_1, c_2, C)$ is defined by

$$\begin{aligned} F(c_1, c_2, C) = &\mu \cdot Length(C) \\ &+ v \cdot Area(inside(C)) \\ &+ \lambda_1 \textstyle\int_{inside(C)} |u_0(x,y) - c_1|^2 dxdy \\ &+ \lambda_2 \textstyle\int_{outside(C)} |u_0(x,y) - c_2|^2 dxdy \end{aligned} \tag{1}$$

where u_0 corresponds to the image, c_1 and c_2 are the foreground and background intensity, and $\mu \geq 0$, $v \geq 0$, $\lambda_1, \lambda_2 \geq 0$ are fixed parameters. The level set formulation of this model is given by considering $C \subset \Omega$ as the zero level set of a Lipschitz function $\phi : \Omega \to \mathbb{R}$, in which Ω is a bounded open subset of $\mathbb{R}^2$. Using the Heaviside function H, and the one-dimensional Dirac measure δ_0, defined by

$$H(z) = \begin{cases} 1, \text{if } z \geq 0 \\ 0, \text{if } z < 0 \end{cases} \delta_0(z) = \frac{dH(z)}{dz}$$

The energy form can now be written as

$$\begin{aligned} F_\epsilon(c_1, c_2, \phi) = &\mu \textstyle\int \delta_\epsilon(\phi(x,y))| \bigtriangledown \phi(x,y)| dxdy \\ &+ v \textstyle\int H_\epsilon(\phi(x,y)) dxdy \\ &+ \lambda_1 \textstyle\int |u_0(x,y) - c_1|^2 H_\epsilon(\phi(x,y)) dxdy \\ &+ \lambda_2 \textstyle\int |u_0(x,y) - c_2|^2 (1 - H_\epsilon(\phi(x,y))) dxdy. \end{aligned} \tag{2}$$

If ϕ is fixed then c_1 and c_2 can be expressed as:

$$c_1(\phi) = \frac{\int_\Omega u_0(x,y)H_\epsilon(\phi(x,y))dxdy}{\int_\Omega H_\epsilon(\phi(x,y))dxdy}$$
$$c_2(\phi) = \frac{\int_\Omega u_0(x,y)(1-H_\epsilon(\phi(x,y)))dxdy}{\int_\Omega (1-H_\epsilon(\phi(x,y)))dxdy} \quad (3)$$

And by keeping c_1 and c_2 fixed, the Euler-Lagrange equation for ϕ can be written as

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon \left[\mu div\left(\frac{\bigtriangledown\phi}{|\bigtriangledown\phi|}\right) - v - \lambda_1(u_0-c_1)^2 + \lambda_2(u_0-c_2)^2\right] \quad (4)$$

In our implementation, we have used the regularization of H, introduced by [6]:

$$H_{2,\epsilon}(z) = \frac{1}{2}\left(1+\frac{2}{\pi}\arctan\left(\frac{z}{\epsilon}\right)\right) \quad (5)$$

While in some cases thresholding may be sufficient for colony segmentation, we have opted to use the active contour model since it is fast (about 1.8 seconds for an image of 512-by-512 with 5 colonies), avoids ad hoc parameter setting, and will be reused for nuclear segmentation. Examples are included in a later section.

3.2 Detection of the Nuclear Organelle with Iterative Scalar Voting

Nuclear regions are convex and often radially symmetric, as shown in Figure 1. Complexities arise as a result of nonuniform staining, overlapping nuclei, and variations in size. The underlying theme is to model the nuclear detection through perceptual grouping; thus making the detection more invariant to noise and variation in size. Voting along gradient direction provides a hypothesis profile for saliency, i.e. an approximate center of mass. The voting kernel is designed in such a way that encodes for a specific saliency and refined at each iteration step. The shape and evolution of these kernels, inferring center of mass, is shown in Figure 3. Detection of radial symmetry is iterative, where gradient magnitude is projected along the radial direction according to a kernel function. The kernel function is smooth and its topography becomes more focused and dense at each consecutive iteration, as shown in Figure 3 and the details of the algorithm can

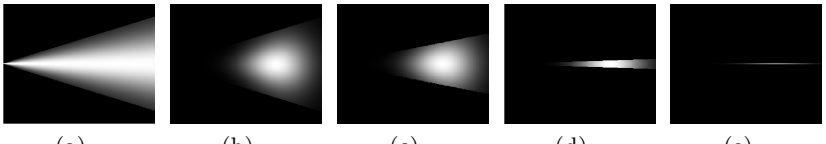


Fig. 3. Kernel topography: (a-e) Evolving kernel for the detection of radial symmetries (shown at a fixed orientation) has a trapezoidal active area with Gaussian distribution along both axes

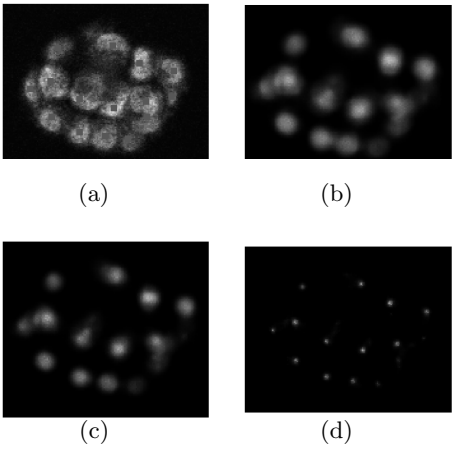


Fig. 4. Detection of radial symmetries for a mammosphere with multiple overlapping nuclei: (a) original image; and (b-d) voting landscape at several intermediate steps indicating convergence to a localized region.

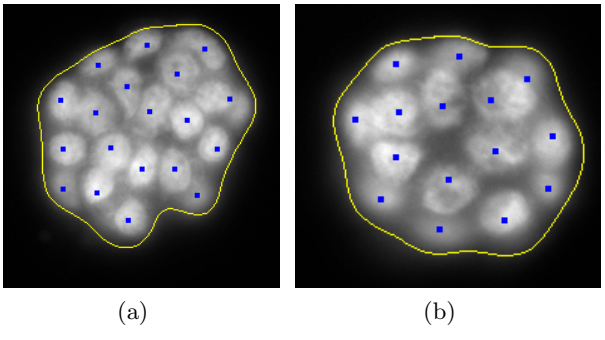


Fig. 5. Examples of voted nuclei from two mammospheres indicates robust performance in the presence of variation in size and contrast

be found in earlier papers. To illustrate the behavior of iterative voting, Figure 4 shows intermediate steps that lead toward final results for overlapping 2D objects that are generated synthetically. The voting landscape corresponds to the spatial clustering that is initially diffuse and is subsequently refined and focused into distinct regions. Two examples of 2D voting are shown in Figure 5, where each nucleus in a mammosphere has been detected.

3.3 Nuclear Segmentation

The voting results are used as vertices to generate a local neighborhood through Voronoi tessellation, as shown in Figures 6a-b. This local neighborhood is then

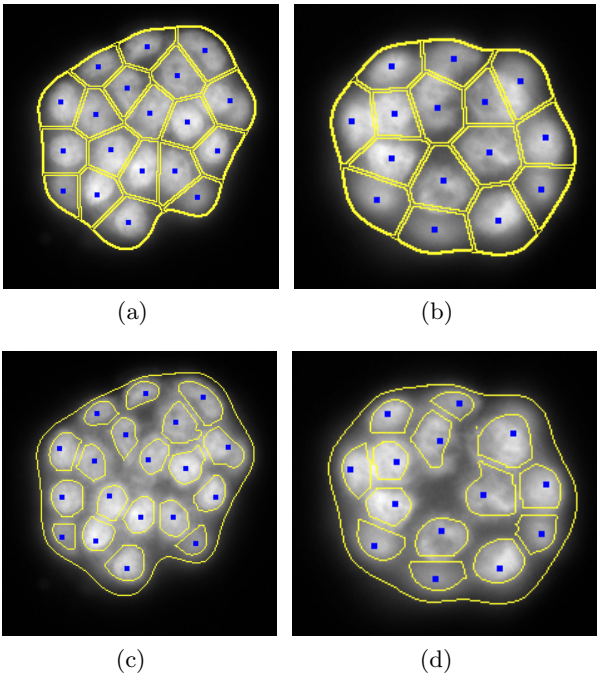


Fig. 6. Steps in refined segmentation: (a-b) Voronoi tessellation of voted regions, (c-d) refined segmentation within Voronoi regions

used to constrain the nuclear segmentation within a small region. The actual nuclear segmentation is performed with the active contour model described earlier [6], and two results are shown in Figure 6c-d.

4 Experimental Results

The proposed approach was implemented and applied to 74 images corresponding to 152 colonies of three-dimensional cell culture models. On average, each colony consisted of 16 cells, and the proposed segmentation algorithm had a 5% error in delineation due to excessive overlap between adjacent nuclei and nonuniform staining of nuclear regions. Figure 7 shows several experimental results corresponding to noisy images, overlapping subcellular compartments, and variation in intensities. Nuclear detection through the voting method indicates high confidence, with only one cell to have been missed in the second image. This is due to the fact that only a small part of the cell is visible at this focal plane. The voting results serve as vertices to initiate Voronoi tessellation and a more detailed segmentation of the nuclear region. In the absence of voting and tessellation, nuclear segmentation within the colony is shown in Figure 8 for comparative analysis. Finally, Figure 9 shows an example of the entire process from colony delineation to nuclear segmentation.

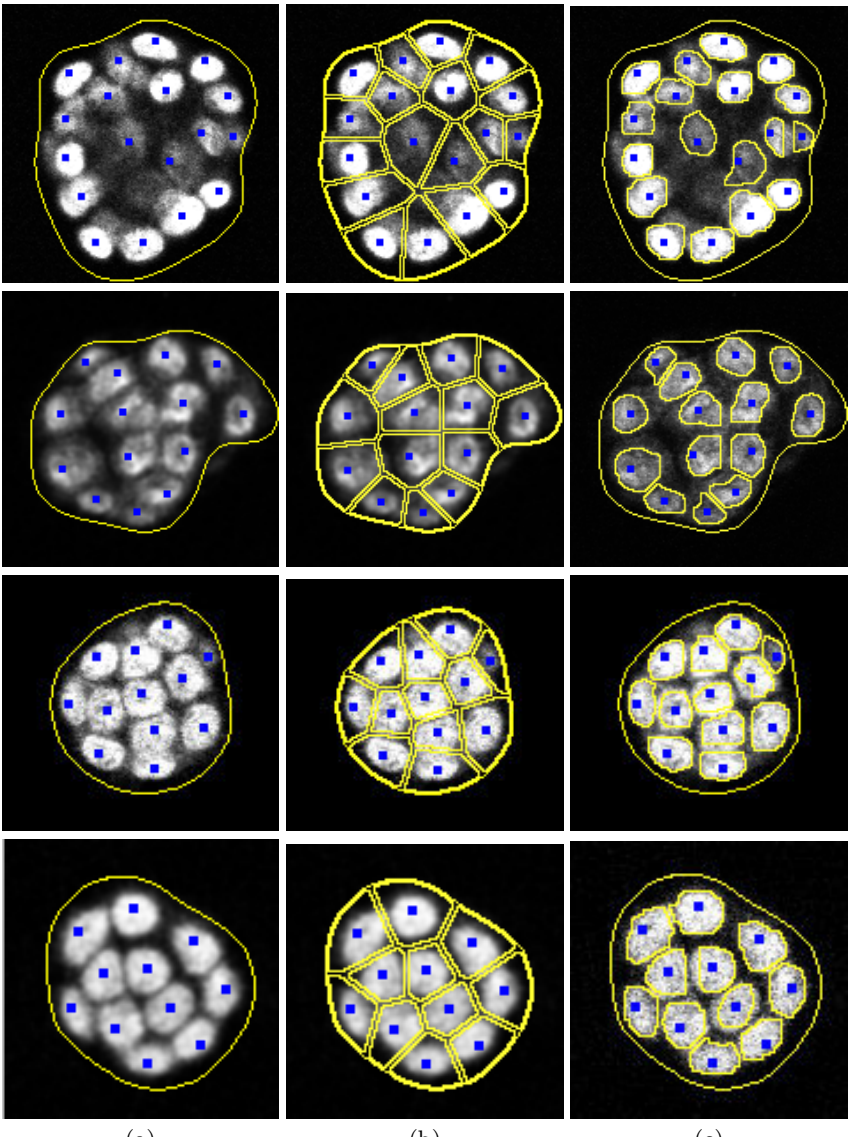
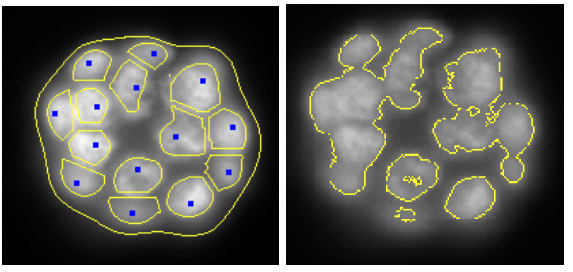


Fig. 7. Several examples of low and high quality data: (a)The voting results; (b) corresponding Voronoi tessellation; and (c) final segmentation through level set method



(a) (b)

Fig. 8. Comparison of segmentation in the absence of a local neighborhood established through Voronoi tessellation: (a) nuclear segmentation with the proposed method; and (b) segmentation results with only level set method applied within the colony

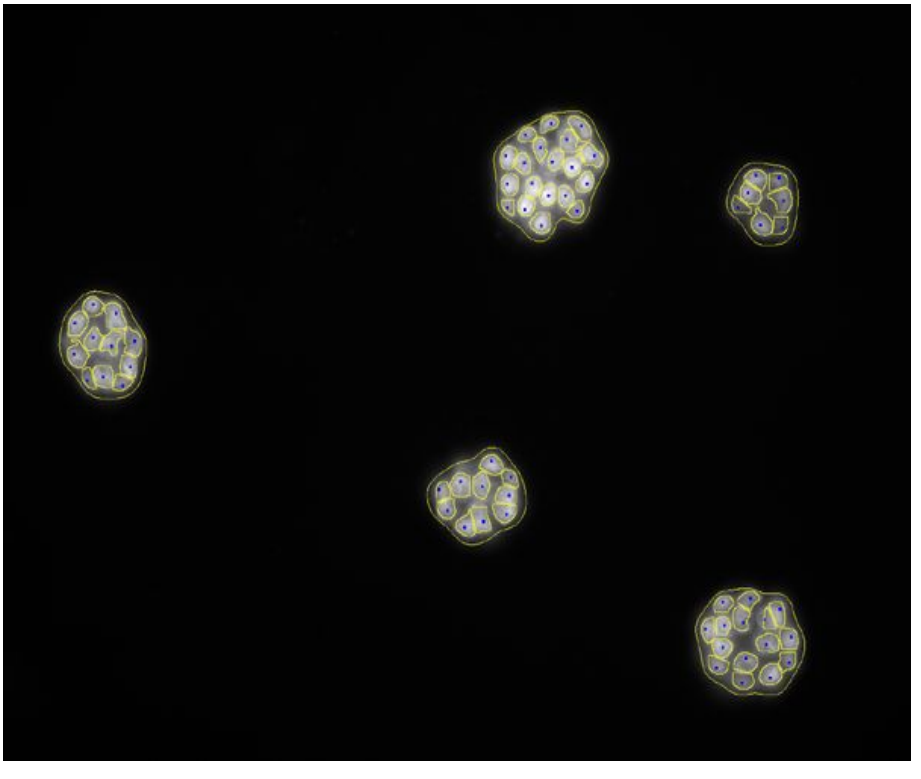


Fig. 9. Multi-colony detection and delineation of each nucleus within the colony

5 Conclusion

We have shown that traditional active contours are not appropriate for segmentation of the nuclear regions; however, by constraining the active contour model to a local region, an improved solution can be realized. These local regions are

estimated based on the fact that the nuclear regions are round and an estimate to their centers of mass can be generated through iterative spatial voting. The voted landscape can then be used to establish a local neighborhood through Voronoi tessellation where the active contour model delineates nuclear regions from their immediate background.

References

1. Weaver, V., Fischer, A., Petersen, O., Bissel, M.: The importance of the microenvironment in breast cancer progression: recapitulation of mammary tumorigenesis using a unique human mammary epithelial cell model and a three-dimensional culture assay. Biochemical Cell Biology **74** (1996) 833–51
2. Yang, Q., Parvin, B.: Harmonic cut and regularized centroid transform for localization of subcelular structures. IEEE Transaction on Biomedical Engineering **50** (2003) 469–476
3. Raman, S., Parvin, B., Maxwell, C., Barcellos-Hoff, M.: Geometric approach segmentation and protein localization in cell cultured assays. In: Int. Symposium on Visual Computing. (2005) 427–436
4. Sarti, A., Ortiz De Solorzano, C., Lockett, S., Malladi, R.: A geometric model for 3-d confocal image analysis. IEEE Transactions on Biomedical Engineering **47** (2000) 1600–1610
5. Caseless, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision **22** (1997) 61–79
6. Chan, T., Vese, L.: Active contours without edges. IEEE Transactions on Image Processing **10** (2001) 266–277

A Semi-automatic 3D Reconstruction Algorithm for Telepresence

Michel Sarkis and Klaus Diepold

Institute for Data Processing (LDV)
Technische Universität München (TUM)
Munich, Germany
{michel, kldi}@tum.de

Abstract. The use of three dimensional computer based models has increased tremendously in the last decades. This is due to the need of numerous emerging applications like telepresence and virtual reality. In the context of telepresence, this paper implements a semi-automatic 3D model reconstruction algorithm from stereo images. The major contribution in this work lies in three main parts: calibration, stereo correspondences, and 3D objects alignement. It is semi-automatic since the calibration technique requires some intervention. However, this will add the ability to change the zoom during the acquisition process.

1 Introduction

Determining the three dimensional structure of a scene from two-dimensional stereo images is a fundamental task in computer vision. Actually, many applications nowadays require the acquisition of the 3D structure of the scene such as robot navigation, virtual reality, and telepresence [1, 2].

The traditional stereo vision techniques used to accomplish this job are dependent on several steps. First, the images have to be calibrated in order to compensate the effect of the radial distortion and the camera intrinsic parameters. Then, the images have to be rectified so that the matching process between the images reduces to a one dimensional search along the epipolar lines [3,4,5,6].

Once the matches are obtained, 3D reconstruction is conducted by triangulation followed by meshing the 3D points for graphic card rendering [2, 7].

In a telepresence scenario, a robot equipped with a stereo camera is installed at a remote area. The human operator located at another place should be able to visualize the scene viewed by the robot using a virtual display such as a head mounted display. Thus, the 3D structure data need to be extracted from the images of the stereo head, sent over the network, and then rendered on the virtual display [8,9]. The schematic of this scenario is shown in Fig. 1. The 3D structure is required since the operator needs the 3D depth information as a feedback that helps controlling the robot in his remote operation; e.g. navigation, surgery, etc. This information cannot be provided from the binocular views alone.

The objective of this paper is to investigate and implement an algorithm to recover the 3D structure of a scene observed by a stereo camera for a telepresence

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 596–606, 2006.
© Springer-Verlag Berlin Heidelberg 2006

scenario. The key issue is to obtain a fast algorithm that results in consistent and dense 3D structures irrespective of the scene under study. The developed method is semi-automatic since the used calibration technique requires some human intervention to compensate for the zooming.

Section 2 presents the semi-automatic calibration procedure used. Section 3 illustrates briefly the rectification algorithm used. Section 4 describes the used stereo correspondence algorithm. Section 5 shows the 3D reconstruction and meshing procedures. Section 6 describes a method to align the 3D structures between consecutive frames. Finally, conclusions are drawn in Section 7.

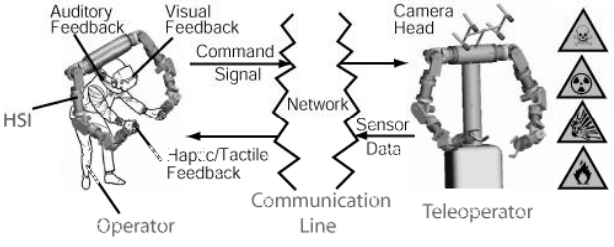


Fig. 1. General Telepresence scenerio

2 Stereo Camera Calibration

The projection of a scene point onto an image is modeled by the equation $\mathbf{x} = \mathbf{PX}$, where $\mathbf{x} = (x, y, 1)^T$ is the vector of the normalized homogenous coordinates of the image point and $\mathbf{X} = (X, Y, Z, 1)^T$ is the corresponding vector of the 3D world point and $\mathbf{P} \in \Re^{3\times4}$ is the camera projection matrix. $\mathbf{P}$ in its turn is decomposed as $\mathbf{P} = \mathbf{K}[\mathbf{Rt}]$, where $\mathbf{K} \in \Re^{3\times3}$ is in upper triangular matrix representing the intrinsic parameters of the camera, $\mathbf{R} \in \Re^{3\times3}$ and $\mathbf{t} \in \Re^{3\times1}$ are the rotation matrix and the translation vector between the world and camera coordinate systems. A good explanation of these parameters can be found in [2].

In order to obtain a metric reconstruction from stereo images, the intrinsic parameters for each camera must be determined along with the metric transformation between the two cameras. In our implementation, it is assumed that the position of the two cameras with respect to each other remains constant. Hence, the transformation between the two cameras which consists of a rotation and a translation should be determined only once and this is done offline using [6].

For a telepresence scenario, zooming plays an important factor, since it allows the operator to better view the far objects and hence, it will make the navigation of the remote robot easier. For this consideration, zooming is allowed to vary in our system. The variation of the zoom will lead to a variation in the intrinsic parameters; thus, a stereo self-calibration method is needed.

Many calibration methods have been developed for stereo rigs in the literature [10,2,11]; however, these methods are not very reliable since they are highly susceptible to errors [12,2]. The major issue is that the estimation in the intrinsic

parameters is not exact which will lead to errors in the 3D reconstruction. Such errors cannot be tolerated in a telepresence scenario.

One way to overcome this problem is to assume that the zoom is only allowed to be changed manually, i.e. the operator controls the zooming of the stereo rig, and that's why the naming semi-automatic. Hence, the variations of the zoom will be known. In this case, the calibration can be done offline at certain levels of the total focal range. The results of the calibration are then saved in a look-up table and later on the zoom will only be allowed to vary at these chosen levels.

This calibration scheme can be performed by any stereo calibration algorithm. In this implementation, the calibration toolbox of Bouguet was used [6]. For our stereo rig, the focal length variation was from 12.5 mm to 85 mm, and the intrinsic parameters were measured at a focal length step-level of 0.5 mm. These values will be the ones stored for the algorithm and allowed to be used by the operator when zooming.

3 Image Rectification

Stereo matching plays the most important role in 3D reconstruction since the matches between the two images leads to the 3D depth information. However, to make the matching process lower in computational complexity, the images have to be rectified in order to reduce the 2D search into a single dimension.

The rectification process is usually done by applying a transformation to the images to make the epipolar lines, where the matches lie, coincide either with the horizontal or the vertical scanlines depending on the stereo rig configuration.

Many rectification algorithms have been proposed in the literature [3, 4, 6]. Any rectification algorithm will work in general. In this implementation however, the rectification algorithm of [6] was used since its main advantage is to take into account the radial distortions of the lenses at the same time, while for the others, the images should be first undistorted and then rectified at a second step.

The key idea behind this technique, is to find two new Projection Matrices (PM) by computing a global rotation that rotates the old PMs around their optical centers until the focal planes are coplanar and, thus, containing the baseline. This rotation ensures that the new epipoles lie at infinity and hence the epipolar lines will be parallel. In its turn, the new translation vector will consist of multiplying the old vector by the global rotation.

The focal length of the new PMs is chosen in a way to take into account the radial and the tangential distortions of the images while the new principal point is computed in such a way to maximize the visible area of the rectified images.

4 Stereo Correspondence

Since the stereo images are now rectified, the stereo correspondence problem has transformed to 1D search across the horizontal scanlines. The stereo matching problem has been dealt with for almost 3 decades and many algorithms have been developed for this purpose. The authors of [5] gave a very good review on

the stereo correspondence algorithms and presented a methodology to test and compare these algorithms. The results can be easily accessed at [13].

In this implementation, we used a modified version of the algorithm proposed in [14]. The choice of this algorithm is due to to its good results when compared to the dynamic programming and graph cuts techniques [14, 15, 16]. In addition, the algorithm resulted in consistent and stable results as will be seen in the presented results. Due to the lack of space, only comparison with the algorithm of [14] will be made.

It should be noted that any algorithm can be used in this case at a condition that it gives stable results if tested on real and complex data sets and if it is fast enough. Therefore, the usage of graph cut algorithms is not a recommended choice due to the huge amount of time needed for the computation; nevertheless, the results in this case are very confident [5, 15].

This section will introduce the Stratified Dense Matching (SDM) algorithm presented in [14], then will present the modifications made to enhance the algorithm. In the end, results on the the data set shown in Fig. 2 are presented.

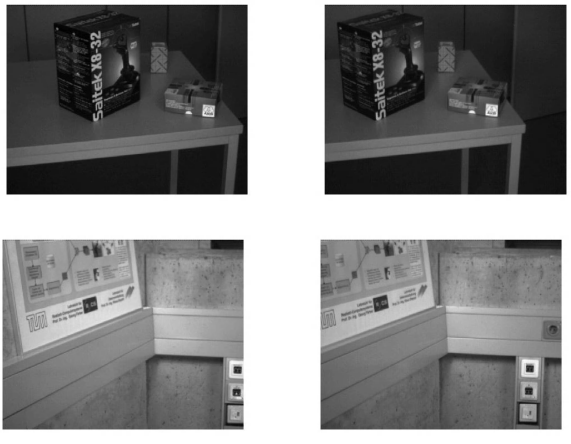


Fig. 2. Two real stereo images captured by our system

4.1 Stratified Dense Matching

The algorithm is divided into four steps. The first one consists of computing pre-matches using a local correlation method. These are then used to trace connected disparity components. This is done by identifying for each pre-match a unique disparity component which is used to find a set of matching windows in the rectified images.

As a third step, for each pre-match the similarity statistics is recomputed using the matching windows found in the previous step. In order to get the different similarity values comparable in their statistical properties, a fixed size disparity

component neighborhood is used to jointly define the windows. Note that the similarity match is only recomputed if the corresponding disparity component is large enough.

In the last step, the final disparity map is computed using the recomputed similarities. The important thing is to obtain the matches with a low error rate. In practice, any stereo correspondence algorithm can be used to determine the final matches. In SDM, the Confidently Stable Matching (CSM) proposed in [17] was used since it produces disparity maps low in error rates.

4.2 Modifications

Since it is a telepresence scenario, the main focus is to obtain a fast algorithm that results in a dense disparity map. Therefore, in order to integrate SDM to our system, some modifications had to be done to enhance it.

First, the images are processed on coarse to fine scheme using image pyramids. Such a methodology is widely used in stereo matching [1, 18, 19]. It is usually applied in order to minimize the computational time required and to obtain more reliable results.

The computation time is minimized since the disparity is computed at the total disparity range only at the coarsest level while at higher levels, the search is reduced to the neighboring pixels to refine the results. In the same manner, the results obtained are more reliable since the candidate matches encountered at the coarsest levels are far less than the ones obtained at the finest level. If multiple matches were found at the coarsest level, the ambiguity is removed by refinement at the next higher level [18].

Returning back to SDM, the images were downsampled and processed. Once a disparity map is obtained, it is upsampled, the disparities are multiplied by two, and then the search is refined using CSM. In other words, only CSM is repeated at higher levels. The modification was implemented in this way, since SDM produces denser matches than CSM and hence, better results [14].

To obtain a sub-pixel accuracy in the results, the disparity values are refined by fitting them to a second degree curve [5]. This is done to reduce the errors introduced through disparity quantization. In this work however, a gaussian low pass filter was used to (smooth) interpolate the pixel values and, thus obtaining the sub-pixel accuracy and removing the false matches at the same time. The standard deviation and the window size of the the gaussian filter varies depending on the disparity level processed. A coarser level disparity map, requires a smaller standard deviation and window to prevent over-smoothing. This step is followed by flood-fill operation to fill in the obvious holes in the disparity map [20]. The overall disparity estimation algorithm is summarized as:

1- Construct the pyramids of the rectified stereo images.
2- Obtain the disparity map at the coarsest level using SDM.
3- Upsample the obtained disparity and multiply it by two.
4- Refine the disparity map using CSM.

5- Repeat steps 3 to 4 until the finest level is achieved.
6- Apply hole filling and then low-pass filtering to obtain sub-pixel accuracy.

4.3 Comparison of the Algorithms

Fig. 3 shows the disparity maps obtained when applying both the SDM and the modified version. The first three sequences represent the Teddy, Cones, and Tsukuba data sets from [5, 13], while the last two sequences are the real data sets shown in Fig. 2. The first column represent the original (rectified) image, the second one shows the disparity output from the SDM algorithm, and the third ones reflects the output of the modified SDM version used in this work. The high disparity values are coded in light pixel values and the low ones are coded to dark values. The black pixels represent the not trusted pixel regions that were labeled as occluded.

In the first two sets, the improvement of the SDM over the modified version is remarkable but not tremendous. It can be noticed that the disparity values have become smoother at the edges; in addition, most of the gaps were correctly filled due to the hole filling process. However going to the Tsukuba data set, the amount of improvement is tremendous. The main reason for this result is due to the lack of structures in some scanlines which make the matching process very difficult due to the many similar candidates. This was solved in the modified version due to the use of the multi-resolution technique from one side which removes partially this ambiguity, in addition to the application of hole filling.

With the data sets of Fig. 2, the same issue has also been concluded. The significant degradation of performance in this case, is due to the lack of structures in the scenes. The modified version still in this case, have resulted in a lot better outcome than the original SDM.

In addition to the above, the use of the coarse to fine approach has enhanced the speed of the disparity computation tremendously. The time needed to compute a disparity map using pyramids varies from 1 to 4 seconds while it requires 6 to 25 seconds in the normal case. The time measurement was done with a 3 GHz Pentium IV processor using Matlab. The timings of the different data sets used are shown in Table 1.

Table 1. Timing results in Seconds of the two algorithms on the different data sets

Data Set	Size	Pyramid Levels	Modified SDM Time	SDM Time
Teddy	375x450	1	1.4	6.5
Cones	375x450	1	1.5	7.3
Tsukuba	288x384	1	1.1	4.2
Data Set 1	640x480	2	3.9	23.2
Data Set 2	640x480	2	3.4	22.5

5 3D Reconstruction

Once the disparity maps are obtained, the 3D reconstruction is performed using triangulation [7]. Since the intrinsic and extrinsic parameters of the cameras are known, the actual depth can be obtained.

In order to construct the final model and display it on a virtual display, meshes should be produced. These were generated using 2.5D Delaunay triangulation. Then, each point was assigned its real 3D coordinate obtained from the 3D reconstruction. The generated textured meshes corresponding to the sequences in Fig. 2 are shown in Fig. 4 and Fig. 5.

6 3D Objects Alignment

In order to enhance the speed of the meshing process, the created 3D objects should be aligned between the consecutive frames. This is necessary since only the additional information should be meshed and added to the initial model.

Many registration algorithms have been developed in the literature [21,22,23, 24]. A very well known technique is the Iterative Closest Point (ICP) algorithm which tries to align the 3D objects iteratively by minimizing the distances among the 3D points. Another algorithm is the Bundle Adjustment (BA) which tries to find the correct pose by iterative reprojections of the 3D points until the error is minimized. Nevertheless, in these both cases, an initial camera pose estimate need to be determined.

Since our system is fully calibrated, the initial pose estimate can be easily calculated from the essential matrix $\mathbf{E}$ between the successive frames since $\mathbf{E} = [\mathbf{t_a}]_\times \mathbf{R_a}$, where $\mathbf{t}_a$ is the desired translation and $\mathbf{R}_a$ is the desired rotation [2]. The best way to compute the essential matrix is using the celebrated 5 point pose algorithm, since it has outperformed other techniques like the 6 and 7 point pose algorithms [23].

Nevertheless, the obtained transformation will only be defined up to a similarity transformation which will deviate from the wanted one. The authors in [24] have solved this problem by calibrating the system at the beginning in order to obtain the similarity transformation and then compensate for it in the computations at later stages. This solution is not very favorable since it requires additional computational effort at the beginning besides the one needed for the calibration of the stereo rig.

In our case however, this similarity transformation can, unlike [24], be obtained automatically due to the fact that our system consists of a stereo rig and not a single moving camera. This is done by first computing the new poses of the left and right stereo cameras using the 5 point pose algorithm [23]. Suppose now that $\mathbf{H}_0$ and $\mathbf{H}_1$ are the relative transformations between the new and old positions of the left camera and the relative poisition between the left and the right camera respectively. Since, the stereo rig is already extrinsically calibrated, the relative position between the left and right camera is known, i.e. $\mathbf{R}$ and $\mathbf{t}$ computed in Section 2. Thus, the similarity transformation can be obtained by

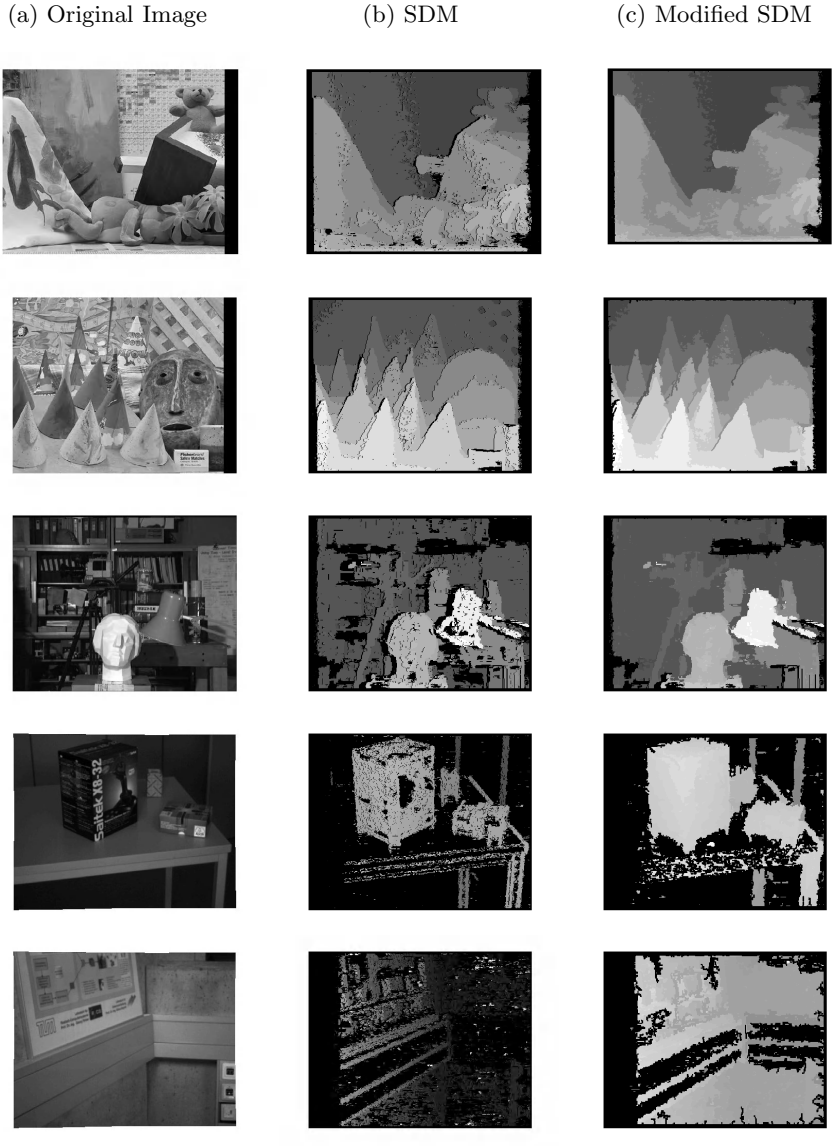


Fig. 3. From up till down: results of the algorithms applied on: Teddy, Cones, and Tsukuba data sets from middlebury [5, 13]. The last two data sets are the results of stereo images in Fig. 2.

finding the transformation $\mathbf{H}_s$ that relates $\mathbf{H}_1$ to the extrinsic parameters of the stereo rig in a least squares sense. As result the transformation that aligns the 3D object will be: $\mathbf{H}_a = \mathbf{H}_s.\mathbf{H}_0$. This transformation can then be refined using BA or ICP. Fig. 6 shows the dense reconstructed 3D structure obtained for the data sets in Fig. 2 after aligning several views of the objects.

(a) Left Image (b) Right Image (c) 3D Mesh



Fig. 4. 3D reconstruction of the first stereo sequence in Fig. 2

(a) Left Image (b) Right Image (c) 3D Mesh



Fig. 5. 3D reconstruction of the second stereo sequence in Fig. 2

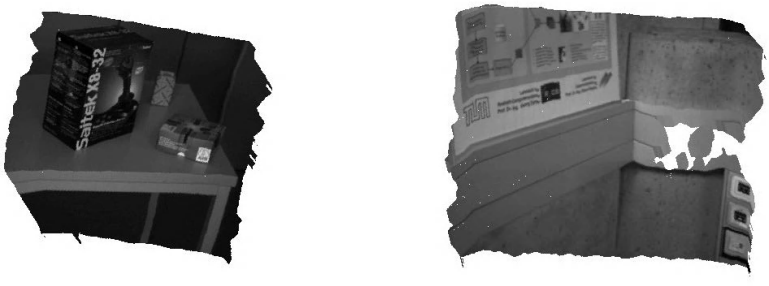


Fig. 6. Dense 3D reconstruction of the data sets in Fig. 2 after several frames

7 Conclusion

This work presents a semi-automatic 3D reconstruction algorithm with application for telepresence. The algorithm is semi-automatic since the operator is only allowed to change the zoom at some predefined steps. The main contributions of this work reside in the technique used to allow for the zoom variations, the modification of existing stereo correspondence algorithm in order to achieve denser and faster reconstruction, and the alignment of the 3D reconstructed objects which also increases the overall speed. However, many points are still open

for further research. First, instead of storing a lookup table containing the calibration results, one might try to derive a model of the variation of the camera parameters, mainly the focal length, which can be used to produce the calibration data instead. Another issue is the computation of the disparity map which is done for each frame. This can be improved by predicting the new disparity map using the old disparity map and the motion parameters.

Acknowledgement

This research is sponsored by the German Research Foundation (DFG) as a part of the SFB 453 project, High-Fidelity Telepresence and Teleaction.

References

1. W. G. Kropatsch and H. Bischof, Eds., *Digital Image Analysis*, Springer-Verlag, 1 edition, 2001.
2. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2 edition, 2004.
3. M. Pollefeys, R. Koch, and L. J. Van Gool, "A simple and efficient rectification method for general motion," in *International Conference on Computer Vision*, 1999, pp. 496–501.
4. A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, vol. 12, no. 1, pp. 16–22, 2000.
5. D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, April 2001.
6. J. Y. Bouguet, "Camera calibration toolbox for matlab," 2003.
7. E. Trucco and A. Verri, *Introductory techniques for 3-D computer vision*, Prentice-Hall, 1 edition, 1998.
8. G. H. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Springer Link*, vol. 16, no. 10, pp. 1389–1402, 2002.
9. T. Burkert, J. Leupold, and G. Passig, "A photorealistic predictive display," *Presence: teleoperators and virtual environments*, vol. 13, no. 1, pp. 22–43, February 2004.
10. R. Horaud, G. Csurka, and D. Demirdijian, "Stereo calibration from rigid motions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1446–1452, 2000.
11. Y. Li and Y. S. Hung, "A stratified self-calibration method for a stereo rig in planar motion with varying intrinsic parameters," *Springer Verlag*, vol. 3175, pp. 318–325, 2004.
12. R. Koch, "3-D scene modeling from image sequences," *ISPRS Archives*, vol. XXXIV, no. 3, September 2003.
13. "Middleburry stereo data sets," http://cat.middlebury.edu/stereo/data.html.
14. J. Kostkova and R. Sara, "Stratifed dense matching for stereopsis in complex structures," in *Britich Machine Vision Conference*, 2003.
15. V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *International Conference on Computer Vision*, 2001.

16. I. J. Cox, S. L. Higorani, S. B. Rao, and B. M. Maggs, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, vol. 63, no. 3, 1996.
17. R. Sara, "Finding the largest unambiguous component of stereo matching," in *ECCV*, 2002.
18. L. Falkenhagen, "Hierarchical blockbased disparity estimation considering neighbourhood constraints," in *International workshop on SNHC and 3D Imaging*, 1997.
19. C. Leung, B. Appleton, and C. Sun, "Fast stereo matching by iterated dynamic programming and quadtree subregioning," in *Britich Machine Vision Conference*, 2004.
20. P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, 1 edition, 1999.
21. B. Triggs, P. McLauchlan, R. Harltey, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," *Springer Lecture Notes on Computer Science, Springer Verlag*, vol. 1883, 2000.
22. S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third International Conference on 3D Digital Imaging and Modeling*, 2001.
23. D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, 2004.
24. W. Zhao, D. Nister, and S. Hsu, "Alignment of continuous video onto 3d point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005.

Motion Detection Using an Improved Colour Model*

Horst Wildenauer**, Philipp Blauensteiner, Allan Hanbury, and Martin Kampel

Pattern Recognition and Image Processing Group – TU Vienna
Favoritenstr 9 - 11, 1040 Vienna, Austria
{wilde, blau, hanbury, kampel}@prip.tuwien.ac.at

Abstract. We discuss common colour models for background subtraction and problems related to their utilisation. A novel approach to represent chrominance information more suitable for robust background modelling and shadow suppression is proposed. Our method relies on the ability to represent colours in terms of a 3D-polar coordinate system having saturation independent of the brightness function; specifically, we build upon an Improved Hue, Luminance, and Saturation space (IHLS). The additional peculiarity of the approach is that we deal with the problem of unstable hue values at low saturation by modelling the hue-saturation relationship using saturation-weighted hue statistics. The effectiveness of the proposed method is shown in an experimental comparison with approaches based on Normalised RGB, $c_1c_2c_3$, and HSV.

1 Introduction

The underlying step of visual surveillance applications like target tracking and scene understanding is the detection of moving objects. Background subtraction algorithms are commonly applied to detect these objects of interest by the use of statistical colour background models. Many present systems exploit photometric colour invariants [1] to achieve a certain degree of insensitivity with respect to changes in scene illumination.

Hong and Woo [2] apply the Normalised RGB space in their background segmentation system. McKenna et al. [3] use this colour space in addition to gradient information for their adaptive background subtraction. The AVITRACK project [4] utilises Normalised RGB for change detection and adopts the shadow detection proposed by Horprasert et al. [5]. Salvador et al. [6] compare several colour invariants and use the $c_1c_2c_3$ model for their shadow detection approach.

Beside the aforementioned approaches, representations of the RGB colour space in terms of 3D-polar coordinates (hue, saturation, and brightness) are used for change detection and shadow suppresion in surveillance applications. François and Medioni [7] suggest the application of HSV for background modelling for

* This work was supported by the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04) and the CABS project.

** Currently Horst Wildenauer is also with the Automation & Control Inst., TU Vienna.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 607–616, 2006.
© Springer-Verlag Berlin Heidelberg 2006

real-time video segmentation. In their work, a complex set of rules is introduced to reflect the relevance of observed and background colour information during change detection and model update. Cucchiara et al. [8] propose a RGB-based background model which they transform to the HSV representation in order to utilise the properties of HSV chrominance information for shadow suppression.

Our approach differs from the aforementioned in the way that we build upon the IHLS colour space, which is more suitable for background subtraction. Additionally, we propose the application of saturation-weighted hue statistics [9] to deal with unstable hue values at weakly saturated colours. Also, a technique to efficiently classify changes in scene illumination (e.g., shadows), modelling the relationship between saturation and hue has been devised.

The remainder of this paper is organised as follows: In Section 2 the Improved Hue, Luminance and Saturation (IHLS) colour space is described. Section 3 gives a short overview of circular colour statistics, which have to be applied on the hue as angular value. Section 4 reviews saturation weighted hue statistics. Section 5 presents how these statistics can be applied in order to model the background in image sequences. The conducted experiments and their results are presented in Section 6. Section 7 concludes the paper.

2 IHLS Colour Space

The Improved Hue, Luminance and Saturation (IHLS) colour space was introduced in [10]. It is obtained by placing an *achromatic axis* through all the grey ($R = G = B$) points in the RGB colour cube, and then specifying the coordinates of each point in terms of position on the achromatic axis (brightness), distance from the axis (saturation s) and angle with respect to pure red (hue θ^H). The IHLS model is improved with respect to the similar colour spaces (HLS, HSI, HSV, etc.) by removing the normalisation of the saturation by the brightness. This has the following advantages: (a) the saturation of achromatic pixels is always low and (b) the saturation is independent of the brightness function used. One may therefore choose any function of R, G and B to calculate the brightness.

The following formulae are used for the conversion from RGB to hue θ^H, luminance y and saturation s of the IHLS space:

$$
\begin{aligned}
s &= \max(R,G,B) - \min(R,G,B) \\
y &= 0.2125R + 0.7154G + 0.0721B \\
cr_x &= R - \frac{G+B}{2},\ cr_y = \frac{\sqrt{3}}{2}(B-G) \\
cr &= \sqrt{cr_x^2 + cr_y^2} \\
\theta^H &= \begin{cases} \text{undefined} & \text{if } cr = 0 \\ \arccos\left(\frac{cr_x}{cr}\right) & \text{elseif } cr_y \leq 0 \\ 360^\circ - \arccos\left(\frac{cr_x}{cr}\right) & \text{else} \end{cases}
\end{aligned} \tag{1}
$$

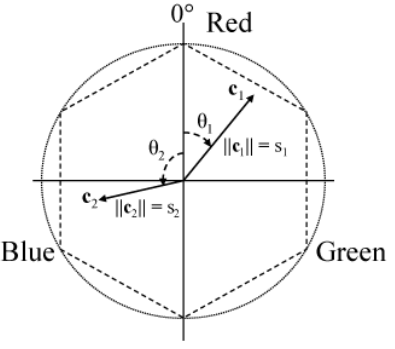


Fig. 1. The chromatic plane of the IHLS color space

where cr_x and cr_y denote the chrominance coordinates and $cr \in [0, 1]$ the chroma. The saturation assumes values in the range $[0, 1]$ independent of the hue angle (the maximum saturation values are shown by the circle on the chromatic plane in Figure 1). The chroma has the maximum values shown by the dotted hexagon in Figure 1. When using this representation, it is important to remember that the hue is undefined if $s = 0$, and that it does not contain much useable information when s is low (i.e. near to the achromatic axis).

Figure 2 shows a comparison of the different formulations of saturation. The undesirable effects created by saturation normalisation are easily perceivable, as some dark, colourless regions (eg., the bushes and the side window of the driving car) reach higher saturation values than their more colourful surroundings. Also, note the artefacts resulting from the singularity of the saturation at the black vertex of the RGB-cube (again, the bushes and the two bottom right cars).

3 Hue Statistics

In a 3D-polar coordinate space, standard (linear) statistical formulae can be used to calculate statistical descriptors for brightness and saturation coordinates. The hue, however, is an angular value, and consequently the appropriate methods from circular statistics are to be used.

Let θ_i^H, $i = 1, \ldots, n$ be n observations sampled from a population of angular hue values. Then the vector $\mathbf{h}_i$ pointing from $\mathbf{O} = (0, 0)^T$ to the point on the circumference of the unit circle, corresponding to θ_i^H, is given by the Cartesian coordinates $(\cos\theta_i^H, \sin\theta_i^H)^T$. [1]

The mean direction $\overline{\theta}^H$ is defined to be the direction of the resultant of the unit vectors $\mathbf{h_1}, \ldots, \mathbf{h_n}$ having directions θ_i^H. That is, we have

$$\overline{\theta}^H = \arctan2\,(\mathcal{S}, \mathcal{C})\,, \tag{2}$$

[1] Note that when using the IHLS space (Eq. 1) no costly trigonometric functions are involved in the calculation of $\mathbf{h}_i$, since $cos(\theta_i^H) = cr_x/cr$ and $sin(\theta_i^H) = -cr_y/cr$.



Fig. 2. Examples of chromatic components. Image from the *PETS2001* dataset (top), it's normalised saturation (cylindrical HSV) (left), IHLS saturation (right).

where

$$\mathcal{C} = \sum_{i=1}^{n} \cos\theta_i^H, \;\; \mathcal{S} = \sum_{i=1}^{n} \sin\theta_i^H, \tag{3}$$

and $\arctan2(y, x)$ is the four-quadrant inverse tangent function.
The mean length of the resultant vector

$$\overline{\mathcal{R}} = \frac{\sqrt{\mathcal{C}^2 + \mathcal{S}^2}}{n} \tag{4}$$

is an indicator of the dispersion of the observed data. If the n observed directions θ_i^H cluster tightly about the mean direction $\overline{\theta}^H$ then $\overline{\mathcal{R}}$ will approach 1. Conversely, if the angular values are widely dispersed, $\overline{\mathcal{R}}$ will be close to 0. The *circular variance* is defined as

$$\mathcal{V} = 1 - \overline{\mathcal{R}}. \tag{5}$$

While the circular variance differs from the linear statistical variance in being limited to the range $[0, 1]$, it is similar in the way that lower values represent less dispersed data. Further measures of circular data distribution are given in [11].

4 Saturation-Weighted Hue Statistics

The use of statistics solely based on the hue has the disadvantage of ignoring the tight relationship between the chrominance components hue and saturation. For weakly saturated colours the hue channel is unimportant and behaves unpredictably in the presence of colour changes induced by image noise. In fact, for colours with zero saturation the hue is undefined.

As one can see in Figure 1, the chromatic components may be represented by means of Cartesian coordinate vectors $\mathbf{c}_i$ with direction and length given by hue and saturation respectively. Using this natural approach, we introduce the aforementioned relationship into the hue statistics by weighting the unit hue vectors $\mathbf{h}_i$ by their corresponding saturations s_i.

Let (θ_i^H, s_i), $i = 1, \ldots, n$ be n pairs of observations sampled from a population of hue values and associated saturation values. We proceed as described in Section 3, with the difference that instead of calculating the resultant of unit vectors, the vectors $\mathbf{c}_i$, which we will call *chrominance vectors* throughout this paper, have length s_i.

That is, we weight the vector components in Eq. 3 by their saturations s_i

$$\mathcal{C}_s = \sum_{i=1}^{n} s_i \cos\theta_i^H, \quad \mathcal{S}_s = \sum_{i=1}^{n} s_i \sin\theta_i^H, \tag{6}$$

and choose the mean resultant length of the chrominance vectors (for other possible formulations see, e.g. [9]) to be

$$\overline{\mathcal{R}}_n = \frac{\sqrt{\mathcal{C}_s^2 + \mathcal{S}_s^2}}{n}. \tag{7}$$

Consequently, for the mean resultant chrominance vector we get

$$\overline{\mathbf{c}}_n = (\mathcal{C}_s/n, \mathcal{S}_s/n)^T. \tag{8}$$

Here, the length of the resultant is compared to the length obtained if all vectors had the same direction and maximum saturation. Hence, $\overline{\mathcal{R}}_n$ gives an indication of the saturations of the vectors which gave rise to the mean of the chrominance vector, as well as an indication of the angular dispersion of the vectors. To test if a mean chrominance vector $\overline{\mathbf{c}}_\mathbf{n}$ is similar to a newly observed chrominance vector, we use the Euclidean distance in the chromatic plane:

$$D = \sqrt{(\overline{\mathbf{c}}_n - \mathbf{c}_o)^T (\overline{\mathbf{c}}_n - \mathbf{c}_o)} \tag{9}$$

with $\mathbf{c}_o = s_o \mathbf{h}_o$. Here, $\mathbf{h}_o$ and s_o denote the observed hue vector and saturation respectively.

5 The IHLS Background Model

With the foundations laid in Section 4 we proceed with devising a simple background subtraction algorithm based on the IHLS colour model and saturation-weighted hue statistics. Specifically, each background pixel is modelled by its mean luminance μ_y and associated standard deviation σ_y, together with the mean chrominance vector $\overline{\mathbf{c}}_n$ and the mean Euclidean distance σ_D between $\overline{\mathbf{c}}_n$ and the observed chrominance vectors (see Eq. 9).

On observing the luminance y_o, saturation s_o, and a Cartesian hue vector $\mathbf{h}_o$ for each pixel in a newly acquired image, the pixel is classified as foreground if:

$$|(y_o - \mu_y)| > \alpha\sigma_y \vee \|\overline{\mathbf{c}}_n - s_o\mathbf{h}_o| > \alpha\sigma_D, \tag{10}$$

where α is the foreground threshold, usually set between 2 and 3.5.

In order to decide whether a foreground detection was caused by a moving object or by its shadow cast on the static background, we exploit the chrominance information of the IHLS space. A foreground pixel is considered as shaded background if the following three conditions hold:

$$y_o < \mu_y \ \wedge \ |y_o - \mu_y| < \beta\mu_y \tag{11}$$

$$s_o - \overline{\mathcal{R}}_n < \tau_{ds} \tag{12}$$

$$\|\mathbf{h}_o\overline{\mathcal{R}}_n - \overline{\mathbf{c}}_n\| < \tau_h, \tag{13}$$

where $\overline{\mathcal{R}}_n = \|\overline{\mathbf{c}}_n\|$ (see Eq. 7).

These equations are designed to reflect the empirical observations that cast shadows cause a darkening of the background and usually lower the saturation of a pixel, while having only limited influence on its hue. The first condition (Eq. 11) works on the luminance component, using a threshold β to take into account the strength of the predominant light source. Eq. 12 performs a test for a lowering in saturation, as proposed by Cucchiara et al. [8]. Finally, the lowering in saturation is compensated by scaling the observed hue vector $\mathbf{h}_o$ to the same length as the mean chrominance vector $\overline{\mathbf{c}}_n$ and the hue deviation is tested using the Euclidean distance (Eq. 13). This, in contrast to a check of angular deviation (see Eq. 16 or [8]), also takes into account the model's confidence in the learned chrominance vector. That is, using a fixed threshold τ_h on the Euclidean distance relaxes the angular error-bound in favour of stronger hue deviations at lower model saturation value $\overline{\mathcal{R}}_n$, while penalising hue deviations for high saturations (where the hue is usually more stable).

6 Experiments and Results

In the experimental comparison, we make use of four colour models: our IHLS model, the HSV model [8] the Normalised RGB (NRGB) model [12], and the $c_1c_2c_3$ model [1]. NRGB and $c_1c_2c_3$ are invariant to changes in viewing direction, object geometry and illumination (assuming dichromatic reflectance and white illumination).

We compare our approach to the following models: a RGB background model using NRGB-based (*RGB+NRGB*), HSV-based (*RGB+HSV*), or $c_1c_2c_3$-based (*RGB+C1C2C3*) shadow detection, and a method relying on NRGB for both background modelling and shadow detection (*NRGB+NRGB*).

All methods were implemented using the *Colour Mean and Variance* approach [13] to model the background. A pixel is considered foreground if $|c_o - \mu_c| > \alpha\sigma_c$ for any channel c with $c \in \{r, g, l\}$ for the Normalised RGB, where r and g are the normalised colour channels and l is the luminance, and $c \in$

$\{R, G, B\}$ for the RGB space respectively. c_o denotes the observed value, μ_c its mean, σ_c the standard deviation, and α the foreground threshold.

The tested background models are maintained by means of exponentially weighted averaging [13] using different learning rates for background and foreground pixels. During the experiments the same learning and update parameters were used for all background models.

For Normalised RGB (*RGB+NRGB*, *NRGB+NRGB*), shadow suppression was implemented based on Horprasert's approach [4,5]. Each foreground pixel is classified as shadow if:

$$\begin{aligned} l_o < \mu_l \wedge l_o > \beta\mu_l \\ |r_o - \mu_r| < \tau_c \wedge |g_o - \mu_g| < \tau_c, \end{aligned} \tag{14}$$

where β and τ_c denote thresholds for the maximum allowable change in the intensity and colour channels, so that a pixel is considered as shaded background.

The shadow check for the $c_1c_2c_3$-approach (*RGB+C1C2C3*) is performed as described in [6]:

$$\begin{aligned} l_o < \mu_l \wedge l_o > \beta\mu_l \\ |c_1 - \mu_{c_1}| < \tau_c \wedge |c_2 - \mu_{c_2}| < \tau_c \wedge |c_3 - \mu_{c_3}| < \tau_c. \end{aligned} \tag{15}$$

In the HSV-based approach(*RGB+HSV*) the RGB background model is converted into HSV (specifically, the reference luminance μ_v, saturation μ_s, and hue μ_θ) before the following shadow tests are applied. A foreground pixel is classified as shadow if:

$$\begin{aligned} \beta_1 \leq \frac{v_o}{\mu_v} \leq \beta_2 \\ s_o - \mu_s \leq \tau_s \\ |\theta_o - \mu_\theta| \leq \tau_\theta. \end{aligned} \tag{16}$$

The first condition tests the observed luminance v_o for a significant darkening in the range defined by β_1 and β_2. On the saturation s_o a threshold on the difference is performed. Shadow lowers the saturation of points and the difference between images and the reference is usually negative for shadow points. The last condition takes into account the assumption that shading causes only small deviation of the hue θ_o [8].

For the evaluation of the algorithms, three video sequences were used. As an example for a typical indoor scene *Test Sequence 1*, recorded by an AXIS-211 network camera, shows a person moving in a stairway. For this sequence, ground truth was generated manually for 35 frames. *Test Sequence 2* was recorded with the same equipment and shows a person waving books in front of a coloured background. For this sequence 20 ground truth frames were provided. Furthermore the approaches were tested on 25 ground truth frames from the *PETS2001* dataset 1 (camera 2, testing sequence). Example pictures from the three video sequences can be found in Figure 3.



Fig. 3. Evaluation dataset. Left to right: *Test Sequence 1, Test Sequence 2, PETS2001.*

For a fair comparison of the change detection algorithms with regard to their various decision parameters, receiver operating characteristics (ROC) based on detection rate (DR) and false alarm rate (FR) were used:

$$DR = \frac{TP}{FN + TP},\ FR = \frac{FP}{N - (FN + TP)}. \tag{17}$$

Here TP denotes the number of true positives, FN the number of false negatives, FP the number of false positives, and N the total number of pixels in the image.

For a dense evaluation, we experimentally determined suitable ranges for all parameters and sub-sampled them in ten steps. Figure 4 shows the convex hulls of the points in ROC space obtained for all parameter combinations. We also want to point out that *RGB+HSV* was tested with unnormalised and normalised saturation; however, since the normalised saturation consistently performed worse, we omit the results in the ROC for clarity of presentation.

As one can see, our approach outperforms its competitors on *Test Sequence 1.* One reason for this is the insensitivity of the *RGB+NRGB* and *NRGB+NRGB* w.r.t small colour differences at light, weakly saturated colours. *RGB+HSV*, however, suffered from the angular hue test reacting strongly to unstable hue values close to the achromatic axis. For conservative thresholds (i.e. small values for τ_c or τ_θ) all four approaches either detected shadows on the wall as foreground, or, for larger thresholds classified the beige t-shirt of the person as shadow. For *Test Sequence 2* the advantageous behaviour of our approach is even more evident. Although the scene is composed of highly saturated, stable colours, *RGB+C1C2C3*, *RGB+NRGB*, and *NRGB+NRGB* show rather poor results, again stemming from their insufficient sensitivity to bright colours. *RGB+C1C2C3* gave slightly better results, but could not take full advantage of the available colour information; similar hue values for the books and the background resulted in incorrectly classified shadow regions. The *PETS2001* sequence shows the problems of background modelling using NRGB as is mentioned in [14]: due to the low brightness and the presence of noise in this scene, the chromatic components are unstable and therefore the motion detection resulted in a significantly increased number of false positives. *RGB+NRGB*, *RGB+C1C2C3*, and our approach exhibit similar performance, mostly relying on brightness checks, since there was not much useable information in shadow regions. *RGB+HSV* performed less well, having problems to cope with the unstable hue information in dark areas.

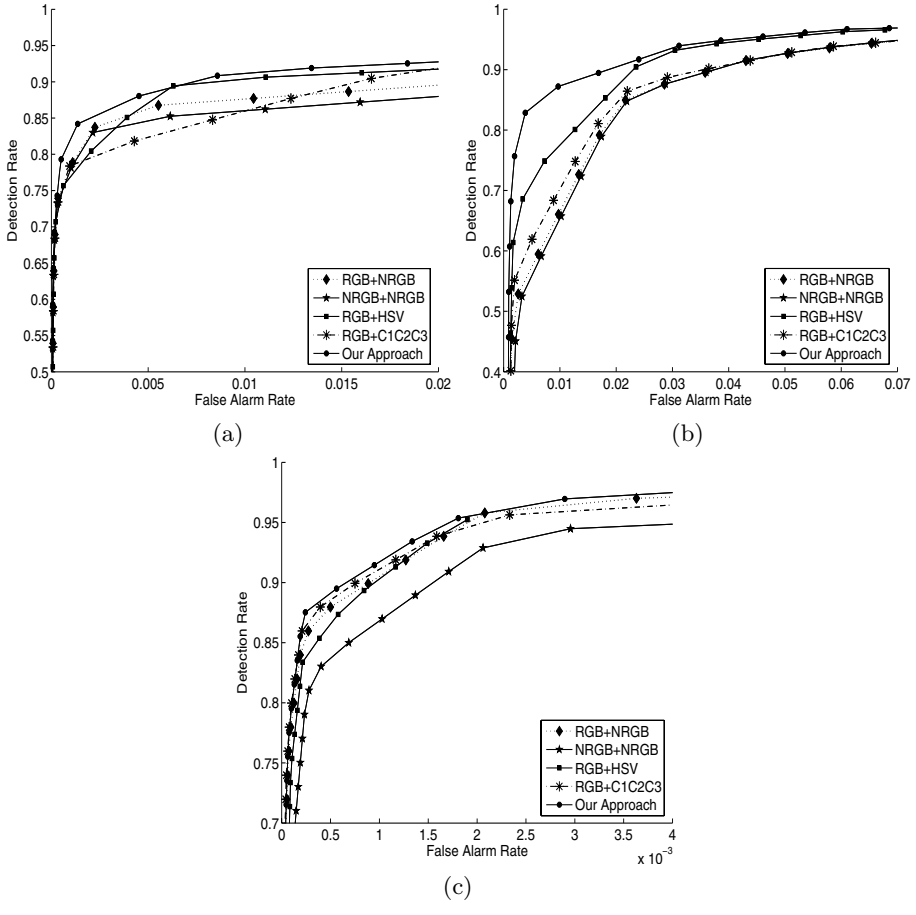


Fig. 4. Experimental results: ROCs for *Test Sequence 1* (a), *Test Sequence 2* (b), and *PETS2001* Dataset 1 (c)

7 Conclusion

We proposed the usage of the IHLS colour space for change detection and shadow suppression in visual surveillance tasks. In the proposed framework, we advocate the application of saturation-weighted hue statistics to deal with the problem of the unstable hue channel at weakly saturated colours.

We have shown that our approach outperforms the HSV model as well as the photometric colour invariants NRGB and $c_1c_2c_3$ in several challenging sequences. Furthermore, our experiments have shown that it is not advisable to use NRGB for background modelling due to its unstable behaviour in dark areas.

One problem of our approach, however, is the fact that due to the use of saturation weighted hue statistics, it is impossible to tell whether a short chrominance vector in the background model is the result of unstable hue information

or of a permanent low saturation. Although in the conducted experiments no impairments were evident, it is a subject of further research in which cases this shortcoming poses a problem. Other fields of interest are the examination of alternatives to the Euclidean distance for the comparison of the chrominance vectors and an experimental in-depth-investigation of the shadow classification.

References

1. Gevers, T., Smeulders, A.W.: Colour-based object recognition. Pattern Recognition **32** (1999) 435–464
2. Hong, D., Woo, W.: A Background Subtraction for Vision-based User Interface. In: 4th Intl. Conference on Information, Communications and Signal Processing and Fourth IEEE Pacific-Rim Conference on Multimedia. (2003) 263–267
3. McKenna, S., Jabri, S., Duric, Z., Wechsler, H., Rosenfeld, A.: Tracking Groups of People. Computer Vision and Image Understanding **80** (2000) 42–56
4. Ferryman, J., Borg, M., Thirde, D., Fusier, F., Valentin, V., Brémond, F., Thonnat, M., Aguilera, J., Kampel, M.: Automated Scene Understanding for Airport Aprons. In: Australian Joint Conference on Artificial Intelligence, Australia (2005) 593–503
5. Horprasert, T., Harwood, D., Davis, L.: A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. In: IEEE Conference on Computer Vision, FRAME-RATE Workshop. (1999)
6. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. Computer Vision and Image Understanding **95**(2) (2004) 238–259
7. François, A.R.J., Medioni, G.G.: Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams. In: International Conference on Imaging Science, Systems, and Technology. (1999) 227–232
8. Cucchiara, R., Grana, C., Piccardi, M., Prati, A., Sirotti, S.: Improving Shadow Suppression in Moving Object Detection with HSV Color Information. In: Intelligent Transport Systems, IEEE (2001) 334–339
9. Hanbury, A., Kropatsch, W.G.: Colour Statistics for Matching in Image Databases. In Beleznai, C., Schoegl, T., eds.: 27th OEAGM Workshop, Austria (2003)
10. Hanbury, A.: A 3D-polar coordinate colour representation well adapted to image analysis. In: Proceedings of the Scandinavian Conference on Image Analysis (SCIA). (2003) 804–811
11. Mardia, K.V.: Satistics of Directional Data. Academic Press, London (1972)
12. Finlayson, G.D., Schiele, B., Crowley, J.L.: Comprehensive Colour Image Normalization. In: 5th European Conference on Computer Vision. (1998) 475–490
13. Wren, C.R., Azarbayejami, A., Darrel, T., Pentland, A.: Pfinder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7) (1997) 780–785
14. Kender, J.R.: Saturation, Hue and Normalized Colors: Calculation, Digitisation Effects, and Use. Technical report, Department of Computer Science, Carnegie Mellon University (1976)

Combining Pixelization and Dimensional Stacking

John T. Langton[1], Astrid A. Prinz[2], and Timothy J. Hickey[3]

[1] Charles River Analytics
jlangton@cra.com
www.cra.com
[2] Department of Biology, Emory University
astrid.prinz@emory.edu
[3] Computer Science Department, Brandeis University
tim@cs.brandeis.edu

Abstract. The combination of pixelization and dimensional stacking yields a highly informative visualization that uniquely facilitates feature discovery and exploratory analysis of multidimensional, multivariate data. Pixelization is the mapping of each data point in some set to a pixel in an image. Dimensional stacking is a layout method where N dimensions are projected into 2. We have combined both methods to support visual data mining of a vast neuroscience database. Images produced from this approach have now appeared in the Journal of Neurophysiology [1] and are being used for educational purposes in neuroscience classes at Emory University. In this paper we present our combination of dimensional stacking and pixelization, our extensions to these methods, and how our techniques have been used in neuroscience investigations.

1 Introduction

Pixelization is a visualization technique that maps each data point of some set to a pixel in a 2D image [2]. Dimensional stacking is a layout method that projects the dimensions of a multidimensional dataset onto the axes of an information display[3]. While both of these techniques have been used for visualizing large databases of multidimensional, multivariate information, their combination has not been thoroughly investigated. We have found that combining these methods uniquely facilitates visual data mining and exploratory data analysis. In particular, visual clustering within this scheme is entirely based on the order of dimensions on each axis. By allowing users to reorder dimensions and specify pixel colorings with SQL queries, data trends and functional dependencies can be revealed. This process has been found to motivate hypothesis about the underlying data and inform statistical analysis and user queries.

In this paper we detail our approach of applying a dimensionally stacked layout to pixelization and how it has aided in the analysis of a vast neuroscience database. We first describe the data we were tasked with visualizing in Section 2. We then present an overview of related research in multidimensional visualization in Section 3. Section 4 presents the application of our approach to the neuroscience data while Section 5 discusses its limitations and how to apply it

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 617–626, 2006.
© Springer-Verlag Berlin Heidelberg 2006

in the general case. We conclude in Section 6 with some observations of how our techniques have been used in practice.

2 The Data

The visualization methods presented in this paper began as a collaboration between researchers in the neuroscience and computer science departments of Brandeis University. A database was constructed to better understand the role of a neuron's individual conductances in determining its behavior. Data from many simulations of a model neuron was classified according to electrical activity and recorded. Figure 1 shows the type of voltage plots generated by the simulator used to create the database. It is is broken into 4 sections for each primary neuron activity type: A) silent, B) spiking, C) bursting, and D) non-periodic (meaning there is no clear repetition of specific behavior).

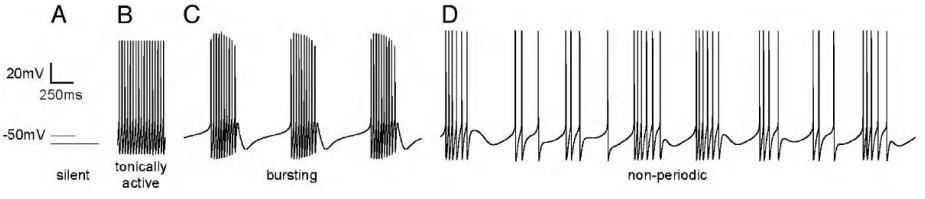


Fig. 1. 4 types of neuron activity

The model neuron used in the simulator consisted of 8 conductances, each of which could take on 6 possible values. The specific conductance parameters were: H, CaS, Na, CaT, leak, KCa, Kd, and A. Every possible combination of the 8 conductance parameter values was evaluated yielding 1,679,616 model neuron simulations. For each simulation, the parameter values were concatenated to form an 8 digit base 6 number, converted to decimal, and used as the primary key for that simulation in every table of the database. For instance, the row with primary key 322209 refers to a model neuron simulation with the parameter values 1,0,5,2,3,4,1, and 3. The other columns of the database were used for various attributes such as the number of spikes in a simulation voltage plot. Details of the simulator and model neuron parameters are available in [4].

It is important to note the delineation between the conductance parameter values which served as the dimensions of our data set, and the other characteristics of each simulation which were held in other database columns. The visualization method we employed exploits this distinction whereas many others view each column of a database as a dimension. This is discussed further in Section 5 along with an analysis of the general applicability and limitations of our approach. The next section discusses related research and the considerations involved in specifying our visualization method.

3 Related Research

Because there was no established technique for visualizing the neuroscience data under investigation, a first effort was to try and fit any visualization to the database. There are a number of methods for multidimensional and/or multivariate visualization [5] [6] [7], [8]. Most, however, could not address the size of the neuroscience database. One common problem was that a glyph representing one data point would obscure another because of the total number of data points being represented. For instance, trying to display 1,679,616 neuron simulations with parallel coordinates would result in overlapping lines and issues of obfuscation. In addition, most software tools could not load the entire dataset into working memory. For instance, XGobi was unable to load a table from the neuroscience database without freezing.

One option was to perform a random sampling of the dataset to derive a smaller set for visualization. The problem with this is that the database already represented a sparse sampling of neuron conductance space. Another option was to construct a matrix of two dimensional scatter plots, where every combination of two conductance parameters was displayed. As Ward notes in [9], this approach may be relatively insensitive to the size of a dataset but complicates the discovery of data trends involving more than 2 dimensions. The more dimensions there are, the more scatter plots that must appear in the matrix, and the less screen real estate there is for each. This can lead to confusion and inefficient panning and scrolling when cross-referencing more than 2 plots to determine correlations.

One method that has been quite successful at visualizing large amounts of data is called pixelization[2]. It is the simple yet powerful technique of mapping every data point of some set to a pixel in a two dimensional image. Keim has shown the effective use of pixelization in visualizing large datasets [10] and prescribes design imperatives for its application in [2]. Keim's software tool VisDB implements pixelization and maps each dimension to a different window [11]. Layout is determined by space filling curves and is the same for each. To determine correlations between dimensions, users compare collocated regions in the windows for each dimension of interest. The semantics of Keim's layout are that data items with similar values or "distances" from user queries are placed next to each other, however their specific location in an image does not impart any information.

It was our preference to let dimensions be assigned to the axes of a pixelization and thus determine the layout. This allows users to identify the specific dimension values for any given data point or pixel based on its location in the image. The color of a pixel can then convey other information such as the result of a user query. One favorable result is that only one frame is used for visualizing the entire database. Users can thus determine data trends without having to toggle between windows or scroll between scatter plots. This also allows users to retain the context of the entire database as they investigate various trends, leading to a deeper understanding of the underlying structure of the data.

To assign the 8 dimensions of our data to the 2 dimensions of a pixelization image, we employed the technique of Leblanc et. al. called dimensional stacking [3]. This approach was implemented in Ward's XmdvTool and is the technique

Polaris used when extended to support visualization of hierarchical data [12]. To use the terminologies of hierarchical databases, the conductance parameters used in the simulations of our database were considered independent variables and seen as dimensions of the model neuron conductance space. The recorded characteristics of each simulation were considered dependent variables or "measures".

4 Dimensionally Stacked Pixelization

The general concept of dimensional stacking is to embed the values of one dimension within each value of another. The origin of our dimensional stacking images is in the bottom left corner. Dimension values increase going upwards and to the right. For readability, only the 2 outer dimensions are numerically labeled in Figures 3 and 4. Figure 3 shows a dimensionally stacked pixelization image where the order of dimensions on the X axis is CaS, A, Na, and H and the order of dimensions on the Y axis is CaT, leak, Kd, and KCa. On the X axis, every value of A is embedded within each value of CaS, every value of Na is embedded within each value of A, and every value of H is embedded within each value of Na. The Y axis follows a similar pattern.

The colors in Figures 3 and 4 represent resting potential of silent neurons, the specifics of which are out of the scope of this paper. The images only look different because each has a different order of dimensions (this is discussed further below). In Figure 4 the order of dimensions on the X axis is H, CaS, Na, and CaT and the order of dimensions on the Y axis is leak, KCa, Kd, and A. Each color represents a different subset of the database as defined by the queries in Figure 2. For instance, the last query in Figure 2 which is associated with the color red, selects neurons with a resting potential between -80 mV and -73 mV. In Figure 4 the red pixels are clustered in the bottom left of the image where H and leak are 0. We can therefore deduce that low values of H and leak in a neuron lead to it having a resting potential between -80 mV and -73 mV.

Query	Color
select model from steadyclass where activity = 0 and p>= .081 and p<=110;	
select model from steadyclass where activity = 0 and .081 > p and p>=.041;	
select model from steadyclass where activity = 0 and .041 > p and p>=-.017;	
select model from steadyclass where activity = 0 and -.017 > p and p>=-.041;	
select model from steadyclass where activity = 0 and -.041 > p and p>=-.061;	
select model from steadyclass where activity = 0 and -.061 > p and p>=-.072;	
select model from steadyclass where activity = 0 and -.072 > p and p>=-.081;	

Fig. 2. ColorMapper

A formal bijection was authored to map between pixel locations and dimension values. The 8 conductance parameters were partitioned into 2 groups of 4 and viewed as a pair of 4 digit, base 6 numbers (since each conductance could take on 6 possible values). These numbers were converted to decimal and served as pixel coordinates. For Figure 3, pixel coordinates were determined from the dimensions as follows:

$$X = CaS * 6^3 + A * 6^2 + Na * 6^1 + H * 6^0$$
$$Y = CaT * 6^3 + leak * 6^2 + Kd * 6^1 + KCa * 6^0$$

We refer to the dimension associated with the most significant digit in one of the pixels' decimal coordinates as the most significant dimension for its axis. In Figure 3 the most significant dimension on the X axis is CaS and the most significant dimension on the Y axis is CaT. Each of the 6^8 model neurons were mapped to one pixel in an image. By using mixed-radix computation, this technique of deriving pixel coordinates can be extended to variable numbers of dimensions with different possible values. This is discussed further in Section 5.

One of our primary extensions to the original methods of pixelization and dimensional stacking was to allow image colors to be specified directly by SQL queries. The queries could return a subset of the database or a set of decimal values which were the result of running some function on select columns of the database. Subsets were mapped directly to one color and the pixels associated with each model neuron contained in the subset were colored appropriately. Decimal values were normalized and mapped to a color gradient. The pixel associated with each model neuron was then colored with the appropriate color gradient given its result for the SQL function. The color map for Figures 3 and 4 is shown in Figure 2 and contains only queries that return subsets of the database.

4.1 Dimension Order

The order of dimensions has a significant effect on the resulting image. Both Figures 3 and 4 show the same query results with the same color map; the only difference is the order of dimensions. Typically, data trends determined by less significant dimensions will appear as a repeating pattern throughout an image. An example of this are the repeating red vertical lines in Figure 3 that appear where $H = 0$ and $leak = 0$. Data trends determined by more significant dimensions appear as visual clusters. Such is the case in Figure 4 where both leak and H have been switched to the most significant dimensions on the X and Y axis, effectively clustering the red pixels in the lower left corner where both conductances are 0.

Because we had 8 dimensions there were 8! possible dimension orderings and $8!/2$ or approximately 20,000 possible images. The division by 2 results from extracting identity flips, i.e. swapping each pixel's x and y coordinates effectively reflects a square image along a diagonal. Ankerst proposes clustering dimensions with similar attribute values [13], however, with the combination of pixelization and dimensional stacking there is a difference between clustering dimension values and clustering the pixels in an image. Within our visualization scheme, colors are specified not by dimension values but by other data attributes such as the simulation characteristics that were functionally dependent on the conductance values of our model neurons.

Much of the previous work on dimension reordering has focused on clutter reduction or making a visualization more discernible [14][15]. We have found that there is not necessarily an optimal dimension order in a dimensionally stacked

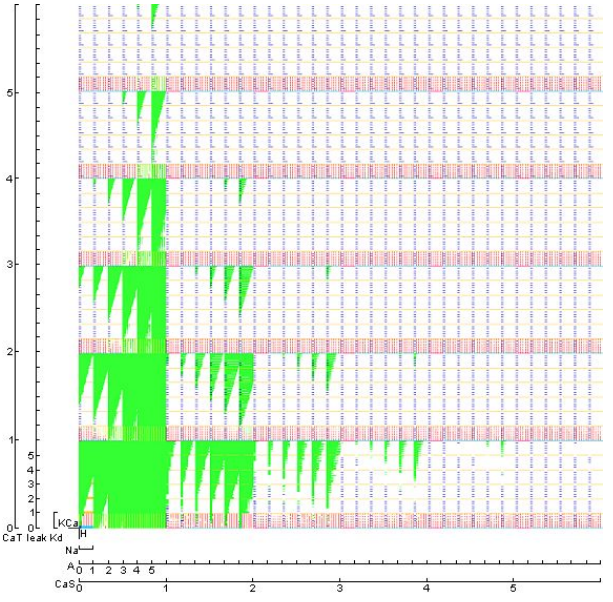


Fig. 3. Dimensionally stacked pixelization image showing resting potential of silent neurons with the dimension order: CaS, A, Na, H on the X axis and CaT, leak, Kd, and KCa on the Y axis

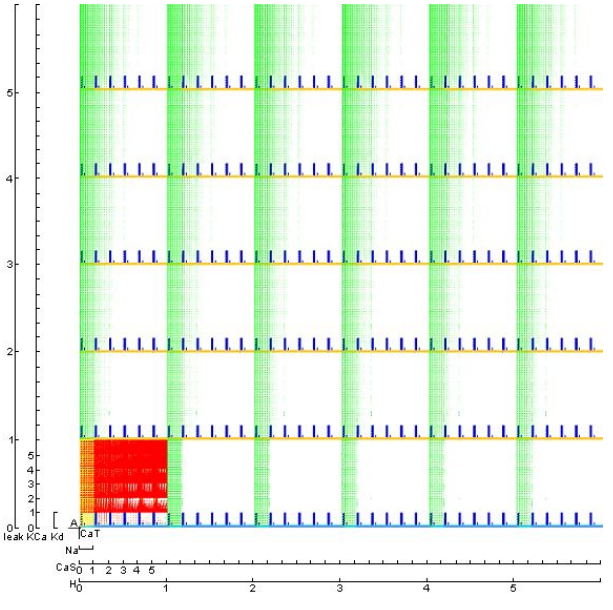


Fig. 4. Dimensionally stacked pixelization image showing resting potential of silent neurons with the dimension order H, CaS, Na, CaT on the X axis and leak, KCa, Kd, A on the Y axis

pixelization; different dimension orders simply reveal different information. Using the dimension ordering in Figure 3, investigators were able to derive a linear constraint for the green pixels (associated with data points returned by the 5th query in Figure 2):

$6 * Leak + 6 * A + Kd + KCa >= 2 + 24 * CaS + 10 * CaT + 3 * Na + H$

Using the dimension ordering in Figure 4, investigators were able to derive linear constraints for the red pixels:

$H < 1, Leak < 1$

By visualizing different dimension orders, users can visually mine data and get a better understanding of the parameter space. This can reveal not only data trends, but what dimensions play a central role in creating them. We have utilized some automatic clustering techniques to assist the user in finding informative dimension orders. These have proven quite useful and is one area for future research.

5 Generalization

5.1 Limitations on Data Formats

We believe the combination and extensions of pixelization and dimensional stacking can be used to effectively analyze extremely large databases, functional databases (e.g. the results of other simulations such as stock market predictions, weather forecasting, etc.), and hierarchical data (e.g. any data held in an XML database). For relational databases, the application of dimensional stacking and pixelization is limited because it must partition the columns into two types: 1. $<$ 20 dimensions with $<$ 20 possible discrete values and 2. data attributes which are used to color pixels. There is no hard constraint on the number of dimensions or the values that they can possess, however, the size of an image grows rapidly as both increase. Attributes should generally be functionally dependent on dimension values for trends to be visible (e.g. for the neuroscience database, neuron activity was considered dependent on conductance values). It is possible to discretize certain columns of a relational database to serve as dimensions, however one must be aware of the possibility of mapping more than one data point to the same pixel. This may not be a problem or may even be desirable if using an approach where images are overlayed or transparency is used. Rows of relational databases can also be aggregated so that a single pixel represents some number of records. We believe our techniques should work especially well with hierarchical data and XML databases. Higher level nodes would simply be mapped to more significant dimensions in one of our pixelization images.

To generally apply our mathematical bijection, consider a database to be a function on the tuple T. T should be a set of n database columns which can be considered dimensions or independent variables and functionally determine the values of the other database columns. The cardinality of possible values each dimension can take on is its base B. When all dimension bases are equal, T can then be viewed as a base B number with n digits. For example, lets suppose $n = 4$ and the dimensions a, b, c, and d take on the values 0 and 1. We can

interpret these dimensions as a 4 digit binary number. Ordering the dimensions in alphabetical order, the number 0101 would map to $a = 0$, $b = 1$, $c = 0$ and $d = 1$ or the decimal number 5. It is possible to have an odd number of dimensions and/or dimensions with unequal bases.

To get a 2D representation we first partition the dimensions into two groups: X and Y. The x coordinate of each pixel is derived from X and y from Y as described above for the neuroscience data and shown in [3]. The width of a dimensionally stacked pixelization is the product of the bases in X, and the height is the product of the bases in Y. Below, we specify the algorithm for deriving dimension values from pixel coordinates (known as mixed radix computation) to complete a bijection between the 2:

1. Given some decimal number m and an order of dimensions, start with the rightmost (or least significant) dimension t_n and iterate left
2. $t_n = m$ modulo (the base of t_n)
3. m = m / (the base of t_n)
4. repeat for t_{n-i} until the value for t_0 is calculated

For the decimal value 7 and 3 dimensions with the bases 2, 3, 2:

1. $7 \; mod \; 2 = 1$ (keep for rightmost parameter = 1)
2. $7/2 = 3$ (ignore remainder pass on as new num)
3. $3 \; mod \; 3 = 0$ (pre-pend to current answer = 01)
4. $3/3 = 1$(ignore remainder pass on as new num)
5. $1 \; mod \; 2 = 1$ (pre-pend to current answer = 101)
6. = 101

6 Conclusion

In this paper we have presented a combination of dimensional stacking and pixelization that uniquely facilitates visual data mining of multidimensional, multivariate data. We extended the original methods in a number of ways such as providing a mathematical formalization and bijection between dimension values and pixel coordinates and specifying image colors with SQL queries. In addition, the queries supported by our color maps are more flexible than those allowed by most other tools and techniques. We discussed the utility of dimension ordering which can be used in concord with the color map queries to draw out data trends and inform exploratory analysis. We then presented a generalization of our methods and how they can be applied within other data domains.

One of the unique properties of our approach is the amount of data that can be visualized at one time. Because each database record is mapped to one pixel and every pixel is utilized in the layout, the only real limitation is the number of pixels on the user's display. This proved quite valuable in allowing neuroscientists to visualize an entire database of 1,679,616 model neuron simulations in one image.

Images produced from our techniques have now appeared in the Journal of Neurophysiology [1] and are used for educational purposes in neuroscience classes

at Emory University. The general use pattern of investigators is to: 1. perform a statistical analysis of a database to motivate hypothesis and queries, 2. use queries to specify a color map in a dimensionally stacked image, 3. permute the dimension order of the image to cluster and draw out data trends, 4. revise queries and/or hypothesis. Automatic methods have been used for finding informative dimension orderings, however, different dimension orders reveal different information. Thus, there is not necessarily an optimal dimension order for any particular set of queries. We have found that different dimension orders can be used to motivate different hypothesis about the underlying data.

The structure of the data we were tasked to visualize lent itself well to the combination of dimensional stacking and pixelization. We believe our methods would work best with vast databases, functional data where there is a clear delineation between independent and dependent variables, and / or hierarchical data such as that found in an XML database. There are many directions for future work including testing our methods with different types of datasets and integrating our visualization with established statistical and knowledge finding algorithms.

Acknowledgements

We would like to thank Professor Eve Marder and Dr. Adam L. Taylor of the Brandeis University Biology Department for many helpful discussions and suggestions for the development of our visualization techniques.

References

1. Taylor, A.L., Hickey, T.J., Prinz, A.A., Marder, E.: Structure and visualization of high-dimensional conductance spaces. Journal of Neurophysiology (in press)
2. Keim, D.A.: Designing pixel-oriented visualization techniques: Theory and applications. IEEE Transactions on Visualization and Computer Graphics **6** (2000) 59–78
3. LeBlanc, J., Ward, M.O., Wittels, N.: Exploring n-dimensional databases. In: VIS '90: Proceedings of the 1st conference on Visualization '90, Los Alamitos, CA, USA, IEEE Computer Society Press (1990) 230–237
4. Prinz, A.A., Billimoria, C.P., Marder, E.: An alternative to hand-tuning conductance-based models: Construction and analysis of data bases of model neurons. Journal of Neurophysiology **90** (2003) 3998–4015
5. Andrews, D.F.: Plots of high dimensional data. Biometrics **28** (1972) 125–136
6. Chernoff, H.: The use of faces to represent points in k-dimensional space graphically. Journal of the American Statistical Association **68** (1973) 361–368
7. Feiner, S.K., Beshers, C.: Worlds within worlds: metaphors for exploring n-dimensional virtual worlds. In: UIST '90: Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology, New York, NY, USA, ACM Press (1990) 76–83
8. Joslyn, C., Lewis, C., Domik, B.: Designing glyphs to exploit patterns in multidimensional datasets. In: CHI '95: Conference companion on Human factors in computing systems, New York, NY, USA, ACM Press (1995) 198–199

9. Ward, M.O.: Xmdvtool: integrating multiple methods for visualizing multivariate data. In: VIS '94: Proceedings of the conference on Visualization '94, Los Alamitos, CA, USA, IEEE Computer Society Press (1994) 326–333
10. Keim, D.A.: Visual exploration of large data sets. Commun. ACM **44** (2001) 38–44
11. Keim, D.A., Kriegel, H.P.: Visualization techniques for mining large databases: A comparison. IEEE Transactions on Knowledge and Data Engineering **8** (1996) 923–938
12. Stolte, C., Tang, D., Hanrahan, P.: Query, analysis, and visualization of hierarchically structured data using polaris. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2002) 112–122
13. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization, Washington, DC, USA, IEEE Computer Society (1998) 52
14. Peng, W., Ward, M.O., Rundensteiner, E.A.: Clutter reduction in multidimensional data visualization using dimension reordering. In: INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04), Washington, DC, USA, IEEE Computer Society (2004) 89–96
15. Yang, J., Peng, W., Ward, M.O., Rundensteiner, E.A.: Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In: IEEE Symposium on Information Visualization. (2003) 14

Detection and Characterization of Abnormal Vascular Patterns in Automated Cervical Image Analysis

Wenjing Li and Allen Poirson

STI Medical Systems, 733 Bishop St. Honolulu, Hawaii, 96813, USA
wli@sti-hawaii.com, apoirson@sti-hawaii.com

Abstract. In colposcopy, mosaic and punctation are two major abnormal vessels associated with cervical intraepithelial neoplasia (CIN). Detection and characterization of mosaic and punctation in digital cervical images is a crucial step towards developing a computer-aided diagnosis (CAD) system for cervical cancer screening and diagnosis. This paper presents automated techniques for detection and characterization of mosaic and punctation vessels in cervical images. The techniques are based on iterative morphological operations with various sizes of structural elements, in combination with adaptive thresholding. Information about color, region, and shape properties is used to refine the detection results. The techniques have been applied to clinical data with promising results.

1 Introduction

Cervical cancer is the second most common cancer in women worldwide, with nearly a half-million new cases and over 270,000 deaths annually. Because invasive disease is preceded by pre-malignant cervical intraepithelial neoplasia (CIN), if detected early and treated adequately, cervical cancer can be universally prevented. The atypical epithelium is usually categorized as CIN grades 1, 2, and 3 histologically or low-grade and high-grade squamous intraepithelial lesions (LSIL and HSIL), depending on the severity of the lesions.

Colposcopy is the primary diagnostic method used in the US to detect CIN and cancer, following an abnormal cytological screening (Papanicolaou smear). The purpose of a colposcopic examination is to identify and rank the severity of lesions, so that biopsies representing the highest-grade abnormality can be taken, if necessary. A colposcopic examination involves a systematic visual evaluation of the lower genital tract (cervix, vulva and vagina), with special emphasis on the subjective appearance of metaplastic epithelium comprising the transformation zone on the cervix. To visualize the cervix, a colposcope is used which is a binocular microscope with a built-in white light source. During the exam, a 3-5% acetic acid solution (vinegar) is applied to the cervix, causing abnormal and metaplastic epithelia to turn white (acetowhitening effect). Cervical cancer precursor lesions and invasive cancer exhibit certain distinctly abnormal morphologic features that can be identified by colposcopic examination. Lesion characteristics such as color and opacity, margin demarcation and shape, and vascular patterns are considered by physicians (colposcopists) to derive a

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 627 – 636, 2006.
© Springer-Verlag Berlin Heidelberg 2006

clinical diagnosis. Due to the subjective nature of the examination, the accuracy of colposcopy is highly dependent upon physician experience and expertise. Colposcopy suffers from low specificity leading to many unnecessary biopsies.

Among the colposcopic signs that identify pre-cancerous regions, the abnormal vascular patterns called punctation and mosaic are major diagnostic features [1]. Punctation is a colposcopic finding reflecting the capillaries in the stromal papillae that are seen end-on and penetrate the epithelium. When the stroma and accompanying capillaries are "pressed" between islands of squamous epithelium in a continuous fashion, a honeycomb or chicken-wire pattern called mosaic is produced. Punctation and mosaic can be seen in both normal and abnormal cervical epithelium. Abnormal vessels can be enhanced during a colposcopic exam by using green-filtered light. If the punctation or mosaic is not located in a field of acetowhite epithelium, it is unlikely to be associated with CIN. The punctation or mosaic pattern is described as fine or coarse. If the vessels are fine in caliber, regular, and located close together, it is more likely that the patterns represents a benign process or low-grade CIN. If the intercapillary distance of the vessels is increased and the vessels are coarser in appearance, the grade of the lesion is usually more severe.

Digital imaging is revolutionizing medical imaging and enabling sophisticated computer programs to assist the physicians with Computer-Aided Diagnosis (CAD). Detection of abnormal vascular patterns has been a very challenging task due to several factors such as lack of good quality, high resolution images, glare and mucus effects, non-uniform illumination, large intra-patient variation, and other artifacts on the surface of a cervix. One pilot study [2] uses a generalized statistical texture analysis technique to characterize six types of cervical vascular patterns. The approach in this study represents the vascular structure using line segments and constructs the statistical distribution of the line segments. First and second order statistics derived from the joint and/or marginal distribution are used as textural measures for cervical lesion classification. However, in the experiments, the region of interest is pre-marked and automatic detection of vessels is not addressed. In [3], preliminary results are presented on detecting mosaic patterns using both color and geometric features. The method used seems to over-detect the vascular structure on both normal cervix region, and in the acetowhite regions. Besides, punctation vessel detection has not been addressed. To our knowledge, a robust method to detect the abnormal vascular patterns in cervical images automatically and effectively has not been achieved.

In this paper, we present our work on automatically detecting and characterizing mosaic and punctation vessels in digital cervical images. First, the abnormal region is detected by locating the acetowhite region (the area turning white after application of acetic acid) of the image. Second, a novel mosaic detection technique is presented, which includes mathematical morphology with a rotating line structure element of multiple sizes, branch point detection, and a vascular structure tracing. Third, the punctation detection technique is introduced by applying a "hole" enhancement procedure, morphological transforms with a multiple-size disk structure element, followed by a refinement procedure using gradient and shape information. The properties of the

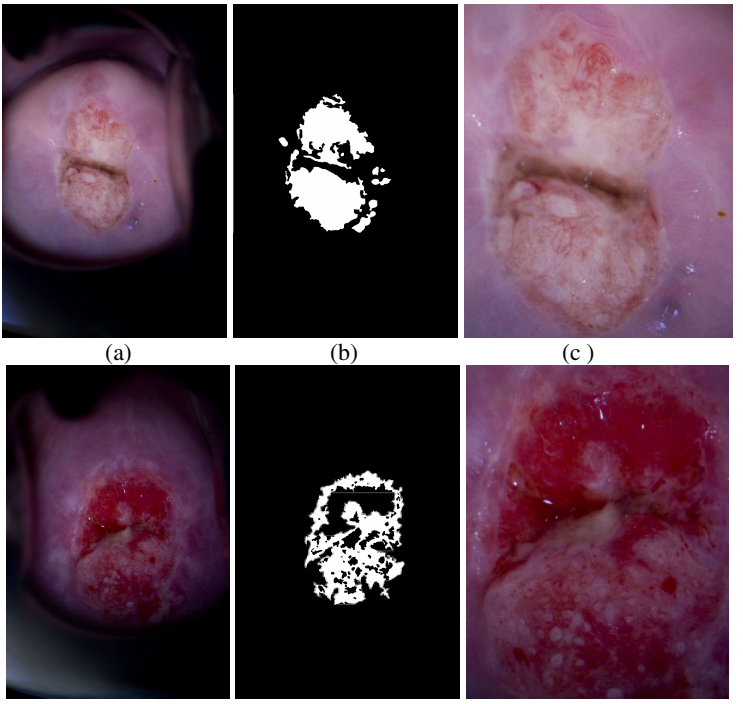


Fig. 1. Preprocessing steps to find the region of interest for vascular detection, (a), (d) Original images, (b), (e) Abnormal regions detected, (c), (f) The regions of interest for vessel detection

mosaic and punctation are also analyzed based on the clinical phenomenology. Experimental results demonstrate the effectiveness of the techniques. Discussions and conclusions are given finally.

2 Abnormal Region Detection

According to colposcopy, acetowhite epithelium refers to epithelium that transiently changes color from pink or red to white after application of 3% to 5% acetic acid. It is the most common colposcopic feature observed when cervical intraepithelial neoplasia is present. We have developed an automatic segmentation of acetowhite region detection algorithm based on color and texture information. The algorithms use multiple unsupervised segmentation techniques to address different anatomic features and different levels of acetowhite lesions. The details can be found in [4]. When mosaic and punctation are not located in the acetowhite epithelium, they are unlikely to associate with CIN. Therefore, we focus on the detection and characterization of the mosaic and punctation within the acetowhite epithelium. Fig. 1 (a) and (d) shows the original cervical images. Fig. 1 (b) and (e) shows the corresponding acetowhite

regions. The region of interest for abnormal vessel detection is the rectangular region for the maximal area of the acetowhite lesion, as shown in Fig. 1(c) and (f).

3 Mosaic Analysis

3.1 Mosaic Detection

Mosaic vasculature appears colposcopically as a red tile-like polygonal grid viewed within an area of acetowhite epithelium. The small blocks of epithelium, encompassed by the mosaic vessels vary in size, shape, and uniformity. The intercapillary distances between mosaic vessels vary depending on the severity of neoplasia. In general, as vessel caliber and intercapillary distance increase, the severity of neoplasia also increases. Based on the phenomenology of the mosaic pattern, we propose a robust abnormal mosaic detection algorithm for cervical images. Taking the acetowhite area of the cervical image as input, the method consists of the steps outlined below.

a. Iterative detection of vascular structure. In 1993, Thackray **[5]** presented a semi-automatic segmentation method for vascular network images using a combination of mathematical morphological opening with a linear rotating structuring element (ROSE). The method has proved to be effective on initial segmentation of dominant vascular networks containing vessels of different size and orientations. However, in cervical images, the mosaic pattern is only a subset of vascular structures. The dominant texture region also covers anatomy of the cervix and artifacts on the surface. Therefore, we adapted this theory in combination with an adaptive thresholding and a thinning procedure to do the initial segmentation. In each iteration, a top hat transform is performed using the line structure element at one particular direction. An average spatial distance metric is computed each time until it reaches a pre-defined threshold. The resulting images are thresholded using an adaptive parameter and then combined with results from the previous iteration. Fig. 2 shows the intermediate results at

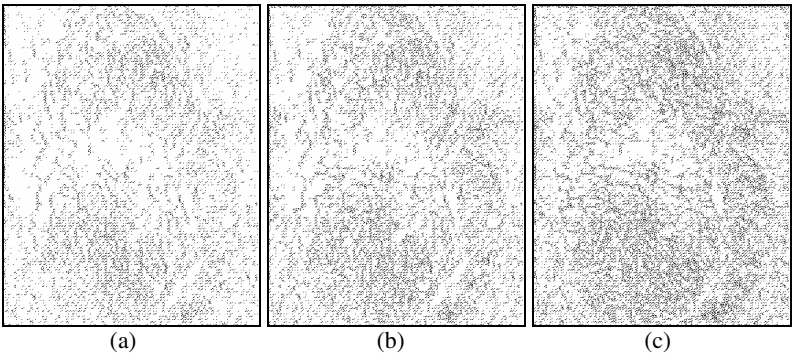
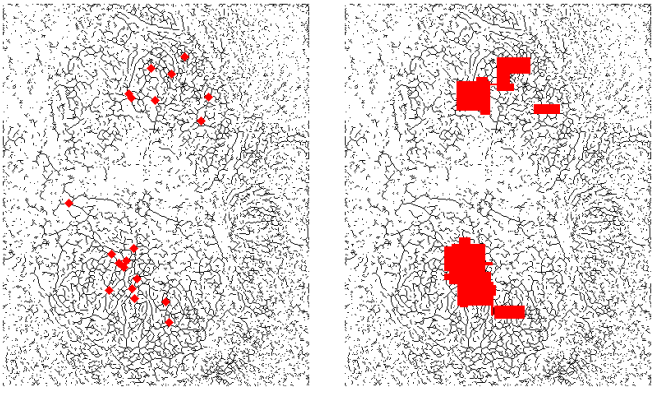


Fig. 2. Vascular Pattern detection (a) 1st iteration, (b) 3rd iteration, (c) final iteration

different iteration steps. It can be seen that after a certain number of iterations, all the fine vascular structures in the image are well captured. However, some short noisy line segments are also included. They will be removed in the next step.

b. Detection of branch points and high probability mosaic regions. The mask image with initial detection results is scanned with a 3 by 3 square window. The positions where the number of white pixels within the square is greater then 3 are marked as initial branch points. The branch points are further refined to remove the false detected ones. Regions with more than two branch points within a window indicate an area with high probability of having a true mosaic vessel pattern. The branch points are used as seeds for vessel tracing the thinned enhanced vessel structure. Fig. 3 shows the branch points and the results of high probability region detection.



(a) (b)

Fig. 3. Detection of (a) branch points and (b) high probability of mosaic regions

c. Tracing of mosaic structures. All branch points within a high probability region in the original vessel structure are used as seeds for tracing and identifying the mosaic patterns. The tracing procedure starts at a branch point and then detects it connected points (vessels) in eight possible directions one by one to locate complete vessel structure. Fig. 4 (a) shows the results for the traced mosaic vessel structure.

d. Mosaic region identification. According to the phenomenology of mosaic vessels in colposcopy, the mosaic regions are extracted as the closed region from the traced vascular structure. Those regions are further refined based on its color properties, surface area, and its intercapillary distance. Since we are only interested in the abnormal area related to the disease severity, the area outside the acetowhite epithelium is eliminated. Fig. 4 (b) shows the detected mosaics and (c) shows the final mosaic vessels after refinement.

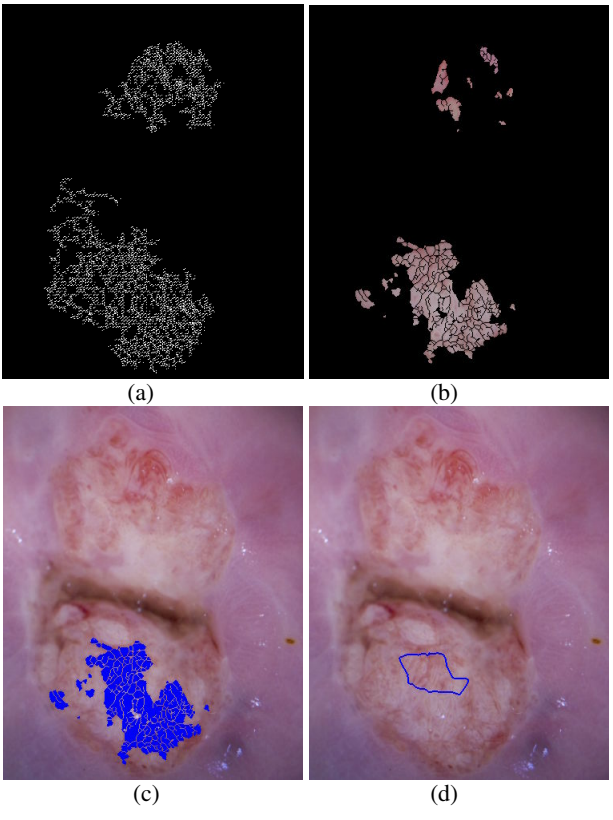


Fig. 4. (a) The traced vessel structures (b) the detected mosaic patterns, (c), the refined mosaic patterns, (d) colposcopic annotation

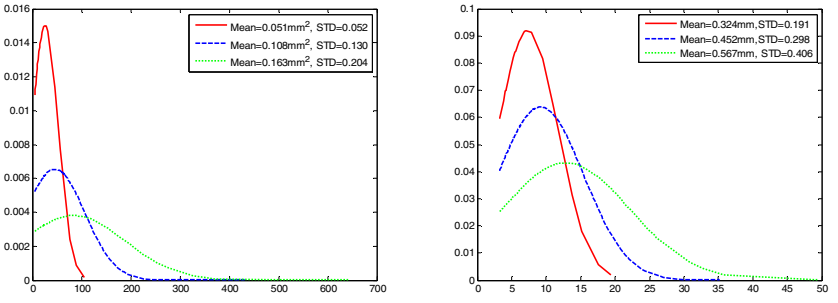
3.2 Mosaic Characterization

Mosaic are categorized colposcopically as fine or coarse, based on their diameter. A fine mosaic pattern is a closely interwoven, lacy, delicate network of capillaries of nearly normal caliber, dispersed perpendicularly in stromal ridges resembling red grouting between small white ceramic tiles. A uniformly consistent small intercapillary distance may be seen with immature metaplasia, a congenital transformation zone, and CIN 1 lesions.

A coarse mosaic pattern is characterized by dilated varicose vessels that enclose larger diameter, irregularly shaped mosaic epithelial blocks. The abnormal coarse vascular pattern is also confined invariably to a well-demarcated, dense acetowhite lesion. The epithelial pegs between the vessels are larger and more varied in shape, reflecting irregularity and an increase in intercapillary distance. A wide, irregular, non-uniform intercapillary distance and coarse-caliber vessels would be typical of a mosaic pattern seen with CIN3. The analyzed mosaic provides insight into the nature of the epithelium being inspected [6]. The mean intercapillary distance between mosaic vessels in CIN2 is significantly less than seen in CIN3. Consideration of the

distribution of a mosaic vessel further helps to determine the type of epithelium observed.

All the properties of the mosaic patterns are recorded including intensity, color, surface area, perimeter, the shortest axis length and the longest axis length. Their distribution is modeled as Gaussian. The standard deviations of the size of the mosaics and their intercapillary distances represent the irregularity of the mosaic patterns. Fig. 5 (a) shows the distributions of surface area for different mosaic regions and (b) shows the corresponding distributions of the intercapillary distances. The intercapillary distance of a mosaic is defined as the distance between the farthest two points in the mosaic. The solid-line curve in the figures is more likely to be fine mosaic since both of the mean and standard deviation are smaller and the dot-line curve in the figure is more likely to be coarse mosaic region. The decision boundary between coarse and fine mosaics is computed from a representative training set with 19 subjects.



(a) (b)

Fig. 5. Characterization of mosaics (a) Surface area distributions of different mosaic regions (b) Intercapillary distance distributions of different mosaic regions

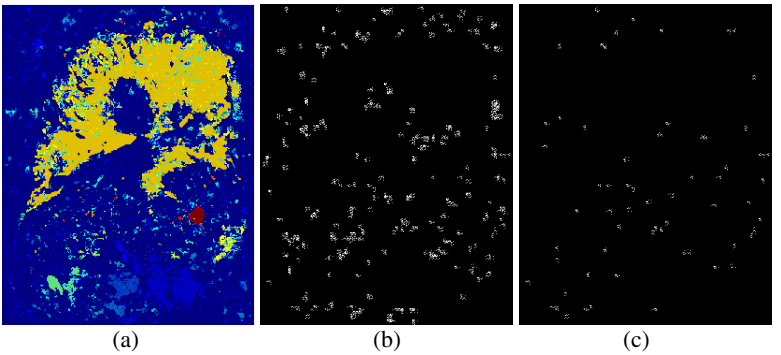
4 Punctation Analysis

4.1 Punctation Detection

Punctation appears as tiny red dots of variable dimensions usually present within an area of acetowhite epithelium. Both the size or diameter of punctation (caliber of the loop vessels) and the distance between punctation (intercapillary distance) vary depending on the severity of the underlying disease. Punctation detection has not yet been addressed in the literature. Since the punctation patterns appear very small in the cervical images, the quality of the image has a major impact on the detection of the punctation vessels. Poor focus, narrow depth of field, and non-uniform illumination are all obstacles to an effective detection.

Our initial punctation detection technique employs the same strategy as the mosaic detection. Instead of rotating line structure elements, disk structure elements with various sizes have been used for the top hat transforms. Due to insufficient image quality, we designed a pre-processing procedure to enhance the punctation structure.

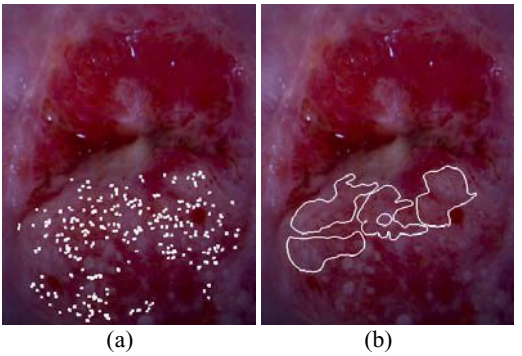
The enhancement procedure consists of a series of morphological operations, including extraction of a green band image, a "hole" closing operation, and a contrast enhancement technique with a toggle operator. The technique highlights the hole-like vascular structures in the image, which is a crucial pre-processing step for the punctation detection. An example of the "hole" enhancement image can be seen in Fig. 6(a).



(a) (b) (c)

Fig. 6. (a) Enhancement of "holes" in the image (b) the punctation mask before the refinement (third iteration) (c) the punctation mask after the refinement (third iteration)

The initial result of punctation detection is usually noisy and has many false positives. A post-process procedure is required to refine the results. Various region properties are computed, including the local gradient information, region area, and the eccentricity of the regions. Gradients and region area use heuristic rules based on the imaging distance and clinical prior knowledge. The eccentricity of the region has a high impact removing the false detected regions. Fig. 6 (b) shows the punctation mask before the refinement in a single iteration, and Fig. 6 (c) shows the corresponding punctation mask after the refinement procedure. Finally, the detected region is further refined by excluding the non-acetowhite regions, glint regions, and other anatomic regions (Os, and columnar epithelium), as seen in Fig. 7(a).



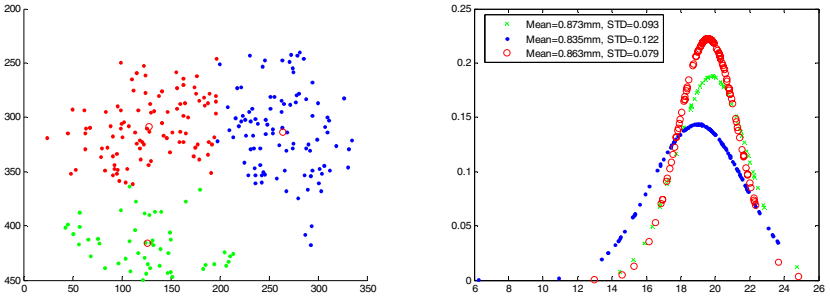
(a) (b)

Fig. 7. Final Punctation detection and it colposcopic annotation (a) the punctation detection results, (b) colposcopic annotation

4.2 Punctation Characterization

Fine punctation is a regular pattern of looped capillaries of narrow diameter, usually closely and uniformly spaced. When fine punctation is confined to an abnormal acetowhite lesion within a field of immature metaplasia, it is usually indicative of a low-grade lesion (CIN1). When fine punctation is confined to an abnormal acetowhite lesion on the original squamous epithelium, it may represent either a HPV-induced lesion or a variation of the normal metaplastic process. In coarse punctation, the capillaries appear more pronounced because the loop capillaries are dilated and the intercapillary distance is greater. Furthermore, coarse punctation is more irregularly spaced. In CIN3 lesions, 57% of vessels exhibit intercapillary distance of more than 300um [7]. A vascular pattern of coarse punctation usually indicates a high-grade lesion, or CIN2, 3, and possibly early invasion.

In order to assess the severity of the lesion, the detected punctations are further classified as different clusters according to their spatial distribution using an EM (Expectation Maximization) algorithm. In each cluster, properties such as the local area, local gradient, and distance matrix are recorded. The intercapillary distance of the punctation is calculated as the average distance of the neighboring punctations to the corresponding point. The neighborhood range is based on a pre-defined window. Fig. 8 (a) shows three clusters of the punctation vessel and Fig. 8 (b) shows the distribution of the intercapillary distances. The similarity of distribution of intercapillary distances indicates a similar type of punctation, coarse in this case. As with mosaic characterization, the decision boundary between the coarse and fine punctations is determined from a representative training set with 30 subjects.



(a) (b)

Fig. 8. Clustering the punctation data (a) the three clusters (b) their corresponding distributions of intercapillary distances

5 Discussions and Conclusions

Using our digital colposcope, we acquired 100 high-quality, cross-polarized image sets from hospitals in Peru. The mosaic and punctation detection results are being evaluated by expert colposcopists. Histopathology results from LEEP (Loop Electrosurgical Excision Procedure) are also available to provide “gold standard” for disease

severity. Fig. 4 (d) and Fig. 7 (d) show two examples of the colposcopic annotations on mosaic and punctation regions. The subjects shown in this paper have been verified by histopathology as having HSIL. The quantitative analysis of the performance will be done soon using a region based match around the cervical os region.

A CAD system for colposcopy has a direct impact on improving women's health care. Detection and characterization of the abnormal vessels are one crucial step in developing the CAD system. This paper presents a novel application of image processing technique in cervical cancer screening and diagnosis. The vessel detection techniques presented here are fast and robust. The computed features of abnormal vessels will serve as inputs to a tissue classification routine, together with other diagnostic features like color, opacity, shape, and demarcation. The results of the classification routine will provide the physician the suggested diagnosis and biopsy sites.

Acknowledgements

The authors would like to thank Daron Ferris (MD) for the colposcopic annotation, Johan Hakansson for his early development work, and Marcelo Soto-Thompson, John Hargrove, and Ulf Gustafsson for the comments and discussions.

References

1. Reid, R. and Scalzi, P.: Genital warts and cervical cancer. VII. An improved colposcopic index for differentiating benign papillomaviral infections from high-grade cervical intraepithelial neoplasia. Am. J. Obstet. Gynecol. 153 (6) (1985) 611-618
2. Ji, Q., Engel, J., and Craine, E.: Texture analysis for classification of cervix lesions. IEEE Trans. Med. Imaging 19 (11) (2000) 1144-1149
3. Srinivasan, Y., Hernes, D., Tulpule, B., Yang, S. et al.: A probabilistic approach to segmentation and classification of neoplasia in uterine cervix images using color and geometric features, in *Proc.of SPIE Medical Imaging,* **5747**. (2005) 995-1003
4. Li, W.: Automated image analysis of uterine cervical images, in *Proc.of SPIE Medical Imaging, submitted,* (2007)
5. Thackray, B.D. and Nelson, A.C.: Semi-automatic segmentation of vascular network images using a rotating structuring element (ROSE) with mathematical morphology and dual feature thresholding. IEEE Trans. Med. Imaging 12 (3) (1993) 385-392
6. Mikhail, M.S. and Romney, S.L.: Computerized Measurement of Intercapillary Distance using Image Analysis in Women with Cervical Intraepithelial Neoplasia: Correlation with Severity. Obstet. Gynecol. 95 (4 Supplement) (2002) 2S-3S
7. Ferris, D., Cox, J.T., O'Connor, D.M., Wright, V.C., and Foerster, J.: Modern Colposcopy, Textbook and Atlas. Second. American Society for Colposcopy and Cervical Pathology, (2004)

Towards a Modular Network-Distributed Mixed-Reality Learning Space System

Timothy J. Rogers, Bedřich Beneš, and Gary R. Bertoline

Purdue University, West Lafayette 47907, USA

Abstract. We propose a Modular Mixed-Reality Learning Space System (MRLSS) that relies on middleware tools and distributed mixed-reality technologies to support multi-modal communications between local, remote, and virtual *audience sets*. Each audience set occupies a *spatial module* represented throughout the system as a *cell* which is visually displayed on specifically aligned projection surfaces in each module. A module can host multiple cells and can be categorized based on scalability and technical advantage. For example, an *Individual Cell* (ICell) could contain a participant with only a web cam and audio. A *Classroom Cell* (CCell) could be a single classroom. A *Virtual Cell* (VCell) is a graphically rendered space with unique possibilities for interaction, experience, and exploration. A *Studio Cell* (SCell) is a specialized facility with advanced systems, services, and scalable spatial capabilities. A *University Cell* (UCell) can host multiple instances of an MRLSS, i.e. simultaneously host and combine more than one MRLSS.

1 Introduction

In *Planning for Neomillenial Learning Styles*, Dede suggests that 21st century students are well accustomed to "the growing prevalence of interfaces to virtual environments and augmented realities." He also notes that students are becoming well versed in new interactive media as well as becoming increasingly immersed in "media-based lifestyles." As a result, the student culture is changing and new styles of learning are emerging. "In the long run," Dede argues, "the mission and structure of higher education might change due to the influence of these new interactive media." [1]

In addition to changes in contemporary learning styles, advances in powerful computing platforms and high-throughput network infrastructures are enabling a broad range of computer-mediated communications(CMC) across data-rich interactive networks such as the Internet. The unique capabilities and popularity of these Information and Communication Technologies (ICTs) are driving major shifts in *how* the education process is orchestrated. Academic operations depend heavily on ICTs, particularly in areas related to accessing campus computer resources, developing course management systems, and managing distance learning courses.

Many traditional classrooms and learning spaces have also transformed in order to get "wired" for online access and to accommodate advanced presentational

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 637–646, 2006.
© Springer-Verlag Berlin Heidelberg 2006

methods. The resulting systems have engendered a lively environment for practical innovations in the design and experimentation of advanced learning spaces while simultaneously supporting new pedagogical techniques. [2,3,4,5,6,7,8]

Like Dede, many educators recognize this technological shift as a valuable opportunity to leverage network-centric educational tools and computer-mediated experiences in order to manage and enhance the learning process. [9,7].

We introduce a conceptual framework for a scalable and modular Mixed Reality Learning Space System (MRLSS) and illustrate its unique design. The system is intended to engender modular and scalable presentational spaces that combine and align multiple viewing volumes into comprehensive spaces for learning, experience, and presentation.

The concepts and techniques associated with distributed mixed reality (MR) in presentational environments are central to the development of our proposed MRLSS system. For example, tessellated mixed reality boundaries (MRBs) are incorporated into the MRLSS using a scalable hexa-cellular layout and configuration scheme.

Three distinct viewing space categories, or *audience sets*, are identified: local, remote, and virtual. Each audience set has inherent tradeoffs in the context of the MRLSS and a major research goal is to identify and manipulate critical factors related to how learning outcomes change as critical presentational materials migrate across boundaries.

The primary education-related goal of this network-distributed system is to effectively synthesize key advantages of both traditional and network-centric pedagogical forms. Special attention is given to shaping the unique features of the MRLSS in order to accommodate the particular needs of neomillennial learners and educators.

2 Related Work

The MRLSS framework is intended to establish methods and standards for orchestrating the synchronous and poly-synchronous confluence of co-located learners, educators, and researchers into advanced learning spaces. The proposed framework is a synthesis of a variety of emerging concepts and technologies related to mixed reality, middleware-enabled open architecture, and synchronous collaborative platforms.

2.1 Mixed Reality

Beginning roughly in the early 1990's, the coupling of real-time 3D computer graphics capabilities with their associated displays and interface devices have inspired novel possibilities for enhanced mediated experiences. Some of these highly interactive experiences involve the merging of shared virtual environments with real-time processing of live video and audio streams. This technique is generally known as *mixed reality*.

The term "mixed reality" (MR) first appeared in the literature in 1994 when Milgram and Kishino published a taxonomy and continuum in order to identify

an emerging subclass of virtual reality technology called Augmented Reality (AR). The authors identified MR as "...the merging of real and virtual worlds" such that "real world and virtual world objects are presented together within a single display" [10].

Starting in the late 1990s, the Communications Research Group(CRG) at the University of Nottingham generated a number of projects and papers related to the applied use of collaborative virtual environments (CVEs). While investigating the shared dynamics across physical and synthetic spaces, the Communications Research Group conducted several experiments within the context of live performance and presentational environments.[11] One series of experiments known as "Inhabited Television" merged CVEs with broadcast television and a live theatre audience. [12,13,14,15,16,17]

One of the most unique research areas to emerge from the CRG was the introduction of the mixed reality boundary (MRB).

Tessalated Mixed-Reality Boundaries.

MRBs were proposed by Benford et al. as a novel approach for "creating transparent boundaries between physical and synthetic spaces."[18]

MRBs demonstrated that MR could be extended beyond just what happened within any given viewer-oriented display. "Thus, instead of being superimposed, two spaces are placed adjacent to one another and then stitched together by creating a 'window' between them." [18,19]

In an eRENA report entitled "Pushing Mixed Reality Boundaries", the authors note that "multiple boundaries might be used to join together many different physical and virtual spaces into a larger integrated structure called a tessellated mixed reality." [20].

The modular system presented in this paper incorporates the tesselated mixed reality technique using hexa-cellular arrangements of distributed mixed reality boundaries. This unique, scalable, design takes full advantage of the inherent mathematic properties of regular hexagons.

2.2 Network and Middleware Modeling

While describing a middleware infrastructure for building ubiquitous mixed-reality applications, Tokunaga et. al [21] state that two key requirements must be satisfied in order to effectively build ubiquitous mixed-reality applications: High-level abstraction to hide heterogeneity of multiple device requirements; and a middleware infrastructure to cope with environmental changes via automatic reconfiguration.

The infrastructure they present, MiRAGe, consists of a *multimedia framework*, a *communication infrastructure*, and an *application composer* that reportedly "hides all the complexities to build mixed reality applications for ubiquitous computing."

The MRLSS is targeted towards accommodating multiple instances of co-located heterogenous users. Unique capabilities of middleware (such as those identified by Tokunaga et. al) will be leveraged for managing and enabling necessary levels of abstraction for the system.

2.3 Designing High Performance Presentational Spaces for Learning and Engagement

Any real-time presentational event that has the power to attract a significant number of participants and/or spectators may benefit by incorporating unique models of networked mixed realities and multimedia into the event space. This is particularly true in research universities with high enrollment. Universities are founded on physical presentational spaces ranging from media-rich cubicles, to large lecture halls. Many universities have a main proscenium theatre with many other alternative performance spaces. Classrooms, however, typically occupy most of the "educational real estate".

Today traditional classrooms are typically equipped with several multi-media options. Collaborative multi-media forms that give advanced controls to individual users greatly increase the cost and complexity of presentational forms such as lectures and research meetings. Previous work related to concepts associated with the MRLSS focused primarily on very small groups in very small, focused, shared spaces. Five examples in particular make use of the *table* metaphor as a familiar gathering point for remotely sharing a space.

In *Extending Tabletops to Support Flexible Collaborative Interactions*, Rogers et. al, demonstrate an interactive tabletop, though all participants are in the same room and gathered around the table.[22]

In *The ARTHUR System: An Augmented Round Table*, Moeslund et. al employ augmented reality to present 3D objects and data onto another shared table concept while incorporating a wireless wand, tangible interfaces, and gesture-based hand and pointing commands within a collaborative context. [23,24]

Regenbrecht et. al, published two papers related to shared spaces that used mixed reality technology. In *MagicMeeting: A Collaborative Tangible Augmented Reality System*, [25] real meeting locations were augmented by incorporating an "Augmented Reality Meeting Table". Again, in *Using Augmented Virtuality for Remote Collaboration*, the cAR/PE! system demonstrated communications within a full virtual environment and generally centering the experience around a virtual table. Live video planes of participants at three different locations were arranged around a table. The cAR/PE! system also allowed for scaling the number of participants around the table. [26]

The National Institute of Standards and Technology (NIST), an agency of the US Department of Commerce, headed the NIST Smart Space Project. The Smart Space Project focused on "Human Information Interaction that transcends the desktop." The project was primarily targeted for US military uses [27]. However, an illustration of a commercial application entitled "Collaborative Design Space" was shown in a NIST presentation related to the project[28]. The example also incorporates the *table* metaphor similar to the cAR/PE project. The focal point, again, being positioned towards an abstracted *table* area for sharing aligned spaces and advanced services.

All of these examples inform the concept proposed in this paper and set the stage for extending the table metaphor into a hexa-cellular arena theatre metaphor.

3 A Modular Mixed-Reality Learning Space System (MRLSS)

The conceptual basis of the MRLSS is built upon the unique properties of scalable interlocking regular hexagons. The hexa-cellular arrangement of mixed reality boundaries (shown via simulation in figures 1, 2 and 3) illustrate how the MRLSS can combine and align multiple instances of local, remote, and virtual audience sets into a shared, comprehensive, learning space. Each audience set occupies a theoretic "spatial module". If each spatial module conforms to a set of operational constraints (such as alignment of boundary cameras and projection surfaces), the hexa-cellular platform concept makes it theoretically simple to parse, switch, and interlock all participating spatial modules. Spatial modules can then be abstracted and represented throughout the system as unique "cells". Each spatial module in the system–depending on its network connection and structural properties–can interlock with one or more cells.

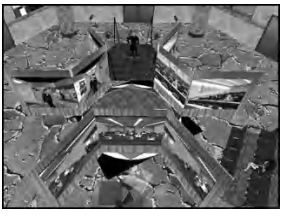


Fig. 1. Overhead view of a full studio simulation with three local/studio cells, a remote cell, and two virtual cells



Fig. 2. A Simulated Example of an MRLSS Modular Learning Space

3.1 MRLSS Cell Framework

MRLSS cells are hierarchically categorized based on the relative scale and technical advantage of each spatial module. This *cell framework* differentiates spatial modules for design purposes in order to isolate and accommodate inherent tradeoffs associated with distinct scenarios. For example, a single participant joining the MRLSS from a coffeehouse with only a web cam and audio headset has many more constraints and limitations than a university researcher presenting visualizations of large scientific data sets in a specialized studio space. A specialized



Fig. 3. Remote Module Point of View(front)

studio would naturally provide many more services including numerous technical and environmental controls.

An Individual Cell, or "ICell", is characterized based on high levels of mobility and/or the use of only a single camera and visual display. An ICell joins the MRLSS via the VCell web interface (see below). ICells are represented as live video avatars in a VCell. Video from a web cam is displayed as a live texture on a geometric quad which is parented to a primary virtual camera. A secondary virtual camera (V-CAM) is positioned and parented facing the quad and tightly framing it for self-viewing and "switched" viewing so that other participants can get close-ups when necessary. The V-CAM can be disengaged from the quad and operated by the participant giving freedom to roam the virtual space and share its viewpoint. The V-CAM also has a 2D graphics overlay for pointing purposes and to display other 2D information.

Classroom-based modules, or "CCells", are constructed in physical learning spaces using specifically anchored audio and video sources arranged to conform to the hexa-cellular platform. The CCell would require a fast intranet/internet connection, wireless access, power outlets, and at least one adjustable camera, semi-permanent projection surface and video projector. A *basic* CCell has only one boundary and is oriented towards taking full advantage of alignment limitations due to permanent, pre-existing, projection systems in typical contemporary classrooms. If the permanent configuration is too far off axis, it is re-classified as an ICell and routed into a VCell (see below). *Flexible* CCells would allow for positional adjustments to projectors and projection surfaces. This allows more freedom to tightly conform to the hexa-cellular platform. Advanced CCells would extend the basic CCell by adding adjacent "wing" MRBs to its primary boundary. Each wing boundary positioned to create a "half-hex" configuration.

A Virtual Cell (VCell) is a graphically rendered space with unique possibilities for interaction, experience, and exploration. (see Figure 4) The VCell includes the ability of ICell participants to login to the virtual cell using a desktop computer, laptop, or other mobile devices that are equipped to interface with the MRLSS (cellphones, PDAs, blackberry's, iPods). VCells can host as many cells as the system is capable of processing and distributing in realtime. VCells also

enable participants to simultaneously occupy more than one spatial module, i.e., a participant with a laptop in a CCell (a nested ICell) could also enter a VCell and, technically, "be" in three cells at the same time. The VCell is unique in that it enables participants to interactively explore advanced levels of content nested within the VCell as well as position themselves to optimally view presentations and participants that occupy other cells.

A VCell is very flexible in that it can be constructed as a classroom, a laboratory, or a simulation of an ancient theatre. Specialized areas within the VCell can be constructed to support multiple visualizations of interactive data sets as well as collections of posters and pre-recorded video displayed on geometry. Previous course content, representative class projects, or collections of seminal research can be spatially archived and accessed within the VCell–perhaps inspiring exploratory visual adventures in research.

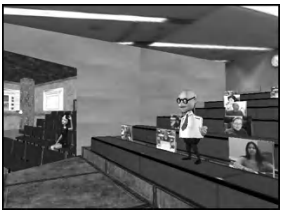


Fig. 4. A "cell wall" feed of a Module looking into a virtual audience set. This set has a 3D animated character and ICells projected on geometry.

A Studio Cell (SCell) is a specialized facility with advanced systems, services, and scalable spatial capabilities that can support modifications to the hexa-cellular platform. SCells are ideal for heavily used learning spaces that may accommodate a wide range of educational content, unique experiences, and special events. The recommendation is that an SCell conform to the hexa-cellular design by architecturally synthesizing it into a specialized arena theatre space. For example, a *full arena* SCell would be capable of hosting up to six additional spatial modules. A *standard* SCell splits the difference spatially between viewing volumes by supporting three local audience sets (offset along 120 degree angles) and three other non-local spatial modules (see Figure 1). Advanced Studio Cells could be constructed out along a hexagonal grid beyond the central SCell. This could potentially enable the creation of complex hexagonal mazes of persistent, interlocking, cell communities.

A University Cell (UCell) can host multiple instances of an MRLSS, i.e. simultaneously host and combine more than one learning space system. Content and course management systems used at universities could interface with the MRLSS for accessing dynamic and time-critical information. Collaborative projects between academic and scientific communities could be fostered to develop shared spaces and remotely view large, sensitive, data sets.

An Alternative Cell (ACell) can be any number of non-standard MRLSS configurations. For example, some classrooms have permanent projection surfaces

that are positioned or sized in ways that make it impossible for the spatial module to visually conform to the hexa-cellular platform. In that case, the ACell must somehow adapt in order to effectively participate. Other forms of ACells may be specialized modules such as full body motion capture volumes, advanced haptic studios, scientific instrumentation labs, live performance spaces, interactive movie/gaming theaters, or even coffee shops with persistent connections to public MRLSS meeting spaces.

By the very nature of the hexa-cellular design, there is no established "center" to the MRLSS. Therefore cell centrism must be negotiated by the participants as to who and where the presentational focus is set. Given the inherent flexibility of VCells and SCells, central focus might orient around one of these types. In a VCell-centric scenario, the emphasis is on the virtual content. In a SCell-centric scenario, the real spaces is centric and other cells gather around the real space. Both have inherent strengths and weaknesses. In a non-centric MRLSS event, many interesting and distributed configurations could be generated–though presentational focus may be sacrificed.

4 Network Structure and Middleware

One of the primary challenges for the proposed MRLSS system will be developing an effective method for the synchronous orchestration of a wide variety of high-bandwidth, real-time, net-centric media objects. We propose a middleware-based collection of show control services or a "show engine" that can be operated, monitored, and controlled by cell-operators through a web portal.

In terms of network traffic, the initial prototype will employ and examine service-oriented designs based on a traffic-per-node or "per cell" basis. Video quality will primarily determine the actual amount of traffic and subsequent system behaviors. H.323 video traffic, for example, will generate low mb/s, whereas HDTV will generate traffic in the gb/s range. The MRLSS assumes a matrix of sites with multiple, heterogenous, data sources. Therefore, in the general case, traffic can increase significantly depending on the number of nodes. However, the use of advanced technologies and algorithms could potentially solve this issue.

In order to address the contextual challenges related to cell interoperability, service tier prototypes will be investigated for identifying, scaling, and filtering real-time media.

5 Discussion

The proposed MRLSS with its projection and camera-based scheme emphasizes conformity to common standards for projection display systems and their associated peripherals–such as screens, lamps, and lenses.

By leveraging the power and flexibility of emerging computer graphics techniques and network computing, traditional time/space constraints between viewer and viewed may soon dissolve within the boundaries of distributed, real-time, modular presentational spaces.

References

1. Dede, C.: Planning for neomillenial learning styles:shifts in students learning style will prompt a shift to active construction of knowledge through mediated immersion. Educause Quarterly **28** (2005)
2. Jamieson, P., Dane, J., Lippman, P.C.: Moving beyond the classroom: Accommodating the changing pedagogy of higher education. In: Refereed Proceedings of 2005 Forum of the Australasian Association for Institutional Research. (2005) 17–23
3. Welker, J., Berardino, L.: Blended learning: Understanding the middle ground between traditional classroom and fully online instruction. Journal of Educational Technology Systems **34** (2005) 33–55
4. Stapleton, C., Hughes, C.E.: Believing is seeing: Cultivating radical media innovations. IEEE Computer Graphics and Applications **26** (2006) 88–93
5. Schwabe, G., Gth, C.: Mobile learning with a mobile game: design and motivational effects. Journal of Computer Assisted Learning **21** (2005) 204–216
6. Oblinger, D.: Leading the transition from classrooms to learning spaces: The convergence of technology, pedagogy, and space can lead to exciting new models of campus interaction. Educause Quarterly **28** (2005) 14–18
7. Salzman, M., Dede, C., Loftin, R., Chen, J.: A model for understanding how virtual reality aids complex conceptual learning. Presence: Teleoperators and Virtual Environments **8** (1999) 293–316
8. Rogers, Y., Scaife, M., Gabrielli, S., Smith, H., Harris, E.: A conceptual framework for mixed reality environments: Designing novel learning activities for young children. Presence **11** (2002) 667–686
9. Chen, J.X.: Learning abstract concepts through interactive playing. Computers and Graphics **30** (2006) 10–19
10. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE (Institute of Electronics, Information and Communication Engineers) Transactions on Information and Systems, Special issue on Networked Reality **E77-D** (1994) 1321–1329
11. Mixed reality laboratory (1997)
12. Benford, S., Greenhalgh, C., Craven, M.: Inhabited television: broadcasting interaction from within collaborative virtual environments. ACM Transactions on Computer-Human Interaction (TOCHI) **7** (2000) p.510–547
13. Craven, M., Benford, S., Greenhalgh, C., Wyver, J., Brazier, C.J., Oldroyd, A., Regan, T.: Ages of avatar: Community building for inhabited television. In: Proceedings of the Third International Conference on Collaborative Virtual Environments. (2000) p 189–194
14. Greenhalgh, C., Benford, S., Taylor, I., Bowers, J., Walker, G., Wyver, J.: Creating a live broadcast from a virtual environment. In: Proceedings of the ACM Siggraph Conference on Computer Graphics. (1999) 375–384
15. Greenhalgh, C., Benford, S., Craven, M.: Patterns of network and user activity in an inhabited television event. ACM Symposium on Virtual Reality Software and Technology, Proceedings (1999) p 34–41
16. Drozd, A., Bowers, J., Benford, S., Greenhalgh, C., Fraser, M.: Collaboratively improvising magic: An approach to managing participation in an on-line drama. In: Proc. ECSCW'01, Bonn, Germany (2001) 159–178
17. Koleva, B., Taylor, I., Benford, S., Fraser, M., Greenhalgh, C., Schndelbach, H., vom Lehn, D., Heath, C., Row-Farr, J., Adams, M.: Orchestrating a mixed reality performance. In: Proceedings of the SIGCHI conference on Human factors in computing systems, Seattle, WA (2001) 38–45

18. Benford, S., Greenhalgh, C., Reynard, G., Brown, C., Koleva, B.: Understanding and constructing shared spaces with mixed-reality boundaries. In: ACM Transactions on Computer-Human Interaction (TOCHI). Volume 5. (1998) 185–223
19. Koleva, B., Schndelbach, H., Benford, S., Greenhalgh, C.: Developing mixed reality boundaries. In: Proceedings of DARE 2000 on Designing aumented reality environments. Designing Aumented Reality Environments, Elsinore, Denmark, ACM Press (2000) 155–156
20. Benford, S., Norman, S.J., Bowers, J., Adams, M., RowFarr, J., Koleva, B., Taylor, I., Rinman, M., Martin, K., Schndelbach, H., Greenhalgh, C.: Pushing mixed reality boundaries. Technical Report CID-84, Center for User Oriented IT Design (CID) (1999)
21. Tokunaga, E., van der Zee, A., Kurahashi, M., Nemoto, M., Nakajima, T.: A middleware infrastructure for building mixed reality applications in ubiquitous computing environments. (2004) 382–391
22. Rogers, Y., Lim, Y., Hazlewood, W.: Extending tabletops to support flexible collaborative interactions. In: Tabletop 2006, Adelaide, Australia, IEEE (2006) 71–79
23. Moeslund, T., Strring, M., Broll, W., Aish, F., Liu, Y., Granum, E.: (The arthur system: An augmented round table)
24. Broll, W., Meier, E., Schardt, T.: The virtual round table – a collaborative augmented multi–user environment. In: CVE2000, The Third International Conference on Collaborative Virtual Environments. (2000)
25. Regenbrecht, H.T., Wagner, M., Baratoff, G.: Magicmeeting: A collaborative tangible augmented reality system. Virtual Reality **6** (2002) 151–166
26. Regenbrecht, H., Lum, T., Kohler, P., Ott, C., Wagner, M., Wilke, W., Mueller, E.: Using augmented virtuality for remote collaboration. Presence: Teleoperators and Virtual Environments **13** (2004) 338–354(17)
27. Stanford, V., Garofolo, J., Galibert, O., Michel, M., Laprun, C.: The nist smart space and meeting room projects: signals, acquisition annotation, and metrics. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03). Volume 4. (2003) Page(s): IV– 736–9
28. Mills, K.: Smart spaces (1999)

Real-Time Multi-view 3D Object Tracking in Cluttered Scenes

Huan Jin[1,3], Gang Qian[2,3], and Stjepan Rajko[1,3]

[1] Dept. of Computer Science and Engineering
[2] Dept. of Electrical Engineering
[3] Arts, Media and Engineering Program,
Arizona State University, Tempe, AZ 85287,USA
{Huan.Jin, Gang.Qian, srajko}@asu.edu

Abstract. This paper presents an approach to real-time 3D object tracking in cluttered scenes using multiple synchronized and calibrated cameras. The goal is to accurately track targets over a long period of time in the presence of complete occlusion in some of the camera views. In the proposed system, color histogram was used to represent object appearance. Tracked 3D object locations were smoothed and new locations predicted using a Kalman filter. The predicted object 3D location was then projected onto all camera views to provide a search region for robust 2D object tracking and occlusion detection. The experimental results were validated using ground-truth data obtained from a marker-based motion capture system. The results illustrate that the proposed approach is capable of effective and robust 3D tracking of multiple objects in cluttered scenes.

1 Introduction

Robust 3D object visual tracking in cluttered scenes is a challenging task for computer vision community. Acquisition of accurate 3D object location is critical for many applications to facilitate further high-level tasks such as trajectory reasoning and motion analysis.

The major challenge for robust tracking is target occlusion. To overcome this challenge, many multi-view tracking approaches have been presented recently. People tracking has been the focus of these methods [1,2,3,4,5,6]. A common assumption made in these approaches is that people walk on a ground plane. This constraint is useful for people tracking in at least two ways. Firstly, people tracking can be simplified into a 2D tracking problem on the ground plane. Secondly, tracking results from multiple cameras can be connected through homographies. For example, most recently, in [6], an innovative approach to multiple people tracking was proposed. Ground plane occupancy is first estimated from foreground images to provide a basis for further people tracking based on color and refinement using global trajectory optimization. However, this assumption is no longer valid when the target being tracked can move arbitrarily in the 3D space. Therefore, robust 3D tracking of arbitrarily moving objects remains a challenge.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 647–656, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we present a robust real-time tracking system and proposed a simple way to handle occlusion. In the occurrence of occlusions, possible locations of the target in the occluded view are being maintained by projection using the 3D target location obtained from available 2D locations in the views that the target is visible. By doing so, the 2D tracking module is able to reject outliers (image regions similar to the target) and the target identity can be maintained when the object reappears in the view after full occlusion.

A closely related method to handle occlusion was presented in [7] for color-based object tracking using a particle filter. In this method, epipolar constraints are adopted to try to recover tracking in a camera view when the target has been occluded in this view. Although some promising results were reported in [7], epipolar constraints may not work well in the presence of false candidates along the epipolar line. In our proposed approach, the target search region is within a circle around the predicted position. There are much less chances that a false candidate will appear in this region than that of along an epipolar line. Therefore, the proposed method for handling occlusion is expected to perform better than the epipolar constraint-based method.

Some assumptions and restrictions exist in our approach. Firstly, in order to better cover the activity space, at least three cameras is required to integrate multi-view 2D tracking results and the combination of several cameras can cover the entire activity space. Secondly, we assume that the object's motion is smooth and there is no abrupt direction change. Thirdly, we consider all cameras are synchronized in advance. Finally, the illumination variation of the space needs to be maintained within an acceptable range due to the property of color tracking.

2 The Proposed Approach

2.1 System Overview

An overview of the proposed system is given by the diagram shown in Figure 1. We will briefly introduce each module.

Initialization. We assume that the cameras are synchronized and calibrated. Camera synchronization and calibration will be discussed in more detail in the next section on system implementation. The system is initialized manually by hand-picking the target object in all cameras views. By doing so, a color model of the object can be constructed accordingly. In addition, the initial 2D locations of the target in different views are also obtained.

3D localization. Once the object locations are given in two ore more camera views, triangulation [8] can be easily carried out to compute the 3D location of the object.

3D Kalman filtering. Based an estimate of the current object's 3D location, a 3D Kalman filter is used for trajectory smoothing and prediction of the object location at the next time instant.

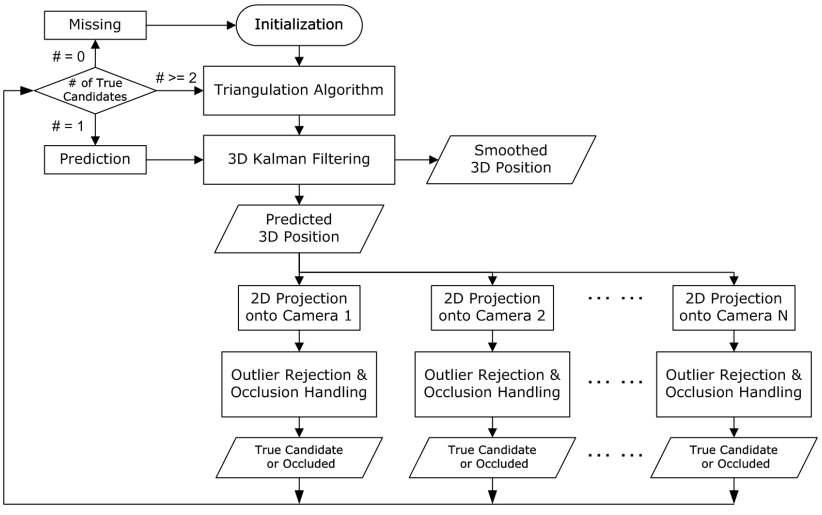


Fig. 1. Diagram of Tracking System

2D tracking. It is possible to have objects appearing similar to the target in the scene. To remove these outliers, the predicted object 3D location is projected onto all camera views to provide a search region so that the outliers outside of the search region can be rejected.

Occlusion Detection and Handling. If the target can not be detected in the search region in a particular camera view, it means that the object is occluded in this view. If there are at least two cameras can see the target at any given time instant, the target can still be tracked reliably. When the target reappears in one of views in which it was occluded, since it will fall into the search region, the target will be immediately detected. Its 2D location will then be used for 3D localization.

When the object is visible only in one view, we have also developed a simple but effective approach to maintain the 3D tracking over a short period of time. When the object stays visible only in one view over a time period longer then a pre-chosen threshold, a warning will be issued by the system indicating inaccurate 3D tracking results. When the object is completely invisible, a warning will be issued by the system indicating the loss of the object.

2.2 Color-Based Object Representation

Color histogram provides an efficient feature for object tracking as it is robust to partial occlusion, rotation and scale invariant and can be computed efficiently. A color histogram for the desired object is constructed in the tracking initialization stage. The Hue, Saturation, Value (HSV) color space is adopted in our system because it separates out hue (color) from saturation and brightness. Each object is represented by a 1D histogram using only the hue channel in HSV space. The

number of bins influences the sensitivity of tracker's response to color change. In our practice, we divide the hue channel into 16 bins. This color-based object representation also makes the tracker insensitive to small variation of ambient illumination.

2.3 Robust 2D Tracking and Outlier Rejection

In many applications the background scenes are cluttered and dynamic. For instance, in our application, there are many people walking in the space. Therefore, background subtraction does not help too much in terms of detection of the desired objects. For this reason, we directly use color histogram to classify objects of interest and employ connected component analysis and histogram template matching algorithm to locate them in 2D image coordinates.

The goal of 2D tracking is to detect possible object locations within the search region and to reject outliers which have similar color to the target being tracked. Outlier rejection can be performed either before or after color segmentation. If outlier detection is performed after color segmentation, firstly the entire image needs to be searched to obtain all possible object candidate locations. Then, all candidates outside of the search region will be rejected as outliers. On the other hand, one can perform only color segmentation within the search region. In this way, outlier rejection is done implicitly and it's more computationally efficient. In our current implementation, outlier rejection is done after color segmentation. The current system processing rate still satisfies our needs.

Color segmentation. In order to detect all object candidates in an image, we first identify all the pixels whose hues belong to the color histogram of the target with certain constraints on saturation and brightness. Then, *Connected Component Analysis* (CCA) and fast *Diamond Search* (DS) algorithm [9] are employed to get all pixel-connected blobs. We further verify each blob by comparing its histogram $\mathcal{H}_j$, j indicating the blob index, with the object model's histogram $\mathcal{H}$ using the *Bhattacharyya coefficient* metric [10], which represents a similarity metric with respect to the model histogram. The metric is given by

$$\rho(\mathcal{H}_j, \mathcal{H}) = \sum_{k=1}^{m} \sqrt{h_k^{(j)} h_k} \tag{1}$$

where $h_k^{(j)}$ is the normalized value for bin k in the j-th blob's histogram $\mathcal{H}_j$, $\mathcal{H}$ represents the object model's histogram. $\rho(\mathcal{H}_j, \mathcal{H})$ ranges from 0 to 1, with 1 indicating a perfect match. The Bhattacharyya distance $d(\mathcal{H}_j, \mathcal{H}) = \sqrt{1 - \rho(\mathcal{H}_j, \mathcal{H})}$ can be easily computed from the coefficient. A blob is a valid object candidate only if its Bhattacharyya distance is smaller than a certain threshold $\mathcal{T}$. A small threshold $\mathcal{T} = 0.2$ was taken in our implementation because of high occurrence of uniformly colored targets in our applications.

Outlier rejection. To remove outliers introduced by cluttered and varying scenes, we specify a search region, based on the object's 2D moving speed, to

detect the true object. The search area $\mathcal{S}^{(t)}$ at time t for a target is given by a circular region centered at the predicted 2D object position $l_{\mathcal{S}}^{(t)}(x, y)$ (projection of the prediction 3D object location) with the radius $r_{\mathcal{S}}^{(t)}$. The value of $r_{\mathcal{S}}^{(t)}$ is determined by (2).

$$r_{\mathcal{S}}^{(t)} = \begin{cases} T_1 & \text{if } v^{(t)} \leq T_1 \\ \alpha v^{(t)} + T_2 & \text{if } v^{(t)} > T_1 \end{cases} \tag{2}$$

where $v^{(t)}$ is the object's 2D speed at time t. α is a scaling factor and T_1, T_2 are thresholds. α, T_1 and T_2 are manually selected.

Any target candidate after color segmentation outside of $\mathcal{S}^{(t)}$ will be rejected as an outlier. It's still possible to have multiple candidates within $\mathcal{S}^{(t)}$. The one closest to $l_{\mathcal{S}}^{(t)}(x, y)$ will then be selected as the 2D target location in this camera view at the current time instant.

2.4 Handling of Occlusions

If the target can not be detected in the search region in a particular camera view, it means that the object is occluded in this view. Steps need to be taken to ensure that the target can be tracked as reliably as possible in the presence of such occlusions.

Target visible in at least two cameras. If there are at least two cameras that can see the target, the target can still be tracked reliably, using the search region in the occluded views. The search region is determined by the projection of target 3D location obtained from the views in which the target is visible. In this way, the target's 2D locations can be consistently estimated in those occluded views. When the target reappears in one of views in which it was occluded, since it will fall into the search region, the target will be immediately detected. This 2D location will then contribute to the estimation of the 3D position of the object.

Target visible in only one camera. Note that the 3D spatial point can be computed using triangulation only if there are two or more cameras detecting the object. When the object is detected in only one camera view, we are not able to recover the 3D location of object reliably. However, sometimes, there is only a very short period of time when the object is visible only by one camera and after that the target is visible by two or more cameras. A persistent tracking module is needed here to bridge this gap by using the most recent 3D location of the object and object location constraint provided by the view in which the object is visible. By doing so, the object can be continuously tracked over the period when it's only visible in only one view. When another view can see the object, reliable 3D tracking can be resumed.

In our system, a simple but effective approach is adopted to find an approximation of the object 3D location when it's visible only in one view. Based on the 2D location of the target in one view, a 3D ray passing through the corresponding camera center and target 2D location can be determined. The object can only

lie on this ray. Assume that the object is moving slowly. Given the most recent estimate of the object 3D location, say X_t, we find the closest 3D point on this ray to X_t and use this 3D point location to be X_{t+1}, i.e. the object location at the current time instant. Within a critical time range T, the estimation error can be alleviated by 3D Kalman filter. However, when the object stays visible only in one camera over a long period of time, this estimation error will accumulate and the 3D location estimate will drift away from the true value. Therefore, in our practice, we set $T = 120$ frames such that when the lasting time is beyond T, the system will report warning inidicating unreliable results.

3 System Implementation

Hardware and software platform selection. Three high-speed firewire Point Grey color cameras(Dragonfly2-COL) were deployed in our system for color-based target tracking. Such cameras can run up to 60 frames per second (fps) at VGA (640 × 480) resolution. One of the reasons that this type of cameras was selected was that they have features to establish autonomous image capture synchronization between cameras connected to the same firewire bus. These cameras can also run at external trigger mode such that they can be easily synchronized with other movement sensing modalities. The proposed tracking system was implemented using C++ and OpenCV on Pentium IV 3.0 GHz PC with 1.0 GB RAM. The system runs more than 30 fps with an image size of 640 × 480. The activity space is given by a $10'(length) \times 10'(width) \times 10'(height)$ volume. The cameras are widely separated around the space so that the cameras' field of view can cover the major part of the activity space (Figure 2).

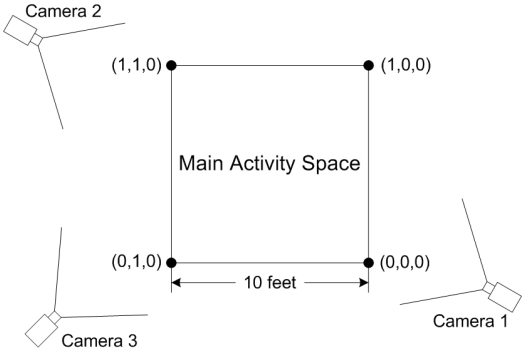


Fig. 2. Map of Real System Setup

Camera calibration. The cameras are calibrated using the multi-camera self-calibration toolkit developed by Svoboda *et al.*[11]. A glowing color ball is moved in the space and its 2D trajectories in all the camera views serve as feature correspondence used by the calibration toolkit.

The coordinate system used in the toolkit (say, C_0) is different from the desired 3D world coordinate system (say, C) that we want to use in our tracking system. The camera extrinsic parameters in C need to be determined. To do that, we first find three points with known cooridnates in C. They are the origin point, x-axis unit and y-axis unit of C respectively. Then the glowing ball is placed at these three points to record their coordinates in C_0. Given the coordinates of three points in C and C_0, it's straightforward to determine the coordinate transformation between these two systems. In our implementation, the unit length C in each dimension is 10 feet. Thus, each dimension of object's 3D position in the activity space will be within the range of $[0, 1]$.

4 Experimental Results

The tracking system has been tested using real video sequences. The tracking results were compared with ground-truth values provided by a 7-camera marker-based motion capture (MoCap) system produced by Motion Analysis Corporation (http://www.motionanalysis.com). The MoCap system is able to produce motion data with the millimeter-level accuracy. Synchronization signal from the MoCap system was used to trigger the video cameras at 30 fps. By doing so, these two heterogeneous systems are well synchronized and aligned.

Tracking results from two experiments are included in this paper. In the first experiment, two balls similar in both color (green) and size (20 cm diameter) were tracked. One of the balls stayed on the ground and the other carried and moved by a subject. The moving ball was sometimes occluded by the subject's body in some of the views. In Figure 3 (a), sample frames from three cameras are shown superimposed with the search region and estimated target location for both balls. The slim green circle represents the search region, the bold yellow circle 2D object position when it is visible, and the bold red circle the projected 2D position suggested by 3D prediction when object is detected to be fully occluded. The results show that our approach is able to handle full occlusion in both one camera view (*frame 151)* and two camera views (*frame 49 and 210).* Object identity can be successfully maintained when object reappear from the full occlusion state. Figure 3 (b) shows the tracking results in $X - Y - Z$ directions. It can be seen that the trajectory is smooth in all three dimensions. Tracking errors are given in Figure 3 (b). Both dimensional (in each direction) and combined (distance from the tracked ball to the ground-truth) errors were plotted. The average tracking error of the moving ball is about 13.2 cm.

In the second experiment, a subject's head was tracked. Two markers were put on both cheeks of the subject. The mean of these two markers provided ground-truth of the 3D head position. The tracking results are given in Figure 4. The average error compared to MoCap system is about 14.3 cm.

We have noticed that in both experiments the tracking errors are relatively large compared to the size of the targets. The major cause of the error could be a less accurate spatial coordinate alignment of the MoCap system and the

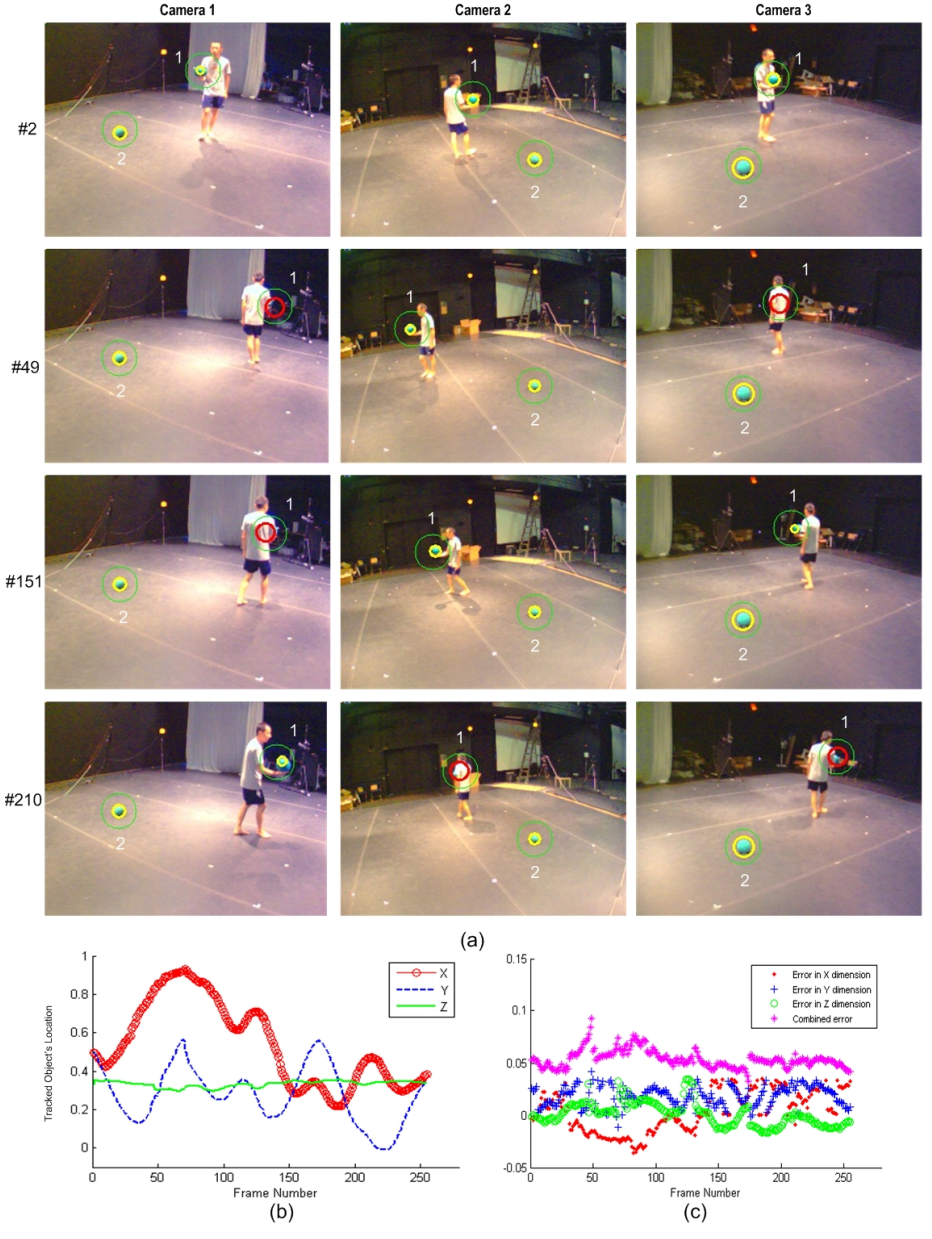


Fig. 3. (a) Tracking of two green balls in the presence of full occlusion; (b) Trajectories of the moving ball in all directions; (c) X ('.'), Y ('+') and Z ('o') directional and combined ('*') tracking errors of the moving ball

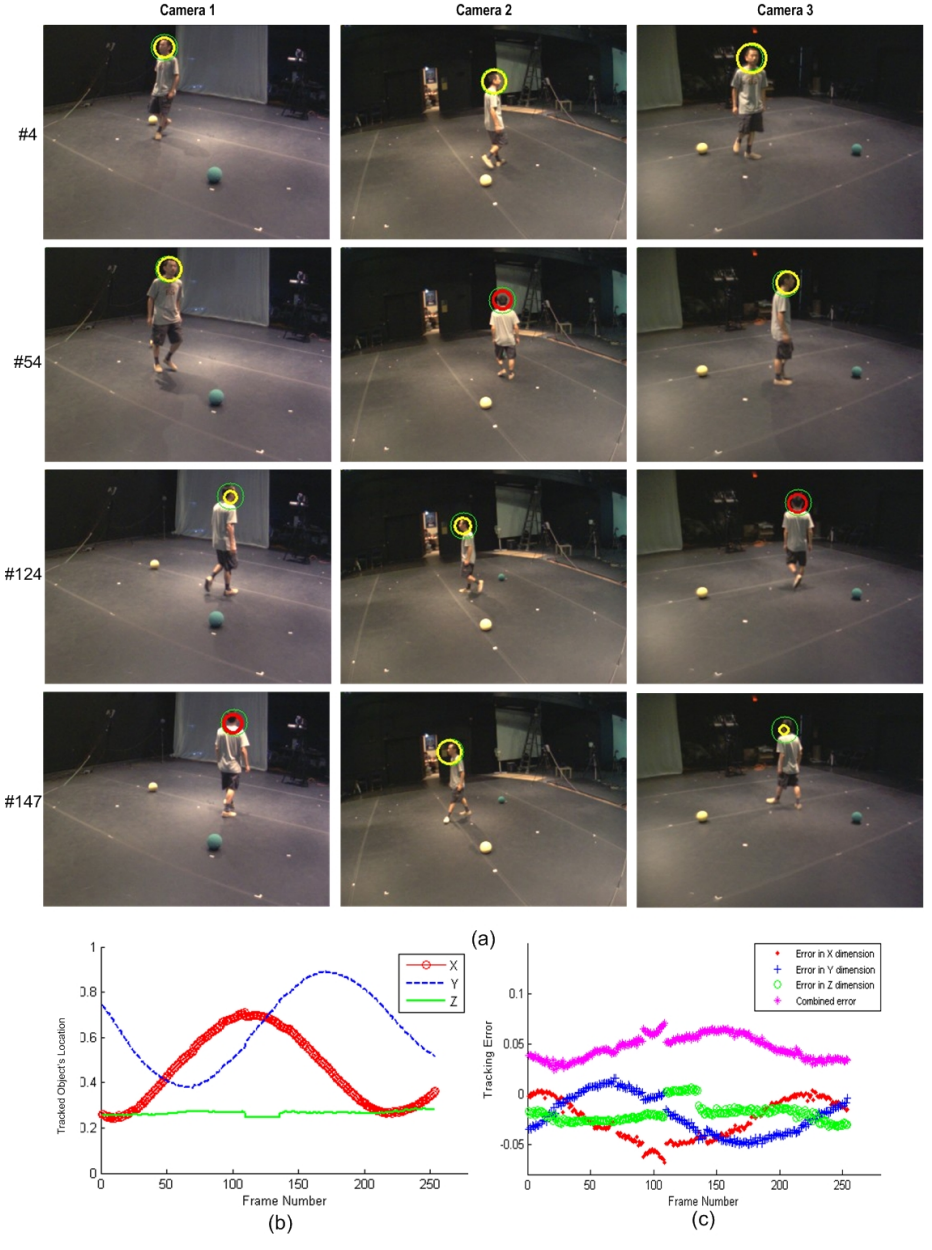


Fig. 4. ((a) Head tracking in the presence of full occlusion; (b) Head trajectories in all directions; (c) X ('.'), Y ('+') and Z ('o') directional and combined ('*') tracking errors of the moving ball

video-tracking system. We will closely examine this issue and improve the coordinate alignment of the two systems.

5 Conclusions

We present an approach to performing real-time multi-view 3D object tracking. The experiment results show that our approach can robustly provide consistent tracking in the presence of occlusions. The future work will mainly focus on removal of outliers within the search region when object is fully occluded.

References

1. Kang, J., Cohen, I., Medioni, G.: Tracking people in crowded scenes across multiple cameras. In: Asian Conference on Computer Vision. (2004)
2. Mittal, A., Davis, L.S.: M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In: International Journal of Computer Vision (IJCV). Number 3 (2003) 189–203
3. Javed, O., Rasheed, Z., Alatas, O., Shah, M.: Knight: A real time surveillance system for multiple overlapping and non-overlapping cameras. In: 4th International Conference on Multimedia and Expo. (2003)
4. Yue, Z., Zhou, S., Chellappa, R.: Robust two-camera tracking using homography. In: Proc. of IEEE Intl Conf. on Acoustics, Speech and Signal Processing. (2004) 1–4
5. Black, J., Ellis, T., Rosin, P.: Multi view image surveillance and tracking. In: IEEE Workshop on Motion and Video Computing. (2002)
6. Berclaz, J., Fleuret, F., Fua, P.: Robust people tracking with global trajectory optimization. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2006)
7. Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., Van Gool, L.: Color-based object tracking in multi-camera environments. In Michaelis, B., Krell, G., eds.: 25th Pattern Recognition Symposium, DAGM'03. Number 2781 in LNCS, Berlin, Germany, Springer (2003) 591–599
8. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
9. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans. on Image Processing **9** (2000) 287–290
10. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 2., Hilton Head Island, South Carolina, USA (2000) 142–149
11. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments **14** (2005) 407–422

Visualizing Natural Stereo Images in Short Distance: A New Approach

Antonia Lucinelma Pessoa Albuquerque*, Sergio Pinheiro, and Rubens Melo

PUC-Rio, Pontifical Catholic University of Rio de Janeiro – Department of Informatics
Rua Marquês de São Vicente, 225, Rio de Janeiro, RJ - Brasil - 22453-900
nelma@inf.puc-rio.br, sergio@visgraf.impa.br,
rubens@inf.puc-rio.br

Abstract. This work approaches the visualization of natural stereo images through webcams, in real-time. After analyzing the conventional stereo model geometry and its constraints related to viewer's movement, it became necessary to eliminate those geometric constrains, when applying the model to visualizing stereoscopic webcams images in the context of remote interactions. Then, based on natural human binocular vision geometry, we propose a stereo processing model for short distance visualization.

1 Introduction

Webcam is a low cost technology that tries to reproduce face-to-face interactions, and it is becoming very popular each day more for applications such as video chat and videoconference. This work proposes real-time stereoscopic visualization through webcams as one solution to enrich visualization in those remote interactions, by adding the depth perception to camera images, under the constraints of not increasing the computing power to produce stereo images and without adding any special hardware to a basic PC system.

In the context of remote interactions, the acquisition and visualization of webcam stereo images require special analysis when considering: the short distance between the viewer and the computer screen, the free movement expected by the viewer to have before the computer, and the real-time images updating required for those kinds of applications. Under those constraints, the conventional stereoscopic model doesn't fulfill such requirements, then becoming inadequate to this approach.

It is known from human vision literature the high influence of distance in depth perception and the stereo acuity dependence on it. The new approach will be based on those human vision concepts, since the human eye has different responses to short and long distances viewing.

This paper proposes a stereo processing model for short distance real-time visualization of camera images. Section 2 discusses the conventional stereo model geometry and its constraints; Section 3 presents the approach basis from binocular human visual system; Section 4 introduces the model framework, its geometry and compares the both models. Next, Section 5 shows the resulting images.

* Corresponding author: nelma_27@hotmail.com (online contact), Tel.: +55 21 9238-7325.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 657 – 667, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Stereoscopic Display Techniques

There are many stereoscopic display techniques developed and grouped as the ones with viewing aids and the ones called auto-stereoscopic. The last ones are still expensive, requiring more computing power and the use of special display screen [8]. Among the first ones, anaglyph is the most traditional, easy to compute and allowing displaying stereo images in hardcopy or in computers, without any special device than a paper filter glasses.

Some developments looking to improve virtual interactions through videos are complex projects requiring both, sophisticated technology and computational processing. Examples are the immersive 3D videoconference system, Im.point [15] and The office of the future project [16].

Since, the addition of special hardware is out of consideration for the approach that uses an ordinary PC and standard display screen, anaglyph will be the visualization technique chosen and the stereo images processing will be only software based. There is no binocular webcam prototype available to this project, so the webcams manipulation is a constraint to the solution.

Next section analyzes the conventional stereo processing geometry related to our context application.

2.1 The Conventional Stereo Model

The conventional stereo model for parallel cameras basically considers a perspective projection, assuming that cameras - "like eyes" - are gazing at infinity, having an interaxial distance between them as shown in Fig. 1.

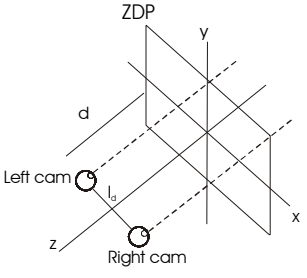


Fig. 1. Parallel cameras stereo model

The zero disparity plane (ZDP) is obtained by translating the two perspective views along the x-axis, which is called horizontal image translation (HIT) [1].

The toed-in cameras model includes a convergence angle between the cameras, but according to stereoscopic literature [1] [3] [5] [7], its use is discouraged because it includes distortions on images when projected on the plane, producing the well-known keystone effect and the depth plane curvature, this last one analyzed in [7].

Graham et al [5] affirm: "good images are hard to get right and the key variable that must be determined for any stereoscopic image creation is the camera separation, as this directly affects the amount of depth perceived in final image". The solution

presented varies cameras separation, for each time the viewer moves, in order to adjust the scene about depth distribution and distortion.

Most stereo algorithms or experiments [5] [8] are based on varying the distance between the cameras and the parallax, to determine the ideal visualization parameters. Those solutions are very applicable to virtual cameras, that are totally programming controlled, or in photography approach.

Some classical geometric constraints in stereoscopic images computation related to viewer changing position, described in [1], are:

if a translation occurs on x or y axis, the two centers of projections need to remain centered about a point perpendicular to the center of the screen;

if a translation occurs on z axis, the z-coordinate of the ZDP will need to be changed;

if any scaling occurs, or any operations that changes ranges in x, y or z, then the z of ZDP, the distance from the cameras, d, and the camera separation need to be changed.

Those constraints establish lots of limitations to visualize real-time natural stereo images, when the viewer expects to move freely before the computer, as in webcam interactions. It is not practical to the viewer to keep adjusting cameras separation each time he/she moves, as it happens with virtual cameras, neither restricts the viewer positioning to always look perpendicular to the screen. Besides, the viewer must not have knowledge of what is a ZDP neither parallax images.

This makes the conventional model geometry inadequate to this approach, being necessary to eliminate those constraints.

Looking for a solution, this work will analyze the human binocular vision concepts related to images fusion, since the human eye has different responses to short and long distances viewing.

3 Binocular Vision Concepts

The main point of interest is to know how the binocular vision is related to images fusion. Disparity is defined as the difference between the angles subtended at the two eyes by two points separated in space.

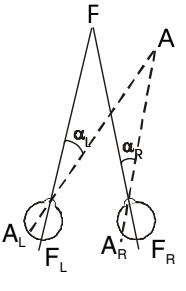


Fig. 2. Binocular disparity

Being a point A in space and F the fixation point, disparity is given by $\alpha_L - \alpha_R$; Fig. 2 shows the disparity concept.

3.1 Panum's Fusional Area and Horopter

Panum's fusional area corresponds to the region of stereo vision. The image on the retina of one eye can be fused with the image on the retina of the other eye only if the images fall in Panum's area, even if the disparity in retinal images exists; otherwise, it will happen diplopia (double vision) [9].

Inside the fusional area, the two monocular images cause a single three-dimensional percept. Objects outside the fusional area produce large disparities and generate monocular half-images that normally cannot be fused [4]. The locus of points within the fusional area that have images in the same position on the fovea in each eye is called Horopter. The points that should have zero disparity lie on the horopter.

3.2 Vieth-Müller circle

The Vieth-Müller circle (VMC) is the theoretical geometric horizontal horopter [11]. It is based on the geometric principle that equals angles within a circle define equal arcs on that circle when their vertices lie on its circumference.

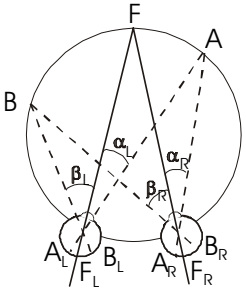


Fig. 3. Vieth-Müller Circle

Thus, the arc FA (Fig. 3), as the arc FB, subtends equal angles at the two eyes, $\alpha_L = \alpha_R$ e $\beta_L = \beta_R$ [11]. Therefore, because the eyes are converged at this angle to foveate at the fixation point, all other points on this circle must be in geometric correspondence [10].

Properties of the Vieth-Müller circle [12]
If the eye movement is a pure version eye movement, the fixation point stays on the same VMC.

Theorem 1 – If point Q is on Vieth-Müller circle, its disparity is zero.

Theorem 2 – Disparity is non-zero outside/inside Vieth-Müller circle (with opposite sign for outside/inside).

The theoretical *vertical* horopter is defined as a straight line through the fixation point. Empirically there are other types of horopter, which are variations of the

theoretical horopter, well described in [10], and some experiments about the empirical horopter considering the Hering-Hillebrand deviation can be found in [11].

The Vieth-Müller circle is related to a symmetric gaze on the horizontal viewing plane. The most general case is to allow fixation at any point, shown in Fig. 4. Helmholtz (1866) developed a third degree curve that forms a single loop spiral within an abstract cylinder projected up and down from the Vieth-Müller circle [14], called the horopter 3D, which corresponds to the region of all asymmetric fixations, in space (Fig. 4). Tyler affirms: "The value of projecting the horopter line onto an abstract cylinder in space is that it allows dissociation of the effects of ocular rotations from the inherent disparity constraints" [10].

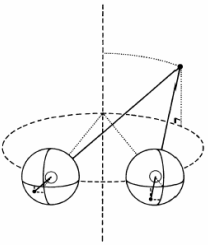


Fig. 4. Asymmetric fixation point, from [10]

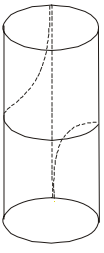
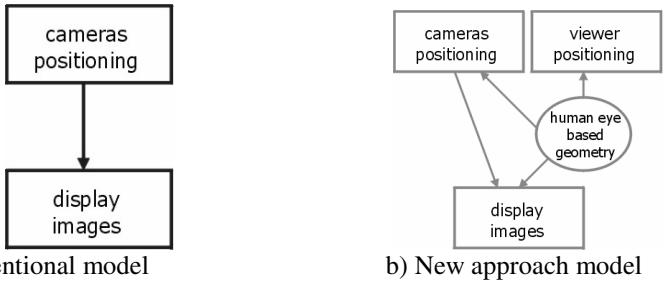


Fig. 5. Horopter 3D

Our solution, described in next section, is an approximation of the complex eye geometry process, based on the theoretical horopter.

4 The Framework

For the webcams system, there is no post processing. The capture and display work dynamically, in real-time: the user is an object in scene, since the cameras are like the "eyes" capturing his/her image; by the other hand, the user is the viewer when he/she sees the generated stereo image. This interpretation let us to enunciate the *User as viewer/object Paradigm*. The real and virtual worlds are interlaced in this paradigm.



a)) Conventional model b) New approach model

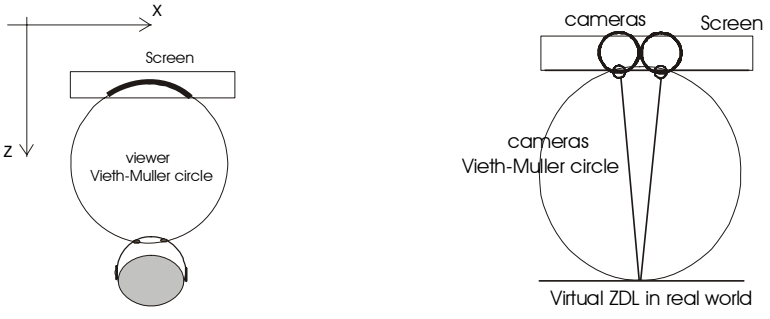
Fig. 6. Stereo Model Framework

The framework, in Fig. 6, separates *cameras space* and *viewer space*. The human eye geometry influences the cameras positioning (calibration and acquisition) and the images display (visualization).

The framework were implemented in two steps: when the geometry only influences the camera positioning, and the visualization process is the same of the conventional processing, which we will refer to as Module1; and the full processing when the eye geometry influences both cameras arrangement and images visualization, which we will refer to as Module2.

4.1 Applying the Horopter Geometry to Our Approach

The *viewer space*, based on the Vieth-Muller circle is shown in Fig. 7.a; the diameter of the circle is the viewer distance to the screen, *d*, and then the theoretical horopter must lie on the screen surface, being the ZDL (zero disparity locus).



a) viewer space geometry; b) cameras space

Fig. 7. Spaces geometry

The *cameras space* follows the same geometry of the viewer space. The interaxial camera distance is the average of the viewer interocular distance, about 6,5cm, which stays fixed all the time, converging to a ZDL (zero disparity locus) in real world, and the convergence distance, i.e. where the optical axis of the two cameras intersect, is the diameter of the hypothetical circle.

Information about intrinsic cameras parameters are not considered for calibration, since there is a large number of webcam brands on the market and its technical features, like intrinsic parameters, are not usually available. Then it makes the solution independent on the camera model, to be used with any off the shelf webcam.

The cameras images are interlaced during the acquisition, at ZDL, having on them a region of zero disparity, and the system keeps that relation unchangeable during the process. There is no post translation or rotation explicitly applied on images, as in conventional model. The two processes, Module1 and Module2 differ for the images visualization process, as explained below.

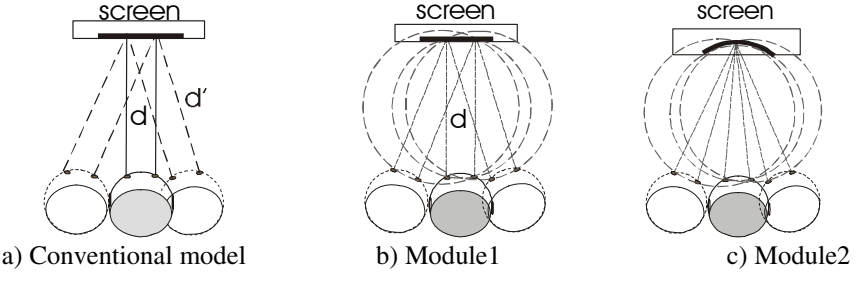
Module1
Module1 projects the acquired images pair onto a plane, as in conventional model. In order to minimize the distortions introduced to the stereo images pair, due to the rotation of cameras axis, the angle between cameras is very small, and the images keep interlaced as in acquisition calibration. This construction of cameras space made possible to fuse images easily and keep the fusion during images visualization, without parallax adjustment. This module is an intermediary step to analyze the geometry applied to cameras space, independently.

Module2
The viewer geometry influences the visualization process. The images are rendered onto a curved surface, corresponding to the theoretical horopter 3D simplification, because the eyes gaze can have asymmetrical fixation in any direction over the images. Based on the viewer Vieth-Müller circle, as shown in Fig. 7.a, the dark curved part showed inside the screen represents the cylindrical surface onto which the images are projected. We are also including the vertical horopter geometry in computation.

This cylindrical surface varies its curvature according to the viewer distance to the screen and his/her viewing angle; this simulates the convergence/accommodation behavior and would correspond to the variation in theoretical horopter curvature, as measured by Hering-Hillebrand deviation [10] [11]. The model incorporates this convergence/accommodation relationship into the stereoscopic processing that is not present in conventional model. Computationally, the geometric distortions introduced by cameras rotations are compensated by the cylindrical projection, as shown in Fig. 12.

4.2 Schematic Analysis of the Geometric Constraints

When the viewer moves on x-axis:
By moving before the computer, according to the conventional geometry model, a shear distortion is introduced by the viewer angle, as shown in Fig. 8.a, even if the images didn't suffer any changing; the distance *d* becomes *d'*.



a) Conventional model b) Module1 c) Module2

Fig. 8. User moving on x-axis direction

The dark target on the screen in Fig. 8 represents the displayed image. Fig. 8.b shows the VMC geometry behavior related to the plane display (Module1). The lines representing the visual direction intersecting the image, shows that, although most

part of the image lies on the theoretical horopter, even when user moves, there are some image areas that stays out of the horopter region, because of the plane geometry. In this case, shear distortion can't be eliminated (Módule1). Fig. 8.c shows that image points lie on the theoretical horopter even when the user moves on x-axis, reducing considerably the problems mentioned as constraints in section 2.

When the viewer moves on z-axis:

In Module1, by moving on z-axis, it is like the diameter of the VMC circle will change, and then this would cause a variation in depth perception on images, but not a loss of images fusion, because the cameras geometry keep the parallax images in the same range of values from acquisition.

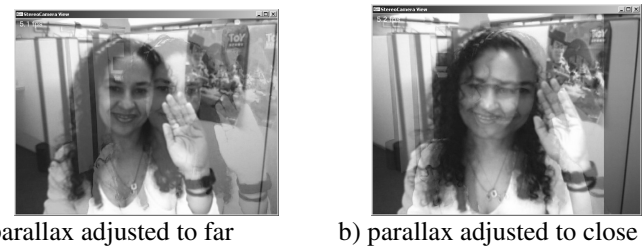
In Module2, by moving on z-axis, the behavior is similar to Module1. But because of the projection surface geometry, and its curvature variation dependent on the viewer distance, it is possible to adjust parallax images indirectly as a function of viewer distance. This allows a better depth perception than in Module1.

5 Resulting Images

All the images were synchronized and captured during a real-time processing, in 640x480 pixels, using a PC Pentium 4, Windows system, 512Mb RAM and 1.6GHZ, a video board NVidia GForce2MX440, a pair of off the shelf webcams and filter paper glasses. The scene is complex including textures, shadows, and illumination of a natural environment.

Conventional model

The images generated by the conventional stereo model are shown in Fig. 9. Cameras were positioned in parallel configuration and it needed to adjust parallax manually in order to obtain fusion. It can be noticed that in Fig. 9.b, the cyan image appear at the right border without the red image, because of parallax adjustment.



a) parallax adjusted to far b) parallax adjusted to close

Fig. 9. Conventional stereo model images

Those images are difficult to get fusion instantaneously. It takes few seconds to get accommodation. When observing those images, by moving in z-axis, it happens loss of fusion and it is confusing to adjust eyes for fusion each time we move before the computer. When observing images, the shear distortion introduced while moving parallel to screen, x-axis, is very evident.

Module1 (camera geometry)
This module improves stereo image visualization because it becomes quite easy to get fusion. Observing the images (Fig. 10), the red and cyan images keep coupled in borders because of images interlacing during acquisition. If the viewer moves in z-axis, perpendicular to the screen, there is no loss of fusion neither needing of parallax readjustment. This geometry still keeps shear distortion when moving in x-axis.



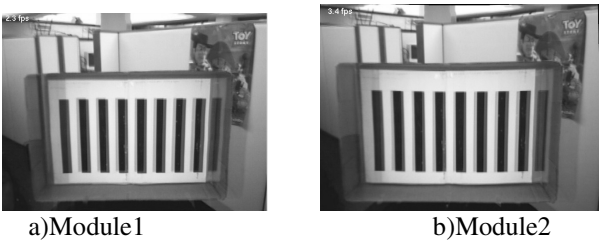
Fig. 10. Module1 images – color and gray anaglyph

Module2 (viewer-camera-geometry)
Fig. 11 shows that is very easy to obtain fusion, it is instantaneous; the distortion when moving parallel to the screen, x-axis, is quite reduced; there is no loss of fusion when moving in z-axis. Images have good quality. Depth perception becomes even better than in Module1. This module adjusts projection according to the viewer distance, but the program still processes this parameter manually. The images keep the same stereo vision, in gray, color and also with dimmed light.



Fig. 11. Module2 images, also in dimmed lighting - color and gray anaglyph

The images in Fig. 12 shows the keystone distortion in Module1 and its compensation in Module2. Those images are good to be compared without glasses.



a)Module1 b)Module2

Fig. 12. Comparing keystone distortion

6 Conclusions

This paper presented the stereoscopic visualization through webcams, in real-time. We discussed the geometry for conventional computer stereo model and its constraints, to conclude that it is not applicable to our approach. It is not possible to avoid the viewer movements when using webcams, neither keep controlling the parallax and interaxial distance continuously simultaneous to the images visualization.

After presenting some human visual concepts, we introduced a new framework for stereo processing in short distance visualization, based on human eye geometry.

The models, conventional and the new approach were compared. The solution presented reduced many problems that are limitations of the conventional stereo model, and eliminated the continuous adjustment of interaxial cameras distance and parallax images during the real-time processing. The stereo model proposed here is a simplification of the theoretical human binocular vision geometry, based on fusional process presented in section 3. Therefore, our model is much closer to the real eye geometry than the conventional model.

It has a project for an experimental design of evaluation to validating the method related to user perception.

The colored images can be seen at http://w3.impa.br/~nelma/ISVC2006.

References

1. StereoGraphics Corporation. StereoGraphics Developers' Handbook. (1997). http://www.stereographics.com/support/downloads_support/handbook.pdf.2.
2. Ugur M.Legoglu). Artificial versus natural stereo depth perception. Hints from Life to AI, edited by Ugur HALICI, METU, 1994.
3. W. A. IJsselsteijn, H. de Ridder, and J. Vliegen. Subjective Evaluation of Stereoscopic Images: Effects of Camera Parameters and Display Duration. IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, No. 2, March 2000.
4. Cheng Hong Yang. Geometric models in stereoscopic video. Université du Quebec. Rapport technique de l'INRS – Télécommunicatons no. 95-12,. 1995.
5. Graham Jones, Delman Lee, Nicolas Holliman, David Ezra. Controlling perceived depth in stereoscopic images. Sharp Laboratories of Europe Ltd, UK, 2001.
6. Gian F. Poggio and Tomaso Poggio. The analysis of stereopsis. Ann. Rev. Neurosi. 7:379-412, by Annual Reviews Inc, 1984.
7. A. Woods, T. Docherty, R. Koch. Image Distortions in Stereoscopic Video Systems. Proceedings of SPIE, Vol. 1915, Stereoscopic Displays and Applications IV, 1993.
8. R.L.J. van Eijk. Beyond the flat screen. Minimal and optimal camera-base distances for viewing 3-D images. Eindhoven University of Technology, 2003.
9. Helga Kolb, E. F.; Nelson, R.. Webvision: The perception of space. John Moran Eye Center, University of Utah, 2005. http://webvision.med.utah.edu/space_perception.html
10. C. W. Tyler. Binocular Vision. In, Biomedical Foundations of Ophthalmology. Duane T.D., Jaeger E.A. (Eds.), Lippincott: Philadelphia, Ch. 24, 1982 (with A.B. Scott).
11. Normal binocular vision Course. Center for development and functional imaging. http://tesla.cdfi.uab.edu:8200/cb/coursemat/PO222/part5.ppt, 2005.

12. J. Malik and Y. Chen. Computer Vision, Lecture 10. University of California at Berkeley, 1999.
13. D. F. McAllister (Editor) (1993). Stereo Computer Graphics and Other True 3D Technologies, Princeton University Press.
14. Tyler, C. The horopter and binocular fusion. In Binocular vision (pp. 19–37). Boca Raton, Ann Arbor, Boston: CRC Press Inc, 1991.
15. Schreer, O.; Kauff, P. An immersive 3d videoconference system based on a shared virtual environment. Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut, 2002. http://www.hhi.fraunhofer.de/english/bs/projects/impoint/impoint.html
16. Henry Fuchs. The office of the future and Group Tele-immersion, 2002. http://www.cs.unc.edu/Research/ootf/

Next Best View Algorithms for Interior and Exterior Model Acquisition

Bradley D. Null and Eric D. Sinzinger

Department of Computer Science, Texas Tech University
Box 43104 Lubbock , TX 79409-3104
{bradley.d.null, eric.sinzinger}@ttu.edu

Abstract. Rapid determination of optimal laser range finder placement is becoming of increased importance in both computer vision and robotics. The need of next-best-view algorithms for intelligent decisions in where to place the laser range finder is important for scanning both objects and landscapes to assure that the scene is fully realized and can be registered accurately. Presented here are methods for determining sensor placement to maximize the amount of information that is learned about a scene or object. Using occupancy grids of voxels and ray tracing, the placement of the sensor can be optimized for maximum collection of new data. This work provides fast algorithms that determine optimal sensor placement both outside an object of interest and inside of a closed environment. These methods take into consideration the limitations of the sensor, the interaction of the sensor to the environment, and its placement in only known areas to restrict the search space.

1 Introduction

To assure an accurate representation of a model in three dimensions, many separate scans of the interest area must be taken and registered. Choosing where to place the sensor such that all of the area is to be scanned can be done robustly by gradually moving the sensor around the area of interest assuring that all data describing the entire area has been collected. This is commonly referred to as the next best view (NBV) problem. The general problem can be divided into two domains. The first is the exterior NBV problem where the scanner is located on the exterior of the object (e.g., a statue). The second is the interior NBV problem where the scanner is located inside the interior of the object (e.g., a room). An extensive overview of NBV can be found in [1].

1.1 Exterior NBV Problem

The exterior NBV problem refers to the placement of the sensor around a specific object, for example a statue, such that the entire model is acquired with the most efficient sensor placements. The previous work can be divided into two general categories — voxel occupancy and mesh representation.

The use of occupancy grids or voxel grids for solving this problem takes the rays casted from the sensor outward and carves out what is known and what

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 668–677, 2006.
© Springer-Verlag Berlin Heidelberg 2006

is not known about the model. The first NBV algorithm in [2] simply places the sensor uniformly around the object and using voxel intersections from the rays casted chooses the NBV. It calculates the surface normals of the object to choose eight optimal positions around the object by scanning the entire space. The algorithm presented in [3] constructs an occluded model of the unknown space. The occluded model is used to orient the sensor such that the normal vector of the sensor points towards the center of the largest occluded region. This works well for sequential scans that read in contiguous parts of an object but suffers from problems in determining the greatest unknown region if there are known portions within a greater unknown region.

By creating a mesh of the data obtained from the range finder, it is easier to determine the shape of the object and therefore cast rays through the connected polygons of the mesh. However, the mesh construction algorithm can be an expensive operation. The method of [4] presents an approach where the mesh of the model is constructed and the regions of void volumes, or unknown space, are found. The sensor position and pose that would uncover the most void volumes, determined by ray tracing through the polygons, is chosen. This method suffers from an inability to easily define self-occluding objects. The two-stage algorithm presented in [5] again assumes the simple calculation of a mesh from the given range data. By using the mesh it determines the occluded areas which are similar to void volumes from the previous algorithm. To find the NBV the sensor is placed at the edge of the largest occluded area and points the normal vector of the sensor towards the center of the occlusion.

The exterior NBV method presented in this paper uses voxel occupancy grids similar to [2] but intelligently constrains the search space for faster computations. The main goal of this work is to scan large objects such as 10 foot tall statues at a distance. This is contrasted to the previous work which generally is concerned with small, handheld models, where there were few limitations on sensor position. For large objects, the application of the algorithm presented must consider objects with the limitations of the sensor available. This led to improvements in our work, by limiting many aspects of the search for a NBV, such as height, scanning region, and range. These optimizations dramatically reduce the unnecessary calculations and create a realtime solution.

1.2 Interior NBV Problem

The interior NBV problem places the sensor inside of the scanned region. This problem is closely related to terrain acquisition research such as simultaneous localization and mapping (SLAM) [6]. The applications of most SLAM methods are in navigation and exploration of unknown regions surrounding an autonomous robot.

The methods described in [7] show a second sensor determining a 2D floor plan of the area to be mapped is created. Once this is obtained the 3D scans are taken and mapped to each edge on the floor plan. To find the NBV, this method navigates the robot to the unseen edges in the 2D grid for further scanning. A similar method to the one presented here is given in [8] where the NBV

position is only considered in safe, or known to be empty, regions. However, the method described in [8] is not optimal because it does not robustly choose potential positions, opting rather to choose random ones. Many of these SLAM algorithms such as [9] simply rely on a priori knowledge of the region and select a next acquisition point at the farthest known edge and the pose towards the region of most occlusion from the current representation. This, though simplistic, works efficiently to allow the robot to move quickly and react to changes in the scene. Even simpler more realistic views scan in all that is possible at each region then divide and conquer each unseen region. This method is shown in [10] and takes considerations for the degrees of freedom of the robot in its NBV algorithm whereas most other NBV algorithms do not.

The methods presented for modeling environments in this paper assume that the sensor's limitations (i.e. how low and high it can be placed, the volume the sensor takes up) or degrees of freedom, as in [10], can be entered into the algorithms as parameters and the 3D map that is generated by the sensor can be used to orient the system and determine its ability to acquire a generated NBV. Unlike the methods that require the generation of a 2D map of the region, this method uses the current 3D map for localization and navigation in the future. The simulation of the sensor in this NBV algorithm assumes that the laser range finder can be rotated 2π around its central axis, the angular span of any single scan is a set angle (e.g., $\pi/9$), and the angular range of the scan is $\pi/2$. In this work, the angular range of the scan is vertical and the angular span of the scan is horizontal.

2 Voxels of Registered Data

A volume of voxels is used to represent the registered data from the scanners. The occupancy grid has three standard states that represent voxels — empty, occupied and unknown. The voxels that are unknown represent what has not been, or what cannot be (inside of a closed volume) scanned by the sensor. Occupied voxels can be labeled by mapping the registered data to the volume voxels. The known empty space is identified by carving the space between the laser scanner optical center and each registered data point. A ray is shot from the laser scanner optical center. As it goes to the known occupied point, all unknown voxels along the path are converted to known empty voxels. The remaining voxels are classified as unknown.

3 The Exterior NBV Algorithm

In this instance of the NBV algorithm it is assumed that one scan of the interesting object has been obtained as a starting point for the process. It is also assumed that in cases where the range of the sensor is unknown, the scan that has the farthest distance from the center of the object determines the range, R, of the sensor from the object center. It is also assumed that the scanner is of a fixed height, H.

When the data for the object is read in to the system, it is normalized with the occupancy grid resolutions for width, height and length to populate the grid. The center of the object with the normalized coordinates is just the midpoint of the occupancy grid and is represented by (C_x, C_y, C_z).

A starting point for the NBV analysis is determined by projecting a ray outwards from (C_x, H, C_Z) with a maximal distance of R. The potential positions for the sensor placement are now determined by the coordinates

$$(X, Y, Z) = (C_x + r\cos(\psi), H, C_z + r\sin(\psi)) \tag{1}$$

where $\psi \in [0, 2\pi)$ represents the angle around the circular plane with respect to the origin, and $r \in [0, R]$ represents the distance. The step sizes of r and ψ, determine the search space size. For the results below, the step size of r is 1 and ψ is $\pi/9$.

To determine the number of unknown voxels intersected using the carving method described in the previous section, rays are cast from the potential sensor position. Every time an unknown voxel is intersected by a ray originating at a potential sensor position, the counter, K, is incremented by the current value of $t = (\ell/R)$ where ℓ is the distance to the current unknown voxel. The reason for the weighted count is that the proximity of the potential point to any given unknown voxel suffers from the fact that when rays are shot out from that point they intersect voxels that are close by more frequently than ones that are farther away. The position with the largest K is the NBV.

This algorithm is quite slow if the scan simulation of each potential search position is performed completely. For efficient usage, a search space reduction is performed. Since the potential sensor positions in this method are outside of the model, the only points of interest that should be scanned at each potential sensor position lie within the occupancy grid volume that was carved out when the model was read. For a given potential sensor position, (X, Y, Z), the search space can be reduced by checking the eight corners, $B = \{(x_i, y_i, z_i)\}_{i=0}^{7}$, of the occupancy grid volume. To limit the search in this fashion the left most, θ_0, and right most, θ_1, angles, θ, in the carving function should be limited to

$$\theta_0 = \max_{(x,y,z)\in B} \left(\arctan\left(\frac{z - Z}{x - X} \right) \right) \tag{2}$$

and

$$\theta_1 = \min_{(x,y,z)\in B} \left(\arctan\left(\frac{z - Z}{x - X} \right) \right). \tag{3}$$

Since the height of the sensor is held constant in this algorithm, the range of $\phi \in [\phi_0, \phi_1]$ can be found by finding the corner on the lower half of the voxel grid, $(x_0, y_0, z_0) \in B$ that has the smallest distance to the new potential position. The calculation of the limiting bottom-most angle is

$$\phi_0 = \arctan\left(\frac{y_0 - Y}{\sqrt{x_0^2 + z_0^2} - \sqrt{X^2 + Z^2}} \right). \tag{4}$$

Since the closest corner on the lower half of the voxel grid is already known, the closest corner on the upper half of the voxel grid is known (x_1, y_1, z_1) and the limiting topmost angle is calculated as

$$\phi_1 = \arctan\left(\frac{y_1 - Y}{\sqrt{x_1^2 + z_1^2} - \sqrt{X^2 + Z^2}}\right). \tag{5}$$

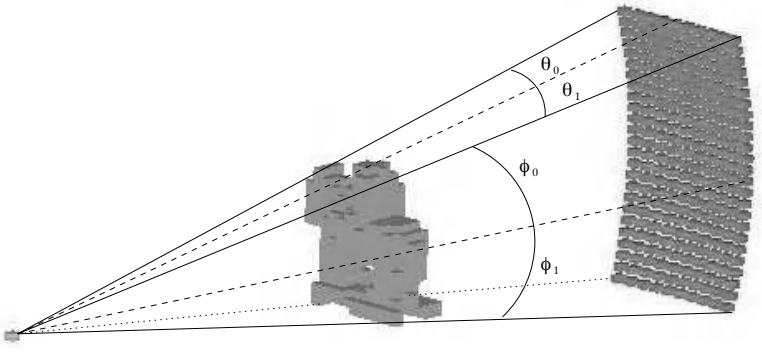


Fig. 1. Diagram of the exterior scanning process and the limiting angles of acquisition

Figure 1 shows the voxelized representation of the statue that was used in the experiments. The voxel where the lines intersect is the camera and the rays leading to the "wall" of voxels is the scanning frustum. This visually illustrates the angles that limit the acquisition area and speed up the algorithm.

4 The Interior NBV Algorithm

This case of the NBV algorithm assumes the interest area fully encompasses the sensor and requires a 2π view of the environment. This case also entails that the potential sensor positions lie within the model and must be limited to areas where voxels are known to be empty and are specific distances from the floor and the ceiling of the model. For more accurate results, the resolution of the occupancy grid should be set to some factor of the dimensions of the environment.

This interior NBV method assumes that there is at least one scan of the model taken so that a voxel occupancy grid and first potential sensor placement can be generated. The original scan is processed by normalizing the data to fit into an occupancy grid of a specific resolution just as in the previous algorithm. The potential positions for the sensor become each empty entry in the occupancy grid. Certain thresholds are set as parameters to the function to keep the sensor a specific distance from the floor, ceiling and walls, but other than this every voxel is potentially a next-best-sensor position as long as it has been designated by the preprocessing as empty.

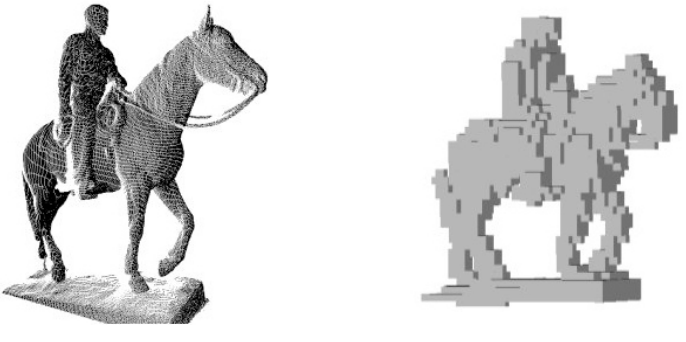
Every potential position is considered. To determine the optimal sensor angle for each potential position, the sphere about the potential point is carved into 36 equally spaced vertical slices. The unknown voxels are projected into the individual slices, j, using the ray tracing algorithm from the previous section and generating a count K_j. The total number of voxels viewable, D_j, from a scanner center in slice j is

$$D_j = \sum_{j=-4}^{4} K_j. \tag{6}$$

The final pose angle, $\alpha = j\pi/9$, corresponds to the j with maximum D_j value. The NBV is the potential position and final pose angle with greatest value.

5 Experimental Results

Both exterior and interior scanning algorithms were tested on a standard 2.0 GHz Intel Core Duo MacBook Pro with 2 GB of memory. To find the NBV outside of an object, a statue known as "Will Rogers and Soapsuds" was used which stands at just under 10 feet tall. The statue's three scanned data sets had an average of 91,207 points per scan. The preprocessing of three sets of this data into a occupancy grid took little time (on the order of 2-3 seconds) so it is negligible in the analysis of the algorithm. The images of both the registered point cloud of the statue and the voxel representation in the NBV exterior algorithm can be seen in figure 2.



(a) (b)

Fig. 2. (a) Point representation of "Will Rogers and Soapsuds". (b) The voxel representation of the statue.

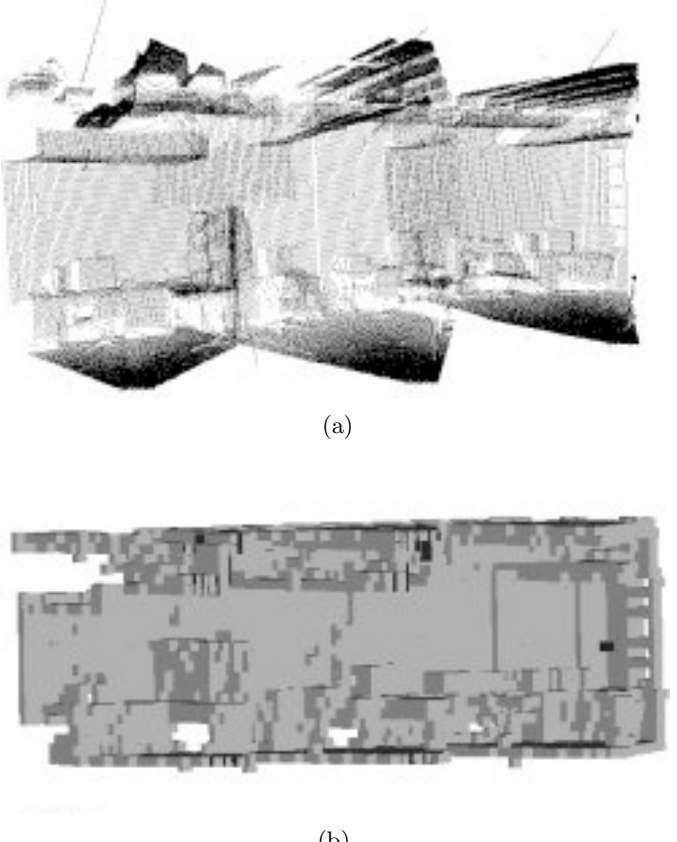
The exterior NBV algorithm was tested at different occupancy grid resolutions starting with 1 square foot voxels. The results are shown in table 1. Figure 4(a) is a horizontal representation of a single level of the voxel representation of the statue which displays the occupied, empty and unknown regions. Figure 4(b) is the same region after the NBV algorithm has been applied and the new position's

Table 1. Exterior NBV algorithm resolutions and times

Voxel Size	Number of Voxels	Time to Compute
1' x 1' x 1'	280	0.21 seconds
6 " x 6 " x 6"	1,960	0.77 seconds
3 " x 3 " x 3 "	15,680	2.90 seconds
1.5 " x 1.5 " x 1.5 "	125,540	11.51 seconds

Table 2. Interior NBV resolutions and times

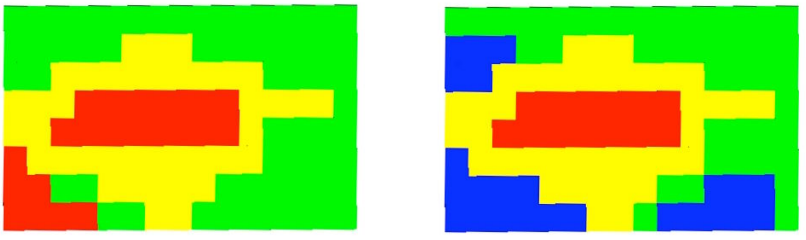
Voxel Size	Number of Voxels	Time to Compute
4' x 4' x 4'	135	0.05 seconds
2' x 2' x 2'	1,020	5.36 seconds
1' x 1' x 1'	8,160	90.68 seconds
6" x 6" x 6"	65,280	1554.80 seconds



(a)

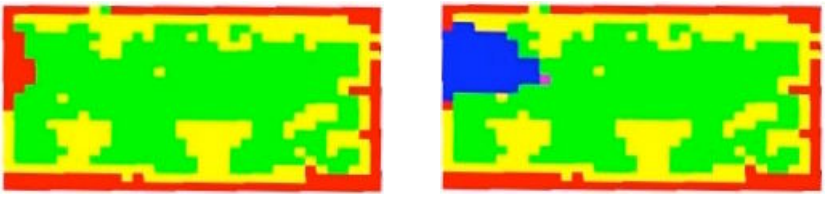
(b)

Fig. 3. (a) West wall of the room used rendered as a point cloud, (b) rendered as a voxel grid



(a) (b)

Fig. 4. A horizontal slice of the topdown view of the statue with a 6" x 6" x 6" resolution. (a) Without a new NBV carved out, yellow represents occupied voxels, green represents empty and red represents unknown. (b) With the carved out new NBV represented by the blue voxels.



(a) (b)

Fig. 5. A horizontal slice of the top down view of the room in voxel form with a 6" x 6" x 6" resolution. (a) The original scans with yellow representing occupied, green representing empty, and red representing unknown voxels. (b) The same representation with the new sensor position in purple and the carved out region in blue.

viewable area has been carved out. Notice how the new sensor position originates at the red voxel region.

To find the NBV inside of a model, scans of a research lab were used. The lab measures 34.25' x 19.5' x 11.67' and has multiple obstructions such as desks, computers and partitions. The eight data sets contained an average of 30,617 points per scan. Figure 3(a) shows the representation of the west wall of the room as a point cloud and figure 3(b) is its voxel counterpart.

The interior NBV algorithm was tested at different resolutions starting at a voxel size of 4 square feet. The results are displayed in table 2. Figure 5(a) shows a top down view of a slice of the entire room just as with the statue. Figure 5(b) shows the room with the selected NBV carved out and the sensor position displayed.

The tables of results reflect the fact that the interior NBV algorithm was far slower and more computationally expensive than the exterior algorithm. This is because the interior algorithm did not have the ability to limit the angles of the

search as the exterior method did. The interior considered a full 2π scan at each potential voxel position whereas the exterior algorithm need only consider the angles that limit the scan to the topmost, bottom-most, leftmost, and rightmost portions of the model that is to be scanned. Since the purpose of the interior algorithm was to map the entire surface, this 2π scan was a necessity.

6 Conclusions and Future Work

Both the exterior and interior NBV algorithms produced realtime solutions, especially when contrasted against the time required to move the scanner and perform a scan. For the exterior NBV algorithm, a fine resolution of 1.5 square inch voxels was used and the algorithm only required 11.51 seconds. Both algorithms were scalable with respect to resolution. At all resolution levels, both a similar sensor position and a similar pose were returned for each test scenario.

For future improvements on this algorithm, considerations of the location of unknown area should be taken so that a general area of interest for the new positions can be found. This would help lower the number of searched positions and dramatically reduce the computation time. The use of octrees to represent the voxel grids would also seem to be an improvement in that it would dramatically reduce the amount of empty voxels that are intersected during the ray tracing steps. Code optimizations for fixed point integers and GPU optimizations can also be used to increase the speed of the computations.

References

1. Scott, W.R., Roth, G., Rivest, J.F.: View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys **35** (2003) 64 – 96
2. Wong, L.M., Dumont, C., Abidi, M.A.: Next best view system in a 3-D object. In: International Symposium on Computational Intelligence in Robotics and Automation Proceedings. (1999) 306 – 311
3. Banta, J.E., Abidi, M.A.: Autonomous placement of a range sensor for acquisition of optimal 3-D models. In: Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation. (1996) 1583 – 1588
4. Pito, R.: A sensor-based solution to the "next best view" problem. 13th International Conference on Pattern Recognition **1** (1996) 941 – 945
5. Garcia, M.A., Velaquez, S., Sappa, A.D.: A two-stage algorithm for planning the next view from range images. In: 9th British Machine Vision Conference. (1998) 720 – 729
6. Dissanayake, M.W.M.G., Newman, P., Clark, S., Durant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation **17** (2001) 229 – 241
7. Surmann, H., Nuchter, A., Hertzberg, J.: An autonomous mobile robot with a 3-D laser range finder for 3-D exploration and digitalization of indoor environments. Robotics and Autonomous Systems (2003) 181 – 198

8. Gonzalez-Banos, H.H., Latombe, J.C.: Navigation strategies for exploring indoor environments. The International Journal of Robotics Research **21** (2002) 829 – 848
9. Sequeira, V., Ng, K., Wolfart, E., Goncalves, J., Hogg, D.: Automated 3D reconstruction of interiors with multiple scan-views. In: SPIE 11th AnnualSymposium. (1999)
10. Papadopoulos-Orfanos, D., Schmitt, F.: Automatic 3-D digitization using a laser rangefinder with a small field of view. In: First International Convference on Recent Advances in 3-D Digital Imagaging and Modeling. (1997) 60 – 67

An Experiential Approach to Interacting with Biological Information

Naureen Moon, Bibek Dev Bhattarai, and Rahul Singh

Department of Computer Science, San Francisco State University
San Francisco, CA, USA
{numoon, bdb}@sfsu.edu, rsingh@cs.sfsu.edu

Abstract. Technological advancements in the life sciences have enabled biologists to generate high volumes of heterogeneous, multimedia data. The challenge today lies in correlating and integrating the information in ways that promote a holistic understanding of the underlying biological phenomena. This paper presents our research in designing an experiential information interaction environment for query-exploration of complex biological information. This entails development of a unified presentation-query-exploration environment that incorporates and relates multimodal data repositories and views. Specifically, our interface captures textual, sequence, and structural views of biological entities and presents semantic correlations between them using Gene Ontology annotations. Additionally, the system extracts and displays the spatial-temporal characteristics of the data to facilitate querying and discernment of relationships. The different views of the data are interactive, in order to facilitate information exploration and assimilation. Experiments and examples demonstrate the system's features, efficacy, and ability to facilitate both concept discovery and information querying.

1 Introduction

Recent technological advancements in the life sciences have enabled researchers to generate vast quantities of data about various bio-chemical entities and phenomena from genes to therapeutic drugs. These technologies have also induced a major paradigm shift in biological research. Gone are the days of detailed analysis focused on a few entities in isolation. Instead the research paradigm today seeks a more holistic view and is based on a multitude of experimental methodologies which probe the domain of interest from multiple perspectives. The various types of information generated as a consequence present interesting challenges towards development of systems meant to support querying, interaction, exploration, and assimilation of such data. This challenge has striking overlaps with issues in management of physical and logical heterogeneity which is currently a central focus of the multimedia and database research communities. Consider for example the problem of developing new therapeutics. The necessary research today requires among others: (a) exploration of the available literature in the area to ascertain, for instance, disease-gene relationships, (b) exploration/query/analysis of the associated genes, and (c) research

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 678 – 687, 2006.
© Springer-Verlag Berlin Heidelberg 2006

involving structural information associated with the disease (such as structure of proteins, enzymes, and small molecules associated with the disease). Even though the aforementioned steps cover but a portion of the associated research spectrum, facilitating these steps requires solving a significant set of critical problems which include:

- Access to multiple repositories each with their own storage structure and logic. Additionally such systems must support handling the large volumes of data present in these repositories.
- Support for interacting with heterogeneous data types, such as textual data, genomic data, and structural data.
- Support for users to assimilate this complex information by exploiting semantic correlations in the data, preserving data and user context, supporting efficient query-retrieval, and aiding in data exploration.

The past few decades have witnessed significant development in databases for life sciences. One comprehensive collection of repositories is the set of NCBI databases [1], which include PubMed, a literature index containing over 15 million citations. NCBI also maintains databases of non-textual information, among them, databases containing protein sequence and structure information, such as Protein and Structure. These databases may be searched, singly or collectively, through the Entrez web query interface [2], whereby results for each database are generated separately and accessible through hyperlinks. The retrieved documents on each results page are also accessible through sets of hyperlinks, and frequently contain links to related information in the other databases.

The development of the NCBI resources has been a significant milestone in supporting access to information related to the life sciences. Further, as part of this effort, paradigms and interfaces to interact with specialized data types (genomic and structural data) have also been developed. However, these solutions only partially address the challenges enumerated earlier. For instance, while the Entrez interface does support unified (keyword-based) querying across multiple repositories, it provides minimal functionality in terms of utilizing semantic correlations that may exist between hits originating from within a single repository as well as those originating from different repositories. This results in deluging a user with data and the user typically has little recourse beyond following each link. The hyperlink-based information traversal strategy often leads to context switching, thus increasing the cognitive load on users [3]. Additionally, there is no effective mechanism by which to search for multiple entities simultaneously to determine relationships or similarities between them. Furthermore, there is no provision to utilize meta-data such as the temporal or spatial characteristics of the information, which can not only be useful to support rapid access [4], but also to explore spatial-temporal and evolutionary trends in the data. Compounding the problem is the sheer volume of data researchers may encounter. For example, a biologist searching for a particular gene on PubMed will often obtain results in the hundreds or thousands. However, it is possible and beneficial to utilize the wealth of data from the NCBI databases as well as others to reveal substantial latent biological information, as we describe below.

2 Research Approach

The paradigm of experiential computing has recently been proposed [3], in multimedia research to develop systems that support assimilation of complex information. Such systems are characterized by: (1) Directness in terms of interaction with the data, (2) Unified query and presentation spaces, (3) Support for user and data contexts, (4) Presentation of information independent of the source, and (5) Facilitation of perceptual analysis and exploration. We note that the characteristics of experiential systems correspond well with many of the issues we have outlined earlier. For instance, relevant biological information can easily reside in multiple sources. Additionally, such information inherently includes interrelationships between entities (e.g., protein interactions and gene-disease connections) as well as multiple modalities (such as textual, genomic, and structural). This makes desirable presentation of data in manners that are source-independent and that simultaneously display and correlate heterogeneous data types. Moreover, the support for information assimilation espoused in experiential systems through direct interactions with the data, context support, and facilitation of perceptual analysis are essential to interacting with complex data such as that in the biological domain.

In this paper, we present the results of our ongoing investigations in developing an experiential system for managing and interacting with information from research in the life sciences. Towards this end, we propose a unified presentation-exploration-query interface that encompasses various modalities of biological data in a reflective manner. Specifically, we provide for a multiple perspective "view" of the information by combining textual data from the literature with Gene Ontology (GO) [5] code assignments and gene/protein sequence data, as well as a graphical representation of relationships and any corresponding structural information (Fig. 3). The various views are *reflective*, that is, that interactions in one view are instantaneously reflected in the other views. This serves not only to mimic real-life interactions (and thereby to allow users to maintain context) but also to provide a perspective on the interrelationships that exist in the information. Additionally, we introduce useful meta-information in the form of spatial-temporal characteristics of the information that display location and publication dates of the retrieved documents. This can be used to support intuitive access to the information as well as analyze the development of knowledge. Finally, our approach uses available manually-assigned GO codes of entities to reveal semantic correlations between them according to a fixed vocabulary of biochemically relevant concepts.

A number of techniques have been developed for automated processing of the literature in order to extract meaning from the vast quantity of available information. These efforts include document clustering [6] and extraction of gene and protein names and interactions [7,8,9,10]. A typical shortcoming of such text-analysis systems, however, is that they are standalone, namely that they do not extend beyond relationship extraction to encompass effective presentation of results, much less interaction. The TransMiner system [11] overcomes this limitation by combining extracted relationships with a graph visualization tool to suggest indirect relationships between entities. While it does include effective visual representation of relationships, interaction is minimal, being limited to user selection of association strength thresholds. The

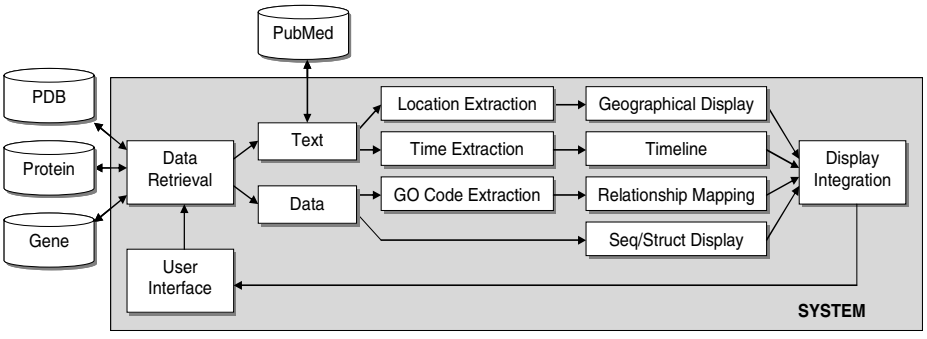


Fig. 1. The system architecture

ALFA architecture [12] enables user-guided information extraction from literature as well as non-textual data in an integrated representation. While it includes an impressive suite of tools for user interaction, its emphasis on user-driven selection of information precludes exploration and full reflectivity of the various components of the interface.

Our system differs most fundamentally from related research due to its combination of attributes, namely multiple perspective information search/exploration (by text (literature), sequence, and structure), support for text-based biological information analysis, inclusion of spatial and temporal characteristics of the information, andsupport for experiential user-information interactions. The proposed approach seeks to underline relationships in the data and facilitate its exploration through reflective interfaces that provide interlinked multiple-perspective views of the information. Such a system can therefore not only be used for query-retrieval, but also for information analysis, and ultimately for hypotheses generation and information exploration. Integrated support of this type is especially necessary given the modern paradigm of systems biology that seeks to span disciplines in an environment of increasing specialization.

3 System Description

The system consists of five major modules: Data Retrieval, Relationship Extraction/Display, GO Code Extraction/ Mapping, Location Extraction/Display, and Time Extraction/Display (Fig. 1). Most data is retrieved through dynamic access of the NCBI databases, PubMed, Gene, and Protein, in particular, using the efetch utilities [13]. Protein structures are obtained from PDB (Protein Data Bank) [14].

The system is initialized through entity identifier-based queries to the user interface. These may be one or more Gene IDs or Protein IDs. The Data Retrieval Module issues the query to the appropriate database using the efetch utility whereby the document for each entity under investigation is obtained. This document generally includes PubMed references for the gene or protein and sequence data as well as the GO codes assigned to gene or protein (when available). Both the relevant PubMed IDs

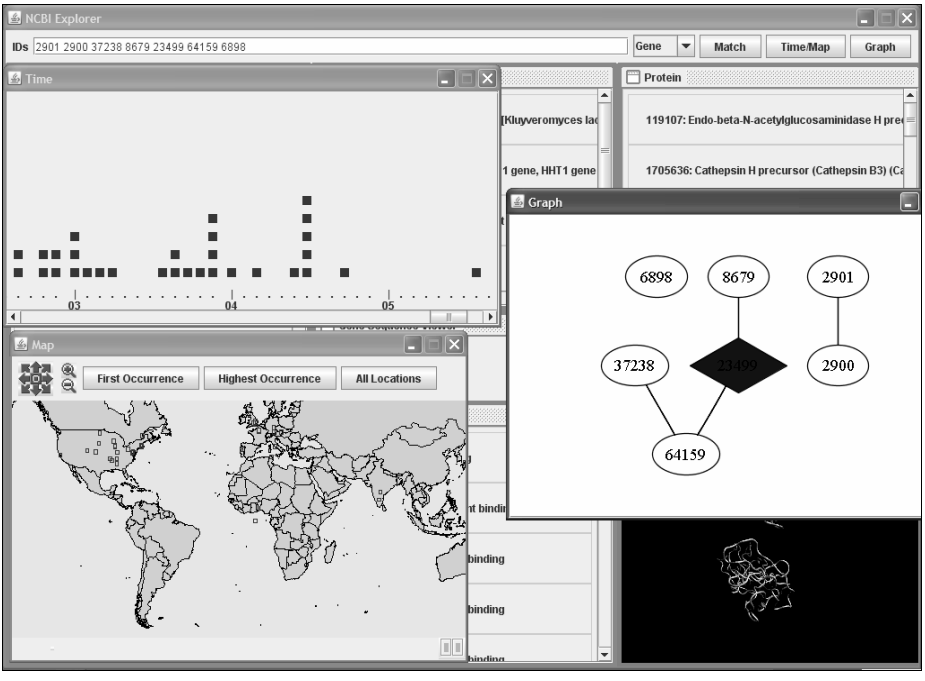


Fig. 2. The interface with spatial, temporal, and relationship visualization overlaid

and GO codes are extracted and stored for display on the user interface as well as for correlating the information about the different genes or proteins. For queries of proteins, a diagram of the protein structure (when available) is obtained from PDB as well.

In the next step, the literature for each entity is accessed using the aforementioned PubMed IDs in conjunction with the efetch utility. The document thus obtained is parsed for information relevant to the views of the data in the UI, namely the title of the literature reference and its spatial-temporal data. Additionally, when a single entity is queried, the MetaMap Transfer Program (MMTx) [15] of the UMLS Metathesaurus is used to map the text to a controlled vocabulary of biological concepts, specifically, to extract names of related genes or proteins for display as a network of relationships (using [16]). When multiple entities are queried, the graph displays each as a node with edges representing GO codes shared by entities (Fig. 2). (It is noteworthy, however, that the visualization may be extended to encompass relationships between different entity types, such as genes and diseases.) Extraction of the spatialcharacteristics of the information requires parsing of the data (specifically, the "affiliation" field) and cross-referencing the terms therein with an index of locations. Once found, the city and country names are used to determine the latitude and longitude values of the location by which they are displayed on the map.

Temporal information of the documents is used to indicate each on the timeline and is, in contrast, much easier to obtain, as it is given in the "date published" field. However, since spatial-temporal information is not fundamental to biological

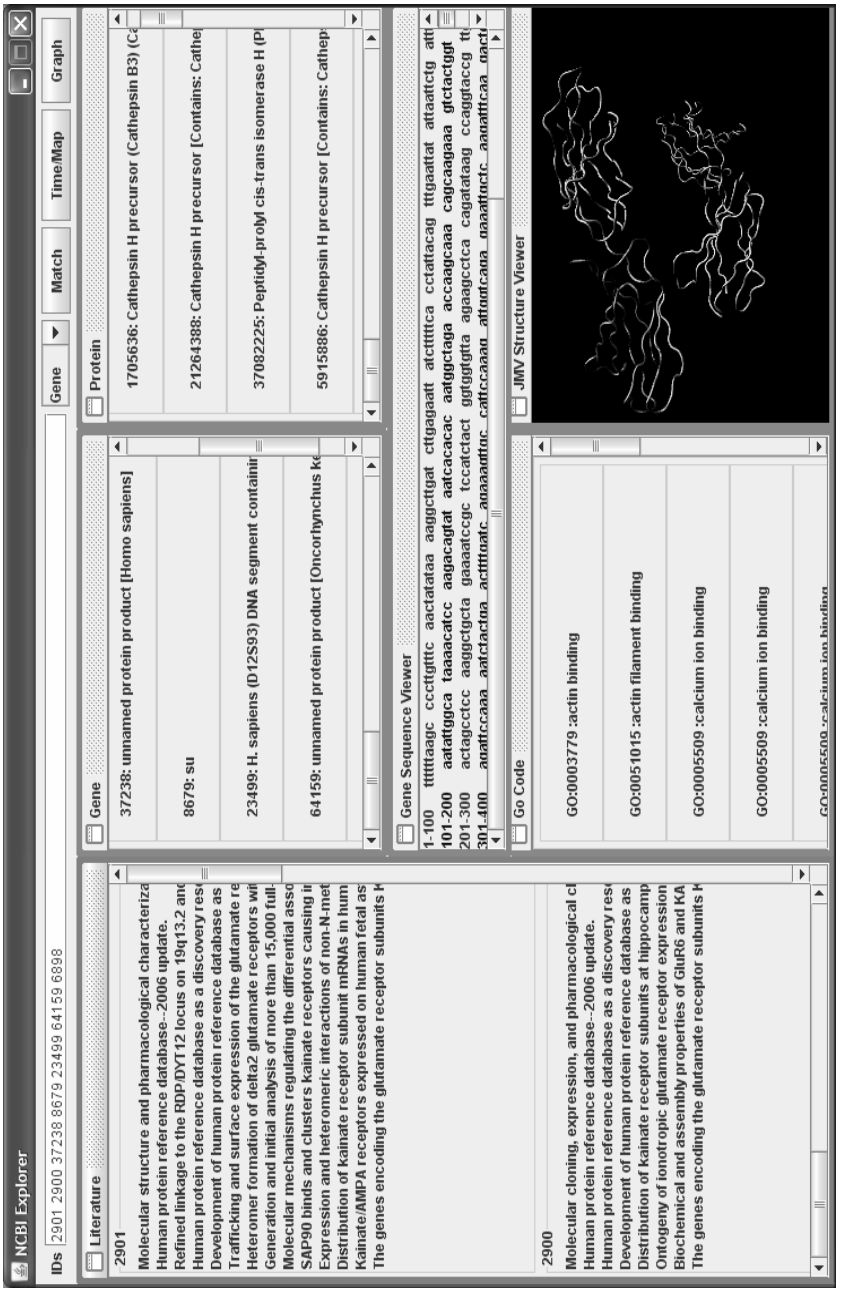


Fig. 3. The unified presentation-query-exploration interface

understanding, the timeline and map views are displayed as pop-up windows rather than having dedicated space on the interface (Fig. 2).

The user interface is populated with various panes displaying a plethora of information about the entities of interest. The leftmost pane lists the entities along with their literature references, as represented by the title of the publication. Elsewhere, the sequence data of the gene/protein is displayed, possibly with the structure as well for proteins (using Java Molecular Viewer [17]). The upper right-hand pane shows, in graphical form, gene-gene or protein-protein relationships suggested by the literature. Lastly, the middle panels contain the GO codes associated with the entities of interest. The lower of these shows the GO codes extracted from the entities, while the higher one lists the entities themselves.

Whether a single or multiple genes or proteins are searched, the graph visualization and displayed GO codes may be used to infer relationships between or co-functionality of entities. As stated above, support for data context entails changes or interactions in one view being reflected in all views. While there are multiple ways that this may be implemented, our system does so mainly by highlighting the information in the other views that corresponds to the selection in the active view. With respect to the GO annotations, for example, selecting an entity in the literature view would show its GO codes only in the lower pane, as well as all of the other entities queried sharing one or more of the codes in the upper pane. Selecting one of the seemingly related entities would serve to highlight those of the GO code(s) it is described by to shed light upon the nature of the similarity or relationship between the entities (Fig. 3) and may be used for functional clustering of entities.

The other views of the data are also reflective. For example, the proteins encoded by the selected gene are highlighted, and its nucleotide sequence displayed in the sequence viewer. In addition, the locations and times corresponding to the selected gene's documents would be highlighted on the map and timeline. Furthermore, selecting an area of the map or timeline would in turn cause the document(s) corresponding to that location or date to be highlighted. This capability allows users to observe spatial-temporal trends in the research.

In accordance with the experiential paradigm, the various features of the system afford users a view of the data from multiple sources and perspectives as well as allowing users to directly interact with the information. As such, users may go beyond simple querying to explore the data and the various trends and relationships present therein. In addition, the unification of the views and query space into a single interface minimizes context switching, thereby easing the cognitive load on the user.

4 Experimental Results

While the system is designed primarily to facilitate exploration of information, some of its components (in particular, the location and time modules) can in fact speed up information extraction. As such, the proposed system was evaluated from two perspectives, that of information discovery and information extraction. The former,

Table 1. The test set of genes, comprising 4 clusters (from [20])

GROUP	GENES	FUNCTION
1	nmda-r1 glur6 ka2 glur1 glur2 glur4 glur3 ka1	Glutamate receptor channels
2	dopamine beta-hydroxylase, tyrosine hydroxylase, phenethanolamine N-methyltransferase, catechol- O-methyltransferase, dopa decarboxylase, monoamine oxidase A, monoamine oxidase B	Catecholamine synthetic enzymes
3	alpha-tubulin, beta-tubulin, dynein, actin, alpha-spectrin	Cytoskeletal proteins
4	tyrosine transaminase, chorismate mutase, prephenate dehydratase, prephenate dehydrogenase	Tyrosine and phenylalanine synthesis

though essentially qualitative in nature, was assessed by demonstrating the system's ability to facilitate information exploration and discovery by presenting an example of display of entity relationships. In contrast, the latter analysis quantifies the contribution of the system's features towards improving performance of information extraction.

In order to meaningfully assess the value of the system vis-à-vis information discovery, it is necessary to compare entity relationships suggested by GO code correlations with known information. As such, we ran a test set of 24 human genes consisting of 4 well-defined functional clusters (Table 1) to gauge the degree to which genes within the same cluster appear more related than those between clusters.

The results (Table 2) seem erratic at first glance. Genes in the first cluster have, on average, over 10 GO codes in common with other genes in the same cluster, while having nearly zero GO codes in common with genes outside of the cluster. Conversely, the remaining three clusters show no such difference, with average values ranging between zero and one codes in common both within and outside the cluster. These values, however, point not to a lack of coherence within clusters, but rather to a paucity of data. It is noteworthy that no definitive negative correlation is observed. When available, the data may indicate strong relationships between entities, however, the approach suffers from the lack thereof.

In the second evaluation, the performance of the system in the realm of information extraction was evaluated by comparison with searches of the NCBI databases through Entrez. In particular, four queries were formulated with information goals constructed in such a way as to gauge the efficacy of the map and timeline modules, with performance determined by two methods: counting the number of mouse clicks and measuring the accession time until the information goal was reached. The results are shown below (Table 3).

Table 2. GO codes as indicators of functional coherence

	Average Number of GO Codes in Common	
Cluster	Within Cluster	Between Clusters
1	10.8	0.4
2	0.8	0.8
3	0.6	0.7
4	0.0	0.2

Table 3. Comparison of information extraction using NCBI and our system

Query	NCBI		Proposed System	
	No. of clicks	Access Time (s)	No. of clicks	Access Time (s)
1	8	67	2	27
2	12	85	2	32
3	2	33	2	29
4	2	46	2	31

The first information goal was, for a given protein, to find a reference that was published in Japan. The second query was similar, but had as its objective the country that produced the most publications about another protein. It is evident that our system is at a clear advantage for queries involving geographical information, with information access accelerated by a factor of 3 to 4 using both time and number of clicks.

The third information goal was to find the number of reference articles published after 2000 about a certain protein, while the fourth was the year in which the most articles were published about the same protein. In this case, there is no clear advantage in speed of information extraction as a result of the timeline module. This is likely due to the fact that the year an article is published is often given along with its title. As such, the temporal information is readily visible and accessible, which is not the case for relevant spatial information. However, despite the ready availability of the date, the timeline provides more intuitive access to the data, and a better sense of its distribution.

5 Conclusion

In this paper, we have presented a novel metaphor to facilitate search of biomedical information. Development of approaches such as these is critical at this juncture in the field due to the proliferation of data. Tools for analysis and synthesis of data have not kept pace with those for its generation, which serves to drastically limit the information that may be derived therefrom. Our approach seeks to aid in the exploration and assimilation of knowledge using an experiential paradigm. As such, users are presented with an interface that combines query and presentation capabilities with interactive views of the data. Moreover, the data itself is derived from various repositories and multimodal in its nature, to further allow the user to "experience" the information from multiple perspectives. Additional features of the system include display of semantic correlations between entities via common GO codes as well as display of the spatial-temporal characteristics of the data. Examples of queries submitted to the interface demonstrate the ability of the system to promote retrieval, exploration, and discovery of information, and thereby illustrate the promise of the approach.

References

1. NCBI, http://www.ncbi.nlm.nih.gov.
2. NCBI Entrez, http://www.ncbi.nlm.nih.gov/Entrez.

3. R. Jain, "Experiential Computing," Communications of the ACM, Vol. 46, No. 7, July 2003.
4. R. Singh, R. L. Knickmeyer, P. Gupta, and R. Jain, "Designing Experiential Environments for Management of Personal Multimedia," ACM Multimedia, 2004.
5. Gene Ontology, http://www.geneontology.org.
6. I. Iliopoulos, A. J. Enright, and C. A. Ouzounis, "TextQuest: Document Clustering of MEDLINE Abstracts for Concept Discovery in Molecular Biology," Pacific Symposium on Biocomputing, 2001.
7. M. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman, "Using BLAST for Identifying Gene and Protein Names in Journal Articles," Gene, Vol. 259, 2000, 245-252.
8. D. Proux, F. Rechenmann, L. Julliard, V. Pillet, and B. Jacq, "Detecting Gene Symbols and Names in Biological Texts: A First Step toward Pertinent Information Extraction," Genomic Informatics Workshop, Vol. 9, 1998, 72-80.
9. K. Fukuda, T. Tsunoda, A. Tamura, T. Takagi et al., "Toward Information Extraction: Identifying protein names from biological papers," Pacific Symposium on Biocomputing, 1998.
10. T. Rindflesch, L. Tanabe, J. Weinstein, L. Hunter et al., "EDGAR: Extraction of Drugs, Genes, and Relations from the Biomedical Literature," Pacific Symposium on Biocomputing, 2000.
11. V. Narayanasamy, S. Mukhopadhyay, M. Palakal, D. A. Potter, "TransMiner: Mining Transitive Associations among Biological Objects from Text," Journal of Biomedical Science, Vol. 11, 2004, 864-873.
12. A. Vailaya, P. Bluvas, R. Kincaid, A. Kuchinsky, M. Creech, and A. Adler, "An Architecture for Biological Information Extraction and Representation," Bioinformatics, Vol. 21, No. 4, 2005, 430-438.
13. NCBI E-Utilities, http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.htm.
14. RCSB Protein Data Bank, http://www.rcsb.org/pdb.
15. A. R. Aronson, "Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program," Proc. American Medical Informatics Association Symposium, 2001.
16. Grappa: A Java Graph Package (AT&T Labs - Research) http://www.research.att.com/~john/Grappa.
17. Java Molecular Viewer (JMV is developed by the Theoretical and Computational Biophysics Group in the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign with NIH support), http://www.ks.uiuc.edu/Research/jmv.

Convex Shapes and Convergence Speed of Discrete Tangent Estimators

Jacques-Olivier Lachaud and François de Vieilleville

LaBRI, Univ. Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France

Abstract. Discrete geometric estimators aim at estimating geometric characteristics of a shape with only its digitization as input data. Such an estimator is multigrid convergent when its estimates tend toward the geometric characteristics of the shape as the digitization step h tends toward 0. This paper studies the multigrid convergence of tangent estimators based on maximal digital straight segment recognition. We show that such estimators are multigrid convergent for some family of convex shapes and that their speed of convergence is on average $\mathcal{O}(h^{\frac{2}{3}})$. Experiments confirm this result and suggest that the bound is tight.

1 Introduction

The problem of estimating geometric quantities of digitized shapes like area, perimeter, tangents or curvatures is an important and active research field. Discrete geometric estimators have indeed many applications in vision, shape analysis and pattern recognition. It is however difficult to compare objectively their respective accuracy, since for a given shape there exists infinitely many shapes with the same digitization. In this paper, we are mainly interested by the *multigrid convergence* property of some estimators, which is one of the few existing objective criteria. This property ensures that a better resolution brings a better approximation. For concrete applications, the *speed of convergence* is a very important criterion too, since it has a huge impact on their accuracy at standard resolutions.

Formally, taking the definitions of [9], let Dig_h be some digitization process of step h. Let $\mathbb{F}$ be a family of shapes in $\mathbb{R}^2$ and let G be a geometric quantity defined for all $X \in \mathbb{F}$. A *discrete estimator of* G is a map that associates to a digitization $\mathrm{Dig}_h(X)$ an estimation of $G(X)$. A discrete estimator $\mathcal{G}$ is *multigrid convergent* toward G for $\mathbb{F}$ and Dig iff, for any $X \in \mathbb{F}$, there exists some $h_X > 0$ for which

$$\forall 0 < h < h_X, |\mathcal{G}(\mathrm{Dig}_h(X)) - G(X)| \leq \tau(h) \ ,$$

where $\tau : \mathbb{R}^+ \to \mathbb{R}^{+*}$ has limit value 0 at $h = 0$. This function defines the *speed of convergence* of $\mathcal{G}$ toward G.

For instance, for the family of plane convex bodies with C^3-boundary and positive curvatures, denoted later on by $\mathbb{F}_c^3$, the best known speed of convergence for an area and or a moment estimator is $\mathcal{O}(h^{\frac{15}{11}})$ [8,10], for a perimeter estimator it is $\mathcal{O}(h)$ [16]. There are fewer results concerning local geometric quantities like

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 688–697, 2006.
© Springer-Verlag Berlin Heidelberg 2006

tangent direction or curvature. The first works on this topic were presented in [4], where some evidence of the convergence of tangent estimators based on digital straight segment (DSS) recognition were given. We report the recent work of [13] which establishes the multigrid convergence of tangent direction estimators based on maximal DSS recognition, with an average speed of convergence of $\mathcal{O}(h^{\frac{1}{3}})$. The recent result of [5] has confirmed that there is yet no curvature estimator proven to be multigrid convergent.

In this paper, we prove a new upper bound for the average speed of convergence of discrete tangent estimators which are based on maximal DSS recognition around the point of interest [6,12]. This new bound of $\mathcal{O}(h^{\frac{2}{3}})$ was suggested by the experimental study of [12] and enhances the previous bound of $\mathcal{O}(h^{\frac{1}{3}})$ [13]. The proof of this enhanced result, obtained for shapes in $\mathbb{F}_c^3$, follows these steps:

Section 2. Digitizations of convex shapes are digital convex polygons (CDP). We achieve thus a better localization of the shape boundary with respect to the digitized boundary (Proposition 1).

Section 3. We first claim that the DSS characteristics have an average asymptotic behaviour depending on their length (Claim 2). The asymptotic edge length of CDP is recalled (Theorem 1), which induces a superlinear localization of the shape boundary in $\mathcal{O}(h^{\frac{4}{3}})$ (Proposition 2). With these relations, the slope of digital edges of CDP is shown to be multigrid convergent to the tangent direction of a nearby boundary point with an average speed of $\mathcal{O}(h^{\frac{2}{3}})$ (Proposition 3).

Section 4. The behaviour of maximal DSS is identical to the behavior of digital edges, since any maximal DSS contains at least one digital edge (Proposition 4). The average speed of convergence of tangent estimators based on maximal DSS is thus achieved (Theorem 3).

We will then conclude and open some perspectives in Section 5.

2 Preliminary Definitions and First Properties

2.1 Digitization and Convex Digital Polygon

Let S be some subset of $\mathbb{R}^2$. Its *Gauss digitization* of grid step h is defined as $\mathrm{D}_h^G(S) = S \cap h\mathbb{Z} \times h\mathbb{Z}$. Thus, the considered digitized objects are subsets of the rescaled digital plane $h\mathbb{Z} \times h\mathbb{Z}$.

A *convex digital polygon (CDP)* Γ is a subset of the digital plane $h\mathbb{Z} \times h\mathbb{Z}$ with a single 4-connected component equal to the Gauss digitization of its convex hull, i.e. $\Gamma = \mathrm{D}_h^G(\mathrm{conv}(\Gamma))$. Its *vertices* $(V_i)_{i=1..e}$ form the minimal subset for which $\Gamma = \mathrm{D}_h^G(\mathrm{conv}(V_1, \ldots, V_e))$ (vertices are ordered clockwise). The points of Γ which are 8-adjacent to some point not in Γ form the *border* of Γ. It is a 4-connected digital path that visits every V_i in order. When moving clockwise, any subpath $\Gamma_{V_i,V_{i+1}}$ is called a *digital edge* of Γ, while the Euclidean straight segment V_iV_{i+1} is an *edge* of Γ.

For small enough grid steps, Gauss digitizations of any shape in $\mathbb{F}_c^3$ are convex digital polygons. This result will be admitted throughout the paper.

2.2 Standard Line, Digital Straight Segment, Maximal Segments

Definition 1. *[15] The set of points (x, y) of the digital plane $\mathbb{Z}^2$ verifying $\mu \leq ax - by < \mu + |a| + |b|$, with a, b and μ integer numbers, is called the* standard line *with slope a/b and shift μ. Standard lines are the 4-connected discrete lines.*

The quantity $ax - by$ is called the *remainder* of the line. Points whose remainder is μ (resp. $\mu + |a| + |b| - 1$) are called upper (resp. lower) leaning points. Any finite connected portion of a standard line is called a *digital straight segment (DSS)*. Its *characteristics* are the slope a/b and the shift μ of the standard line containing it with smallest $|a| + |b|$.

Most of the results demonstrated here are directly transferable for 8-connected curves since there is a natural bijective transformation between standard and naive digital lines. In the paper, all the reasoning is made in the first octant, but it extends naturally to the whole digital plane.

2.3 Recursive Decomposition of DSS

We here recall a few properties about *patterns* composing DSS and their close relations with continued fractions. They constitute a powerful tool to describe discrete lines with rational slopes [2,7]. Without loss of generality all definitions and propositions stated below hold for standard lines and DSS with slopes in the first octant (e.g. $\frac{a}{b}$ with $0 \leq a \leq b$). In the first octant, only two Freeman moves are possible, 0 is a step to the right and 1 is a step up, 4-connected digital paths can be expressed as words of $\{0, 1\}^*$.

Definition 2. *Given a standard line (a, b, μ), we call* pattern *of characteristics (a, b) the word that is the succession of Freeman moves between any two consecutive upper leaning points. The Freeman moves defined between any two consecutive lower leaning points is the previous word read from back to front and is called the* reversed pattern*.*

Since a DSS has at least either two upper or two lower leaning points, a DSS (a, b, μ) contains at least one *pattern* or one *reversed pattern* of characteristics (a, b). There exists recursive transformations for computing the *pattern* of a standard line from the *simple continued fraction* of its slope (see [2], [18] Chap. 4 and [9] Chap. 9), here Berstel approach better suits our purpose.

A rational slope z in $]0, 1]$ can be written uniquely as the continued fraction

$$z = 0 + \cfrac{1}{u_1 + \cfrac{1}{\ldots + \cfrac{1}{u_n}}}$$

and is conveniently denoted $[0, u_1, \ldots, u_n]$. The u_i are called the *partial coefficients* and the continued fraction formed with the k first *partial coefficient* is said to be a *k-th convergent* of z and is a rational numbers denoted by z_k. The

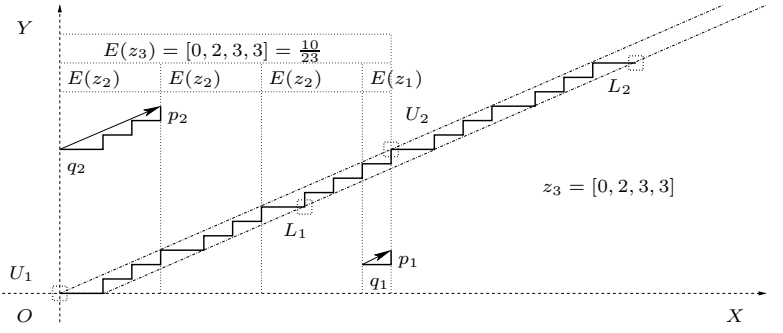


Fig. 1. A digital straight segment of characteristics $(10, 23, 0)$ with an odd depth slope, taken between the origin and its second lower leaning point

depth of a *k-th convergent* equals k. We conveniently denote p_k the numerator (resp. q_k the denominator) of a *k-th convergent*.

We recall a few more relations regarding the way convergents are related and which will be used later on in this paper:

$$\forall k \geq 1 \quad p_k q_{k-1} - p_{k-1} q_k = (-1)^{k+1} \tag{1}$$

$$p_0 = 0 \quad p_{-1} = 1 \qquad \forall k \geq 1 \quad p_k = u_k p_{k-1} + p_{k-2} \tag{2}$$

$$q_0 = 1 \quad q_{-1} = 0 \qquad \forall k \geq 1 \quad q_k = u_k q_{k-1} + q_{k-2} \tag{3}$$

Given a rational slope between 0 and 1 its continued fraction is finite and for each i, u_i is a strictly positive integer. In order to have a unique writing we consider that the last *partial coefficient* is greater or equal to two; except for slope $1 = [0, 1]$.

Let us now explain how to compute the *pattern* associated with a rational slope z in the first octant. Let us define E a mapping from the set of positive rational number smaller than one onto the Freeman-move's words. More precisely: $E(z_0) = 0$, $E(z_1) = 0^{u_1}1$ and the other values are expressed recursively:

$$E(z_{2i+1}) = E(z_{2i})^{u_{2i+1}} E(z_{2i-1}) \tag{4}$$

$$E(z_{2i}) = E(z_{2i-2}) E(z_{2i-1})^{u_{2i}} \tag{5}$$

It has been shown that this mapping constructs the pattern (a, b) for any rational slope $z = \frac{a}{b}$. Fig. 1 illustrates the construction of an odd pattern using the mapping E. The Minkowski $\mathcal{L}^1$ length of $E(z_k)$ equals $p_k + q_k$ and can be computed recursively using Eq. (2) and (3). Moreover we recall that any digital edge is a pattern or a succession of the same pattern, its *digital parameters* are its slope, denoted by $\frac{p}{q}$, and the number δ of repetitions of the pattern $E(\frac{p}{q})$.

2.4 Localization Accuracy of Digitized Convex Shapes

One can expect that the boundary of a convex shape is approximately at distance h from the border of its digitization of grid step h. In fact, for a convex shape

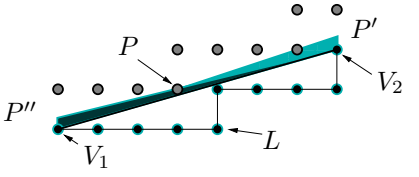


Fig. 2. Two local constraints for the real underlying convex shape

S, its *boundary* ∂S is much closer to the convex hull of $\mathrm{D}_h^G(S)$ than h. A better localization of the shape is thus possible, as stated below and illustrated in Fig. 2:

Proposition 1. *Let S be a convex shape such that $\mathrm{D}_h^G(S)$ is a CDP Γ for some h. Consider an edge V_iV_{i+1} of Γ with slope in the first octant. Then any point of the boundary ∂S above the straight segment V_iV_{i+1} has a vertical distance to it no greater than $\frac{h}{q_{n-1}}$, where the slope of this edge is the irreducible fraction $\frac{p_n}{q_n}$.*

The proof of this proposition can be found in [11].

3 Asymptotic Behaviour of Edges of Digitized Shapes

We study here asymptotic properties of edges of digitized convex shapes. Their average length is first exhibited and an experimental study of the digital parameters of edges is presented. The direction of digital edges is shown to converge toward the tangent direction.

3.1 Asymptotic Number and Digital Parameters of Edges

Let S be some shape of $\mathbb{F}_c^3$. We have the following theorem:

Theorem 1. *(Adapted from [1], Theorem 2) For a small enough h, the Gauss digitization $\mathrm{D}_h^G(S)$ of S is a CDP and its number of edges $n_e(\mathrm{D}_h^G(S))$ satisfies:*

$$c_1(S)\frac{1}{h^{\frac{2}{3}}} \leq n_e(\mathrm{D}_h^G(S)) \leq c_2(S)\frac{1}{h^{\frac{2}{3}}}$$

where the constants $c_1(S)$ and $c_2(S)$ depend on extremal bounds of the curvatures along S. Hence for a disc c_1 and c_2 are absolute constants.

As an immediate corollary, the average Minkowski $\mathcal{L}^1$ length of edges grows as $\Theta(h^{\frac{2}{3}})$. The question is: what is the average behavior of the digital parameters p, q and δ of edges, knowing that the average digital length $l = \delta(p+q)$ satisfies $\Theta(h^{\frac{2}{3}}/h) = \Theta(1/h^{\frac{1}{3}})$? Since the slope of edges should tend toward the slope of points on the boundary of S and since almost all these points have irrational slope, p and q should tend toward infinity almost everywhere with a bounded δ. This observation is confirmed by experiments, as illustrated on Fig. 3, which plots the means and standard deviations of $\frac{q}{l}$ and δ for edges on finer and finer digitizations of a disk. It is clear that q (but also p) satisfies the same asymptotic law as l while δ remains bounded on average. We make hence the following claim:

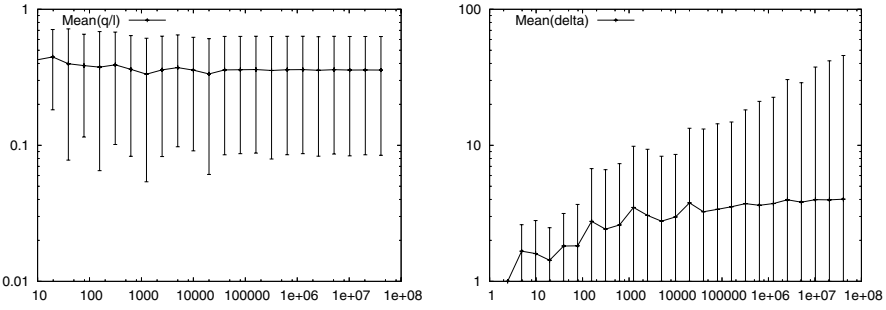


Fig. 3. Plots in log-space of the means of digital parameters $\frac{q}{l}$ (left) and δ (right) for the edges of Gauss digitizations of a disk, as functions of the inverse $\frac{1}{h}$ of the grid step. Standard deviations are symbolized with errorbars.

Claim 2. *On average, the digital parameters p, q of the edges of* $\mathrm{D}_h^G(S)$ *with* $S \in \mathbb{F}_c^3$ *grow as* $\Theta(\frac{1}{h^{\frac{1}{3}}})$*, whereas* δ *is bounded, when h tends toward 0.*

This claim induces a first result about the convergence speed of the localization constraints of Proposition 1 (q_{n-1} has the same asymptotic law as q_n).

Proposition 2. *Let* $S \in \mathbb{F}_c^3$ *and* $\mathrm{D}_h^G(S)$ *its Gauss digitization. For a small enough h, in each octant, the vertical distance between any edge of* $\mathrm{D}_h^G(S)$ *and* ∂S *is bounded. On average, this bound is* $\mathcal{O}(h^{\frac{4}{3}})$ *(and never worse than* $\mathcal{O}(h)$*).*

3.2 Convergence of Tangent Estimation Based on Edge Direction

Let M be a point on the boundary of a shape $S \in \mathbb{F}_c^3$ and let $\theta_M(S)$ be the tangent direction of ∂S at M. We propose to estimate $\theta_M(S)$ as the arctangent of the slope of the digital edge of the CDP $\mathrm{D}_h^G(S)$ lying below M (for the first octant; definitions for other octants are straightforward). We denote this discrete estimator by $\mathcal{T}_M^{\mathrm{edge}}$. Assuming Claim 2, we have the multigrid convergence of this tangent estimator:

Proposition 3. *Let M be a point in the plane and let* $\mathbb{F}_c^3(M)$ *be the subset of shapes S of* $\mathbb{F}_c^3$ *with* $M \in \partial S$*. The discrete estimator* $\mathcal{T}_M^{\mathrm{edge}}$ *at point M is multigrid convergent toward the tangent direction* θ_M *for* $\mathbb{F}_c^3(M)$ *and Gauss digitization. Furthermore, its speed of convergence is on average* $\mathcal{O}(h^{\frac{2}{3}})$*.*

Proof. We have to prove that for a shape $S \in \mathbb{F}_c^3(M)$, there exists some grid step h_S for which $\forall 0 < h < h_S, |\mathcal{T}_M^{\mathrm{edge}}(\mathrm{D}_h^G S) - \theta_M(S)| \leq \tau(h)$. Without loss of generality, we assume that the tangent direction $\theta_M(S)$ is in the first octant and we locally parameterize ∂S as $(x, f(x))$, setting M as the origin of the coordinate axes ($(x_M, y_M) = (0, 0)$). Let h_0 be the digitization step below which $\mathrm{D}_h^G(S)$ is a CDP.

Since the slope of ∂S around M is in the first octant, there exists some $h_S < h_0$ such that for any $0 < h < h_S$, the vertical straight segment going down from M intersect some edge V_iV_{i+1} of $\mathrm{D}_h^G(S)$. Let δ, p and q be the digital parameters of this edge, let z be its slope $\frac{p}{q}$ and let n be the depth of z (i.e. $z = \frac{p_n}{q_n}$). The horizontal distance between M and either V_i or V_{i+1} is necessarily greater than half the horizontal length of V_iV_{i+1}. We assume V_{i+1} maximizes this distance, without loss of generality. We get:

$$\frac{h\delta q_n}{2} \leq x_{v_{i+1}} \leq h\delta q_n \tag{6}$$

From Proposition 1 we have for any $x \in [x_{v_i}, x_{v_{i+1}}]$:

$$zx - \frac{h}{q_{n-1}} \leq f(x) \leq zx + \frac{h}{q_{n-1}} \ . \tag{7}$$

Inserting Taylor expansion of $f(x)$ about $x = 0$ in Eq. (7) induces:

$$|z - f'(0)| \leq \frac{h}{xq_{n-1}} + \mathcal{O}(x) \ . \tag{8}$$

Setting $x = x_{v_{i+1}}$ in Eq. (8) and using both sides of Eq. (6) gives the inequality

$$|z - f'(0)| \leq \frac{2h}{h\delta q_n q_{n-1}} + \mathcal{O}(h\delta q_n) \ . \tag{9}$$

We notice that $z = \tan(\mathcal{T}_M^{\mathrm{edge}}(\mathrm{D}_h^G(S)))$ and that $f'(0) = \tan(\theta_M(S))$. Moreover, for any angle $s, t \in [0, \frac{\pi}{4}]$, we have $|s - t| \leq |\tan(s) - \tan(t)|$. With these two remarks, Eq. (9) implies:

$$|\mathcal{T}_M^{\mathrm{edge}}(\mathrm{D}_h^G(S)) - \theta_M(S)| \leq \frac{2}{\delta q_n q_{n-1}} + \mathcal{O}(h\delta q_n) \ . \tag{10}$$

Since δq_n is no smaller than half the edge length, it follows from Theorem 1 that the dominant term $\frac{2}{\delta q_n q_{n-1}}$ is at least some $\mathcal{O}(h^{\frac{1}{3}})$ on average. In [13] it is also shown that there is no edge of bounded length as h tends toward 0. Then the right part of Eq. (10) tends toward 0. The multigrid convergence is thus shown. At last, assuming Claim 2, Eq. (10) then induces

$$|\mathcal{T}_M^{\mathrm{edge}}(\mathrm{D}_h^G(S)) - \theta_M(S)| \leq \mathcal{O}(h^{\frac{2}{3}}) \ , \tag{11}$$

which indicates that the average speed of convergence of $\mathcal{T}_M^{\mathrm{edge}}$ is $\mathcal{O}(h^{\frac{2}{3}})$. □

4 Tangent Estimators Based on Maximal DSS Recognition

This section discusses the convergence speed of discrete tangent estimators based on maximal digital straight segment (maximal segment) recognition. Along a digital path, maximal segments are the inextensible digital straight segments, otherwise said adding the next point to the front or to the back constructs a set of digital points that no standard line contains. The set of all maximal segments

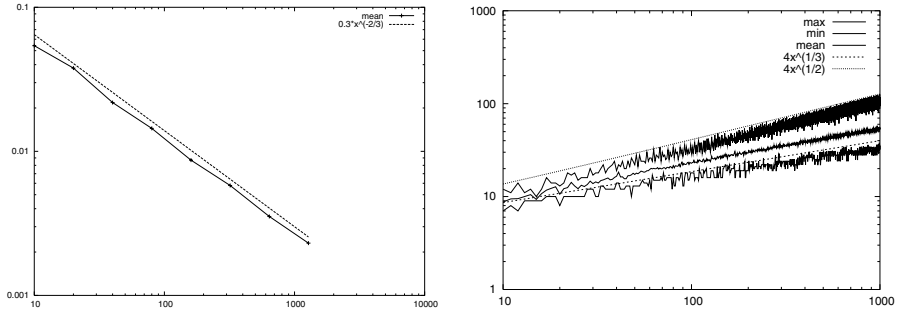


Fig. 4. For both plots, the digitized shape is a disk of radius 1 and the abscissa is the inverse of the digitization step. Left: plot in log-space of the mean absolute error between the λ-MST tangent direction and the theoretical one (mean taken over all digital points). The convergence speed on this shape is likely to be in $\Theta(h^{\frac{2}{3}})$. Right: plot in log-space of the digital length of maximal segments, which is on average $\Theta(1/h^{\frac{1}{3}})$.

of a digital path can be extracted efficiently in time linear with its number of points [6]. Maximal segments have deep links with edges of convex hulls [5,14].

Estimating the tangent direction at some point is then achieved by considering specific DSS [17] or maximal segments [6] containing this point. A recent experimental evaluation [12] has shown that tangent estimators based on maximal segments are accurate and preserve convexity properties of the real shape.

Let M be a point on the boundary of a shape $S \in \mathbb{F}_c^3$ and let $\theta_M(S)$ be the tangent direction of ∂S at M. We propose to estimate $\theta_M(S)$ as the arctangent of the slope of any maximal segment of the CDP $\mathrm{D}_h^G(S)$ lying below M (for the first octant; definitions for other octants are straightforward). We denote this discrete estimator by $\mathcal{T}_M^{\mathrm{ms}}$. We shall prove in Theorem 3 that this estimator is multigrid convergent with average convergence speed of $\mathcal{O}(h^{\frac{2}{3}})$. As a corollary, the Feschet-Tougne tangent estimator [6], which uses the most centered maximal segment around M, and the λ-MST estimator [12], which makes a convex combination of the directions of all maximal segments around M, are also multigrid convergent with same average speed.

Before proving this theorem, note first that the experimental evaluation of the λ-MST estimator, whose absolute error is displayed in Fig. 4, left, suggests that its average convergence speed is indeed likely to be in $\mathcal{O}(h^{\frac{2}{3}})$. Secondly, the average Minkowski length of maximal segments behave as the average length of the edges, that is in $\mathcal{O}(h^{\frac{2}{3}})$ as shown on Fig. 4, right (the Minkowski length is h times the digital length). Thirdly, every maximal segment contains a digital edge as stated in the following proposition whose proof can be found in [11]:

Proposition 4. *On the border of a CDP Γ, any maximal segment, whose slope is some $\frac{p_n}{q_n}$, contains at least one digital edge that has either the same slope or its $n-1$-convergent $\frac{p_{n-1}}{q_{n-1}}$.*

We may now state our main result, which relies also on Claim 2.

Theorem 3. *Let M be a point in the plane and let $\mathbb{F}_c^3(M)$ be the subset of shapes S of $\mathbb{F}_c^3$ with $M \in \partial S$. The discrete estimator $\mathcal{T}_M^{\text{ms}}$ at point M is multigrid convergent toward the tangent direction θ_M for $\mathbb{F}_c^3(M)$ and Gauss digitization. Furthermore, its speed of convergence is on average $\mathcal{O}(h^{\frac{2}{3}})$.*

Proof. We take the same notations as in the proof of Proposition 3. We consider a maximal segment MS below M with slope z_n. According to Proposition 4, it contains a digital edge VV' of slope z_n or z_{n-1}, which may not be below M. Worst case is for a digital edge with slope z_{n-1} and this is the one considered below. We have:

$$\begin{aligned}|\mathcal{T}_M^{\text{ms}}(S) - \theta_M(S)| &\leq |z_n - f'(0)| \\ &\leq |z_{n-1} - f'(0)| + |z_n - z_{n-1}| \\ &\leq |z_{n-1} - f'(x)| + \mathcal{O}(x) + \tfrac{1}{q_n q_{n-1}} \ ,\end{aligned}$$

using Taylor relation and Eq. (1) to get the last inequality.

If we choose some abscissa x such that $(x, f(x))$ is above the edge VV', then the proof of Proposition 3 indicates that $|z_{n-1} - f'(x)| \leq \mathcal{O}(h^{\frac{2}{3}})$ on average. Since $(x, f(x))$ is above MS too, x cannot be greater than the length of MS, which is also some $\mathcal{O}(h^{\frac{2}{3}})$ on average. At last, Claim 2 similarly provides $\frac{1}{q_n q_{n-1}} = \Theta(h^{\frac{2}{3}})$. Summing all these bounds concludes the proof. □

As an immediate corollary, the λ-MST estimator [12] and the Feschet-Tougne tangent estimator [6] have the same asymptotic behaviour.

5 Conclusion

We have studied properties of digital convex polygons and exhibited several new results (Proposition 1 and Proposition 4). We have also examined the asymptotic properties of digital edges on digitized convex shapes (Claim 2), which has led to a position estimator of average convergence speed $\mathcal{O}(h^{\frac{4}{3}})$ (Proposition 2) and to a tangent direction estimator of average convergence speed $\mathcal{O}(h^{\frac{2}{3}})$ (Proposition 3). At last we have shown the new bound of $\mathcal{O}(h^{\frac{2}{3}})$ for the average convergence speed of tangent estimators based on maximal segments (Theorem 3), which matches experimental evaluation.

These results indicate that curvature estimators relying on surrounding DSS have most probably an error of $\mathcal{O}(h^0)$: position $\equiv f(x)$ in $\mathcal{O}(h^{\frac{4}{3}})$, tangent direction $\equiv f'(x)$ in $\mathcal{O}(h^{\frac{2}{3}})$, curvature $\equiv f''(x)$ probably in $\mathcal{O}(1)$ (uncertainty on tangent further divided by $\mathcal{O}(h^{\frac{2}{3}})$). The problem of exhibiting a multigrid convergent curvature estimator is thus still open.

Another straightforward extension of this work would be to investigate the properties of discrete surfaces and estimators based on digital plane recognition. However since the problem of finding an enclosing polyhedron with a minimal number of 2-facets has been proven to be NP-hard (see [3]), the problem would get much harder than the two dimensional case.

References

1. A. Balog and I. Bárány. On the convex hull of the integer points in a disc. In *SCG '91: Proceedings of the seventh annual symposium on Computational geometry*, pages 162–165. ACM Press, 1991.
2. J. Berstel and A. De Luca. Sturmian words, lyndon words and trees. *Theoret. Comput. Sci.*, 178(1-2):171–203, 1997.
3. V.E. Brimkov. Discrete volume polyhedrization is np-hard. Research Report CITR-TR-179, Centre for Image Technology and Robotics, University of Auckland, New Zealand, 2006.
4. D. Coeurjolly. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. PhD thesis, Université Lumière Lyon 2, France, 2002.
5. F. de Vieilleville, J.-O. Lachaud, and F. Feschet. Maximal digital straight segments and convergence of discrete geometric estimators. In *14th Scandinavian Conference on Image Analysis*, LNCS 3540, pages 988–997. Springer-Verlag, 2005.
6. F. Feschet and L. Tougne. Optimal time computation of the tangent of a discrete curve: application to the curvature. In *Discrete Geometry and Computer Imagery (DGCI)*, LNCS 1568, pages 31–40. Springer Verlag, 1999.
7. G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. Oxford University Press, fourth edition, 1960.
8. M. N. Huxley. Exponential sums and lattice points. *Proc. of London Math.Soc.*, 60:471–502, 1990.
9. R. Klette and A. Rosenfeld. *Digital Geometry - Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco, 2004.
10. R. Klette and J. Žunić. Multigrid convergence of calculated features in image analysis. *Journal of Mathematical Imaging and Vision*, 13:173–191, 2000.
11. J.-O. Lachaud and F. de Vieilleville. Convex shapes and convergence speed of discrete tangent estimators. Research Report RR-1398-06, LaBRI, University Bordeaux 1, Talence, France, 2006.
12. J.-O. Lachaud, A. Vialard, and F. de Vieilleville. Analysis and comparative evaluation of discrete tangent estimators. In E. Andrès, G. Damiand, and P. Lienhardt, editors, *Proc. Int. Conf. Discrete Geometry for Computer Imagery*, LNCS 3429, pages 240–251. Springer-Verlag, 2005.
13. J.-O. Lachaud, A. Vialard, and F. de Vieilleville. Fast, accurate and convergent tangent estimation on digital contours. *Image Vis. Comput.*, 2006. In revision.
14. H. Reiter-Doerksen and I. Debled-Rennesson. Convex and concave parts of digital curves. In *Dagstuhl Seminar "Geometric Properties from Incomplete Data"*, March 2004.
15. J.-P. Réveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique*. Thèse d'etat, Université Louis Pasteur, Strasbourg, 1991.
16. F. Sloboda, B. Zaťko, and J. Stoer. On approximation of planar one-dimensional continua. In R. Klette, A. Rosenfeld, and F. Sloboda, editors, *Advances in Digital and Computational Geometry*, pages 113–160, 1998.
17. A. Vialard. Geometrical parameters extraction from discrete paths. In *Proc. DGCI'96*, volume 1176 of *LNCS*, pages 24–35. Springer, 1996.
18. K. Voss. *Discrete Images, Objects, and Functions in* $\mathbb{Z}^n$. Springer-Verlag, 1993.

Towards Obtaining an Ideal Real Time Panoramic Video

Harsh Dhand and Lakshmi Pavan Daggubati

Philips Research
Bangalore, India

Abstract. Panoramic image construction is a well known image processing technique, which allows capturing of a large field of view in one single image or video. Panoramic video construction from videos captured from multiple cameras pose interesting challenges to the existing panoramic image construction algorithms. Computation time and quality of video output are major issues in panoramic videos. While the time of computation is required to meet basic real-time constraints of video streams, the quality should be good enough such that the stream can be used for the purposes intended. In this work, we quantify the various stages in panoramic image construction. We propose various intra-frame and inter-frame optimizations on the top of basic panoramic image construction algorithm and a methodology to achieve real-time panoramic videos.

1 Introduction

The field of view of normal cameras is limited to a maximum of 90 degrees in the horizontal plane. Applications like surveillance, scene activity detection and driver assistance in automobiles require a larger field of view, typically between 180 and 360 degrees. In such scenarios, multiple cameras are used. The video/images captured by these separate cameras could be either used separately for scene activity detection/object detection or a panoramic video/image could be constructed by stitching together the multiple videos/images. Method one may miss out on some scene activity or object that is spread across (captured partially by) different cameras. A good panoramic video will solve this problem and standard object detection or scene activity detection algorithms can be used directly on it.

Panoramic image construction is a well known image processing technique, which allows capturing of a large field of view in one single image. Often, panoramic video is referred to as the combination of panoramic images taken using a single camera that is swept through 360-degree angle. However, the panoramic videos that we are referring to here, correspond to stitching videos obtained from two or more different cameras, which are capturing some overlapping areas simultaneously. Panoramic video construction from videos captured from multiple cameras pose interesting challenges to the existing panoramic image construction algorithms. While the computation time should be such that the basic real-time constraints of video streams (e.g. output at 30 frames per

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 698–708, 2006.
© Springer-Verlag Berlin Heidelberg 2006

second) are met, the quality should be good enough such that the stream can be used for the purposes intended e.g. driver assistance, video surveillance, etc.

To understand the details of optimizations that we propose in panoramic video stitching, it is essential to know the methods that are typically followed in fully automatic panoramic image creation. Panoramic image construction can be thought of as an intelligent overlap of images. There is a requirement for calculating a homography matrix [1] [2], which is a 2D image warp to represent a 3D reprojection required in bringing pictures from different planes onto the same one. Image warping using homography matrix can be done in two ways:

1. **Using feature points:** Computation of homography matrix using features in the image, e.g. Harris corners can be extracted and normalized cross-correlation of local intensity values can be used to match them. Homography matrix can be estimated from the obtained feature points correspondences using RANSAC scheme [3].
2. **Direct methods:** Iterative estimation of homography matrix by minimizing error function based on intensity difference in area of overlap. This method depends on the complete image data and hence is accurate. But is also dependent on fragile brightness constancy assumption and is also computationally expensive.

Images typically have hundreds of feature points. Feature point matching and homography matrix estimation are two computationally intensive tasks. In this work, we quantify the various stages in panoramic image construction using a variety of methods. We propose various intra-frame and inter-frame optimizations on the top of basic panoramic image construction algorithm and a methodology to achieve real-time panoramic videos.

The rest of the paper is organized as follows. Section 2 deals with the related work. It serves to introduce the commonly used parlance for panoramic mosaicing and explains current day technology. Section 3 details the optimizations done for obtaining real-time videos. Section 3 shows some of the results, while section 4 compiles all the results together. Section 5 concludes discussing our future work.

2 Related Work

Panoramic image construction has been studied extensively and implemented in commercial products like the digital cameras, which allow the user to pan the camera across a scene and produce a stitched image. A number of different approaches and improvisations have been suggested on top of the basic work of David Browne and Richard Lowe [2]. We will not detail out the technology required in basic panoramic image construction. A good reference for the same is the work of Richard Szeliski [1].

We will focus on two key things in this section. The first being the various approaches suggested to speed up the panoramic image construction and second, the requirement of such fast image construction in panoramic video construction. Panoramic videos have been mentioned by many people [4] [5]. What is generally

referred to as a panoramic video is a video composed by joining images which are captured by a camera panning across a scene, and hence consists of different images, which have overlap. Typical applications requiring such a panoramic video consist of capturing the complete beauty of a place or capturing the motion of an athlete etc. An example is illustrated in [4]. Such a panoramic video does not typically have a requirement of construction in real time, since the video is required after the capture of all the images. The panoramic video that we are constructing here, consists of stitching videos that are captured by multiple cameras simultaneously (as shown in figures 1 and 2), and hence there is an implicit requirement of producing the video in real-time.



Fig. 1. Images from two video streams(with some overlap region)

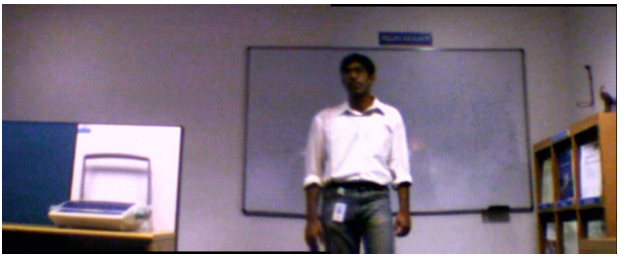


Fig. 2. Image from panoramic video stream

A very close parallel to our work is that done by Sumantra and Amarnath [6]. They develop a method for intergrating data in real-time from multiple camera sources. Another recent work of Udhav et. al. [7] describes the basic issues faced in image mosaicing (stitching). Image alignment, image cut and paste and image blending are identified as three compute intensive steps. They used a hash table to store the coordinates required for transforming images, indexed by the basis points. Prominent works in speeding up panoramic image creation, include one by Irani and Anandan [8], which details out the various methods for image mosaicing, all of which employ the direct methods for matching. They mention techniques like coarse-to-fine iterative estimation, which is based on the fact that given proper filtering and sub-sampling, the induced image motion decreases as we go from full-resolution images to small resolution images. Another work which

deals with quantifying the quality of mosaiced images is that of Anat et. al. [9]. They demonstrate seamless image stitching in the gradient domain by defining several formal cost functions and optimizing them during stitching.

In our work, we quantify the computation times of various stages in panoramic video construction. This helps in evaluating the various optimizations that we have proposed in the process of panoramic video creation.

3 Real Time Panoramic Video Creation and Optimizations

Real-time panoramic video creation is a difficult task due to the computational complexity of the steps involved. We studied the dependence of the quality of image and the computation time on parameters that are used at various stages in the image construction. The optimal values of these parameters are then selected. The optimizations are classified as intra-image (between a set of juxtaposed images to be stitched) and inter-image (between two panoramic images in the panoramic video stream).

3.1 Intra-image Optimizations

The optimizations are explained for each of the steps which are done in panoramic image construction. The numerical values of computation times shown are average values obtained by running the algorithm (with different parameters) for 12 video streams having around 300 frames each. The videos are shot both in internal (lab and living room) and external (open air) conditions with both moving objects and static objects. We implemented the algorithm on a low cost PC (OS: Linux-2.2.4, CPU: Pentium IV - 2.8 GHz with Dual core and RAM: 1024 MB).

1. Image Normalization: In practical situations, intensity levels of two images may vary considerably. To avoid errors in feature point matching and to ensure uniformity in panoramic image, we need to scale the intensity levels of one image w.r.t another. This step involves getting a multiplicative factor for second image w.r.t first image and scaling of intensities with this factor. For a fixed image size, the cost of this remains constant as 4 msec and 10 msec for a 320*240 and 640*480 images respectively (because we are loading the image into memory, which is limited, there is not a direct increase with the image size) and cannot be optimized further. For the overlap region, this intensity normalization results in relatively uniform seam.

2. Feature point extraction and matching: For feature point extraction and matching, conventional techniques takes 4 seconds for 320*240 image and 16 seconds for a 640*480 image [10]. This clearly cannot serve in the construction of a panoramic video. For the purpose of finding the homography matrix, finding the complete set of feature points (usually very large) in each image and then doing matching is clearly an overkill. Feature point selection and feature point matching are regarded as two separate techniques in panoramic image

construction. Since for the case of panoramic image construction, we know that the extraction step leads to the matching, it is possible that they can be put together.

We use a method for feature point selection and matching, which is computationally two orders of magnitude less than the conventional techniques. The method assumes the overlap region to be between 5% and 40% of the image sizes. This is a reasonable assumption made for any panoramic image construction. Now in this region, we select representative blocks from the first image, and use them as the basis to find the closest matching corresponding blocks in the second image. The search algorithm spots the minima in a prescribed range. A correlation function between the reference block from the first image and the corresponding block being searched in the second image is taken as the measure of disparity. Also the search algorithm is a learning one, and adapts its search space based on consistent previous matches, i.e. on the basis of previously obtained positions of matching blocks, the search for new blocks is made adaptively starting from around the region which is most likely to have a correspondence.

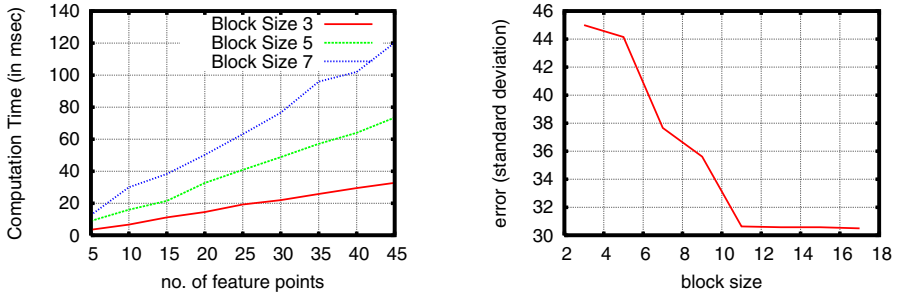


Fig. 3. Computation Times and Error in feature point selection: Image size 320*240

For a correlation function, SAD (Sum of Absolute Differences), figure 3 shows trends of error and computation time for different block sizes and number of feature points. Figure 4 shows average minimal SAD value for various block sizes.

3. Homography matrix estimation: The homography matrix (3*3 matrix with 8 degrees of freedom) determines the transformation of the second image to the plane of the first image and the requisite horizontal and vertical adjustments required in the the second image, such that stitching takes place appropriately. Ideally four point correspondences are sufficient for estimating the homography. With the large number of feature points found in step 2, we can assume an overdetermined system and use singular value decomposition (SVD) to find out the homography matrix [11]. This method (compared to RANSAC [3]) is not only computationally less expensive, but it is also extensible to varying image sizes, and the computation times do not depend on the image sizes. Computation time of the homography matrix depends on the number of feature points selected. But we observed that accuracy stabilizes as we increased the number of feature points selected. For various sets of images considered and for feature points greater than

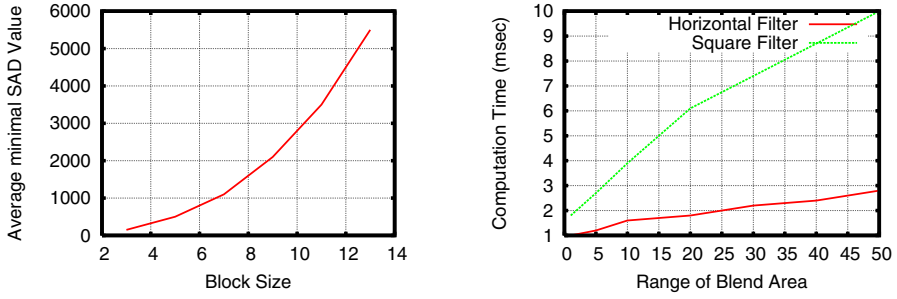


Fig. 4. Average minimal SAD Values and Computation times for Blending: Image size 320*240

20, the homography obtained is the same, and the image obtained is also the same, within 5% deviation. Computation time is of the order of 20 msec (for 20 feature points) and doesn't depend on image size.

4. Homography matrix application: In this step, homography matrix is applied to the pixel coordinates in the second image. It involves matrix multiplication, with the matrix sizes also remaining constant, hence the computation time is constant. It is of the order of 6 msec for 320*240 image size and 23 msec for image size of 640*480. The obtained values from the second image are used to determine the position of the pixel values in the final image. This step of copying the pixel contents onto the panoramic image takes about 7 msec for 320*240 image size and 17 msec for 640*480 image size.

5. Image Blending: Several methods can be used for blending [12] [9]. The requirement for blending arises to smoothen out the edge in the stitched portion. The better the initially computed homography and the image stitching, the lesser intensive need the blending be. For the above implementations we experimented with both a horizontal filter and a square filter. The blending is typically done in a region surrounding the stitched portion, to ensure uniformity. As the size of the region around the seam, in which blending takes place, is decreased, the time of blending also decreased. However, the quality of the blended image (by visual perception) stabilizes at a range of 20 pixels around the seam. A comparison of square and horizontal filters is also done in terms of the computation times for image size 320*240 (as shown in figure 4). The square filter gave better quality stitched images, but is also more expensive.

3.2 Inter-image Optimizations

In this section we explain various inter-image optimizations. Homography matrix depends on relative positioning of cameras. If the cameras are static w.r.t each other, homography matrix needs to be estimated only once and the same can be used for subsequent frames. This is not the case when the cameras have a relative motion between them. However for a small set of continous

frames, feature point matches (hence the homography) are nearly same. A major contribution of our work is a methodology for the identification of the frequency of update of the feature points and homography matrix, such that the video is still obtained at a desired rate. The more often the feature points, and hence the homography are computed, the better will be the image construction. But this is not possible considering the computational cost of these applications. The image normalization and image blending are to take place invariably for all frames in a video, and hence constitute the minimum computation required. The feature point matching and homography matrix estimation can be done selectively ensuring real-time behaviour. Before we describe the formulation to know the minimum number of frames, for which homography should be taken as constant, for real-time behaviour, we briefly describe the steps again, now with respect to inter-frame optimizations.

1. Image normalization: To ensure uniformity of panoramic image, this step has to be performed on every frame. The cost of this constitutes the fixed overhead and should be included in analysis.

2. Feature point extraction and matching: Feature points actually vary across frames, but for the construction of panoramic video construction, where ehe feature points simply serve to tell the homography matrix, there is no requirement to update the feature points with every frame. The step can be performed after every n frames, where n is a number computed using real-time requirements, as explained in section 4.1.

3. Homography matrix estimation: Many a times, relative orientations of cameras can get altered in due time. Hence, it needs to be periodically updated. The approach that we describe in section 4.1, shows how it is possible to know the minimum time after which the homography matrix can be recomputed, without losing the real-time behaviour of the algorithm.

4. Homography matrix application: This is an interesting stage, a clear case of deciding between time versus space complexity. Given a constant homography matrix, we could each time recompute the corresponding values to which the pixels from the second image are mapped. The other approach is to store these values in memory and use the mapping to simply compute the values to be filled in the subsequent panoramic images. The later approach has been used in our algorithm and also for calculations in section 4.1.

5. Image Blending: Image blending is a step which needs to be done for every frame. Hence its computation time also constitutes the fixed overhead in the formulation of finding the frequency of homography computation.

4 Results

Figures 4 and 6 are again from two streams captured by cameras simultaneously and the corresponding frames from the panoramic video. In the example shown, the panoramic video allows us to capture motion to a greater extent.



Fig. 5. Starting pair of frames from Video Streams

4.1 Maximal Computation of Homography

Let the computation times of the various stages described in section 3.1 and 3.2 be denoted as t_i, where i denotes the step number. t_4 (time for homography matrix application) can be further split into t'_4 (time for mapping the coordinates of second image w.r.t axes of first image) and t''_4 (time for filling the pixels values at the mapped locations). As explained above t_1 (normalization), t_5 (blending) and t''_4 are minimal overhead and need to be done everytime. However t_2 (feature point matching), t_3 (homography estimation) and t'_4 can be computed after every n frames (i.e. when the homography matrix needs to be updated). Assuming a good pre-fetch exists, we can take the simplest case of the average time per frame, to be $(t_2 + t_3 + t'_4 + \text{n}^*t_1 + \text{n}^*t''_4 + \text{n}^*t_5\ )$ / n.

This number should be less than the maximum available processing time for a frame for obtaining a real-time output. Let the real-time requirement be 30 frames per second. This implies that for each frame computation time should be less than 33 msec.

$$(n * t_1 + t_2 + t_3 + t'_4 + n * t''_4 + n * t_5)/n \leq 33 \tag{1}$$

The above equation hence gives us a value of n, which would allow us to obtain the real-time behaviour, in this case 30 frames per second. We experimented on many video sequences varying different parameters involved in panoramic image construction. As mentioned in sections 3.1, block size of 11, 20 feature points and square filter blending with range of 20 pixels around the seam are found to be optimal. Average computation times for each step of panoramic image creation are shown in table 1.

From the table, the minimum processing time for each frame $(t_1 + t''_4 + t_5)$ is 17 msec for a 320*240 image and 43 msec for a 640*480 image. For a 320*240



Fig. 6. Frames midway in Video Stream

Table 1. Average computation time (in milliseconds)

stage	time for 320*240	time for 640*480
normalization t_1	4	10
feature selection and matching t_2	100	220
homography estimation t_3	20	20
homography application t_4'	6	23
homography application t_4''	7	17
blending t_5	6	16

image, substituting the values in equation 1, we get $(126 + n * 17)/n \leq 33$ which gives $n \approx 7$.

In the case of 640*480 image, minimum overhead time exceeds the 33 msec. Therefore, using these parameters of block size 11 and square filter blending with range of 20 pixels, 30 frames per second cannot be acheived. For a 15 frames per second constraint, we have $(263 + n * 43)/n \leq 66$ giving $n \approx 11$. However, if one can compromise a little on quality of output video, by using block size of 9 and horizontal filter for blending, real-time constraint of 30 frames per second can be met. In that case, we have $(203 + n * (29)/n \leq 33$ and $n \approx 25$ (i.e. feature points and hence homography are updated at every 25th frame).

5 Conclusions and Future Work

Panoramic video construction (by stitching of overlapping video streams) though difficult, can be acheived in real-time by various intra-image and inter-image optimizations. Computationally intensive steps of feature point matching and homography matrix estimation are done in an intelligent manner to speed up

the overall algorithm significantly so as to make it meet real-time constraints. The quality of the video panorama is not to be compromised in this process. We quantified the computation times of the various stages and estimated the parameters to ensure real-time video creation.

The panoramic video generated, when used in automotive and surveillance applications, serve as the basis for doing more processing on it, such as object detection and tracking, again in real-time. This is actually one of the future activities that would be done by us. Another interesting problem that needs to be handled in real-time is the occlusion problem, which arises when there are multiple planes being captured by the images, and the homography matrix determines which plane is stitched best. In such a case, people have proposed the segmentation of the image, isolated applications of homography and regrouping of objects into the image [13] [14]. Doing this in real-time especially when background clutter is high (as in many real life applications) is difficult. We believe that in the case of panoramic image creation, there could be more intelligent solution to the same problem. This would also be a part of our future work.

References

1. Szeliski, R.: Image alignment and stitching. Technical Report MSR-TR-2004-92, Microsoft Research, Redmond, USA (2004)
2. Brown, M., Lowe, D.: Recognising panoramas. In: Proceedings of Ninth IEEE International Conference on Computer Vision (ICCV 2003). Volume 2., IEEE Computer Society (2003) 1218 – 1225
3. Vincent, E., Laganiere, R.: Detecting planar homographies in an image pair. In: Proceeding of 2nd International Symposium on Image and Signal Processing and Analysis (ISPA01). (2001)
4. Bartoli, A., Dalal, N., Bose, B., Horaud, R.: From video sequences to motion panoramas. In: Proceedings of the IEEE Workshop on Motion and Video Computing, 2002, IEEE (2002) 201–207
5. Yongmin, L., Morrison, G., Nightingale, C., Morphett, J.: Robust panorama from mpeg video. In: Proceedings of 2003 IEEE International Conference on Multimedia and Expo. Volume 1., IEEE Computer Society (2003) 81–84
6. Dasgupta, S., Banerjee, A.: A real-time panoramic vision system for autonomous navigation. In Ingalls, R., Rossetti, M., Smith, J., Peters, B., eds.: Proceedings of the 2004 Winter Simulation Conference. (2004)
7. Bhosle, U., Chaudhuri, S., Roy, S.: A fast method for image mosaicing using geometric hashing. IETE Journal of Research **48** (2002) 237–244
8. Irani, M., Anandan, P.: All about direct methods. In Triggs, W., Zisserman, A., Szeliski, R., eds.: Vision Algorithms: Theory and practice, Springer-Verlag (1999)
9. Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless image stitching in the gradient domain. In: Proceedings of European Conference on Computer Vision (ECCV 2004). Lecture Notes in Computer Science, Springer (2004) 377–389
10. Sun, C., Talbot, H., Ourselin, S., Adriaansen, T.: Automatic image mosaic system using image feature detection and taylor series. In: Proceedings of the Seventh International Conference on Digital Image Computing: Techniques and Applications (DICTA 2003), CSIRO Publishing (2003)

11. Hartley, R.: In defence of the 8-point algorithm. In: Proceedings of the Fifth International Conference on Computer Vision (ICCV 1995), IEEE Computer Society (1995) 1064 – 1070
12. Jia, J., Tang, C.: Eliminating structure and intensity misalignment in image stitching. In: Tenth IEEE International Conference on Computer Vision (ICCV 2005). Volume 2., IEEE Computer Society (2005) 1651–1658
13. Yang, L., Crawfis, R.: A panoramic walkthrough system with occlusion culling. In: Proceedings of the workshop on Virtual environments 2003. ACM International Conference Proceeding, ACM Press (2003) 169–177
14. Bhat, K., Saptharishi, M., Khosla, P.: Motion detection and segmentation using image mosaics. In: Proceedings of 2000 IEEE International Conference on Multimedia and Expo. Volume 3., IEEE Computer Society (2000) 1577–1580

Object Recognition Using Local Descriptors: A Comparison

A. Salgian

Department of Computer Science
The College of New Jersey
Ewing, NJ 08620
salgian@tcnj.edu

Abstract. Local image descriptors have been widely researched and used, due to their resistance to clutter and partial occlusion, as well as their partial insensitivity to object pose. Recently Mikolajczyk and Schmid [1] compared a number of such descriptors and concluded that the SIFT-based ones perform best in image matching tasks. This paper compares the effect that three local descriptors have on object recognition: SIFT [2], PCA-SIFT [3] and keyed context patches [4]. We use a data set containing images of six objects on clean and cluttered backgrounds, taken around the whole viewing sphere. We conclude that keyed context patches perform best overall, but they are outperformed for some objects by the second best feature, PCA-SIFT.

1 Introduction

Many different local image descriptors have been proposed in the computer vision literature, and they have been successful in a wide variety of applications, including object recognition [5], [6], [7], [8], [2], [4]. These descriptors can be computed efficiently, are resistant to clutter and partial occlusion, and are somewhat insensitive to pose, i.e. they change relatively slowly as the view of the object changes.

Research on local image descriptors concentrated mostly on matching image pairs, without looking at their effect on the performance of higher level systems. In this paper we look at how different local image descriptors compare when used in an object recognition system.

One of the most popular and successful local descriptors is SIFT (Scale-Invariant Feature Transform), introduced by Lowe [2]. SIFT features use smoothed weighted histograms of the image gradient. They are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint.

Ke and Sukhtankar [3] introduced a PCA-based representation that uses SIFT keypoints, called PCA-SIFT. Their approach applies principal component analysis to the normalized gradient patch centered on a keypoint, and obtains better results than the original SIFT.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 709–717, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Recently Mikolajczyk and Schmid [1] compared a number of local image descriptors, including SIFT [2] and PCA-SIFT [3], steerable filters [9], differential invariants [10], complex filters [11], moment invariants [12], and cross-correlation for different types of interest regions. They concluded that the ranking of descriptors is mostly independent of the interest region detector, but is dependent on the representation used to model the image patch around the interest point.

Since in their experiments SIFT-based descriptors performed best, we looked at SIFT and PCA-SIFT in the context of object recognition and compared them with the keyed context patches introduced by Nelson and Selinger [13]. These features are centered around prominent contour fragments extracted from the image and, similarly to SIFT features, are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They were designed for the hierarchical object recognition system described in [4] and obtained very good performance.

For each of these three features, given a set of labeled, clean, black background images of six objects taken around the whole viewing sphere (the gallery), and sets of clean and cluttered images of the same objects in different views (the probe set), we try to classify each probe image by finding the gallery image that matches it best. We then look at the percentage of probe images that are correctly classified using these features.

We conclude that while PCA-SIFT performs better than SIFT (a result that we expected based on results in [3], and [1]), the best overall performance is obtained by the system based on keyed context patches. We also note that the results of the SIFT-based methods and the keyed context patches are completely uncorrelated, with PCA-SIFT outperforming the keyed context patches on several objects. Therefore combining the two methods could potentially lead to even better results.

The remainder of the paper is organized as follows. Sections 2, 3 and 4 give brief overviews of the local descriptors that we used. Section 5 describes our experiments in detail and provides an analysis of the results. Finally, we conclude and give directions for future work in Section 6.

2 SIFT

The SIFT method, as described in [2], consists of four major stages:

1. Scale-space extrema detection. Potential interest points that are invariant to scale and orientation are identified by scanning the image over all scales and locations. This is implemented efficiently by using a difference-of-Gaussian function.
2. Keypoint localization. Candidate keypoints are localized to sub-pixel accuracy and eliminated if found to be unstable.
3. Orientation assignment. One or more dominant orientations are assigned to each keypoint location based on local image gradient directions.
4. Keypoint descriptor. A local image descriptor is built for each keypoint, based upon the image gradients in its local neighborhood. This is done by

sampling the magnitudes and orientations of the image gradient in the patch around the keypoint, and building smoothed orientation histograms to capture the important aspects of the patch. A 4x4 array of histograms, each with 8 orientation bins, captures the rough spatial structure of the patch. This 128-element vector is then normalized to unit length and thresholded to remove elements with small values.

For image matching and recognition, SIFT features are first extracted from a set of gallery images and stored in a database. A new image is matched by individually comparing each feature extracted from this image to the features in the database, and finding candidate matching features based on the Euclidean distance between feature vectors. The best candidate match for each keypoint is found by identifying its nearest neighbor in the training database.

Many features in an image will not have a correct match in the database because they arise from background clutter or were not detected in the training images. Discarding these features based on a global threshold on the distance to the nearest neighbor does not perform well, as some descriptors are more discriminative than others. A more effective measure is the ratio between the distance of the closest neighbor and that of the second-closest neighbor.

For object recognition, the gallery image that has the largest percentage of its features matched by features from the incoming probe image is selected as a match. This image will give identity as well as approximate pose information for the object being recognized.

Performance can be improved further by imposing viewpoint consistency constraints between the set of resulting matches to eliminate outliers.

3 PCA-SIFT

The PCA-SIFT algorithm [3] uses the same keypoints as the SIFT method, the only difference being the descriptor itself. Given the sub-pixel location, scale, and dominant orientations of a keypoint, the PCA-SIFT algorithm extracts 41x41 patches at the given scale, centered over the keypoint, and rotated to align its dominant orientation to a canonical direction. It then applies principal component analysis (PCA) to these patches. The descriptor is the eigenspace projection of the image gradient vector of the patch (where the eigenspace was previously computed from other patches).

Matching is performed using Euclidean distance, similarly to SIFT. First, an eigenspace is computed from image gradient vectors of patches extracted at SIFT keypoints from training images. The set of training images may or may not overlap with the set of gallery images. Then, patches are extracted at SIFT keypoints from all the gallery images, and their image gradient vectors are projected onto the eigenspace. A new image is matched by individually comparing each extracted feature to the features stored in the database. A candidate is obtained by finding the nearest neighbor based on Euclidean distance. Candidates whose distance falls above a preselected threshold are discarded. Contrary to SIFT, a

global threshold works better than the ratio between the distances to the closest and second-closest neighbors.

4 Keyed Context Patches

The keyed context patches used in [4] are different in structure and do not use the same keypoints as the SIFT-based methods. However, they are used for recognition in a somewhat similar manner.

First, contours are extracted from images using a stick growing method developed by Nelson [14]. The method uses both gradient magnitude and direction information to extract a set of boundary fragments terminated at corners (regions of high curvature).

In the second stage, keyed context patches are constructed by taking the prominent (i.e. longest) contour fragments (key curves) and embedding them in a local context consisting of a square image region, oriented and normalized for size by the key curve, which is placed at the center. Each keyed context patch contains a representation of all other segmented curves, key or not, that intersect it.

For object recognition, keyed context patches are extracted from the set of gallery images, and stored in a database together with the identity of the object that produced them, and the viewpoint they were taken from. The basic recognition procedure consists of four steps. First, keyed context patches are extracted from the probe image. In the second step, these keyed context patches are used to access the database and retrieve information about what objects could have produced them. Verifying a match between a patch from the image and a stored patch is done using a form of directional correlation. Each context patch from the image may match zero, one or more context patches from view models in the database, and each match generates a hypothesis about the identity and pose of an object that could have produced it. Patches that are consistent with the same view model in the same configuration form a group that accumulates evidence for that configuration and view model. Finally, in the fourth step, after all features have been processed, the hypothesis corresponding to the group with the highest evidence score is selected.

5 Experiments

5.1 The Data Set

We tested object recognition performance on images of 6 objects that were chosen to be "different" in that they were easily distinguishable from each other: a cup, a toy bear, sports car, a toy rabbit, a plane and a fighter plane (see Figure 1).

The gallery set consisted of 583 clean, black background images, taken at about 20 degrees apart over the viewing sphere. We had 106 images per object (53 images per object hemisphere), except the sports-car that had only 53 images (we covered only the top hemisphere since the bottom hemisphere was flat and

black, and thus uninteresting). These images were obtained automatically, using a combination of a robot-mounted camera, and a computer controlled turntable covered in black velvet.

Since translation, scaling and in-plane rotation are handled by all the image descriptors being tested, these images give us recognition over the full 6 orthographic degrees of freedom.

We used two different sets of probe images: one with images taken on a clean, black background (Figure 1), and another one with images taken on a heavily cluttered background (Figure 2). Each set contains 24 images per object hemisphere (24 images total for the sports-car and 48 images for each of the other objects), positioned unevenly in between training views. The black background pictures were taken using the same setup as for the training images. The cluttered background images were taken by placing the objects on a colorful poster and moving them around to make sure that the clutter features did not repeat in the images.

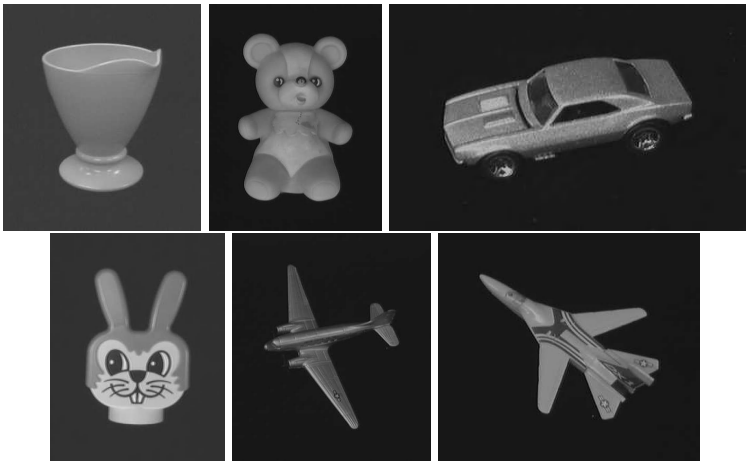


Fig. 1. Images of objects on cluttered background

5.2 Results and Discussion

We trained and tested each local image descriptor as described in the previous sections.

Lowe [2] states that in an object recognition system, the matching of SIFT features should be followed by a Hough transform for identifying clusters that agree on object pose, least-squares pose determination, and final verification. In fact, all three descriptors that we tested could benefit from these steps. A loose version of the Hough transform is used in the keyed context patch recognition system, but is not easily separable. The other steps are missing from all the methods we tested. We have chosen to skip the additional steps not only for the sake of simplicity, but also because of the structure of our training data. We get



Fig. 2. Images of objects on cluttered background

pose consistency implicitly, by matching an incoming image against a specific gallery image (and not a just collection of features belonging to different gallery images of the same object).

To be consistent with previously published results, we used the implementations provided by the authors for each method that we tested, including their suggested distance thresholds.

To test the SIFT method, we extracted SIFT features from all our gallery images. Then, for each probe image, we looked for the gallery image that matches it best, i.e. the gallery image that has the highest percentage of features that match the probe. The identity of the object in this gallery image was selected as the identity of the object in the probe image.

As expected, the recognition performance was much better on the black background images at 91%, than on the cluttered background images at 46%. Detailed error matrices for the two test sets can be seen in Tables 1 and 2.

Table 1. Error matrix for SIFT, clean background

class name	index	samples	0	1	2	3	4	5
cup	0	48	47	1	0	0	0	0
toy-bear	1	48	2	46	0	0	0	0
sports-car	2	24	9	2	12	0	1	0
toy-rabbit	3	48	2	0	0	46	0	0
plane	4	48	3	2	0	0	43	0
fighter	5	48	3	0	0	0	0	45

For PCA-SIFT we first needed to compute an eigenspace. We chose not to use the one provided by the authors, since the images that they used may not necessarily be representative for our domain. Instead, we randomly selected 100

Table 2. Error matrix for SIFT, cluttered background

class name	index	samples	0	1	2	3	4	5
cup	0	48	28	7	0	7	6	0
toy-bear	1	48	21	21	0	5	1	0
sports-car	2	24	12	3	4	2	3	0
toy-rabbit	3	48	5	5	0	37	1	0
plane	4	48	26	9	0	4	8	1
fighter	5	48	12	6	0	3	2	25

images out of our gallery, and used the image gradient vector of the patches extracted from these images to build our eigenspace. This amounted to 8607 patches.

Similarly to SIFT, we looked for the gallery image that has the highest percentage of its features matched by the probe image. Recognition performance was 97% for black background images, and 63% for cluttered images, much better than the performance obtained using the original SIFT features. This is consistent with results on image matching, where PCA-SIFT was found to perform better than SIFT [3]. Detailed error matrices for the two test sets can be seen in Tables 3 and 4.

Table 3. Error matrix for PCA-SIFT, clean background

class name	index	samples	0	1	2	3	4	5
cup	0	48	48	0	0	0	0	0
toy-bear	1	48	0	48	0	0	0	0
sports-car	2	24	1	0	20	0	3	0
toy-rabbit	3	48	0	0	0	48	0	0
plane	4	48	0	0	0	0	48	1
fighter	5	48	0	1	0	0	1	46

Table 4. Error matrix for PCA-SIFT, cluttered background

class name	index	samples	0	1	2	3	4	5
cup	0	48	39	2	0	0	3	4
toy-bear	1	48	21	23	3	0	1	0
sports-car	2	24	6	5	8	1	1	3
toy-rabbit	3	48	1	0	0	41	6	0
plane	4	48	13	4	0	0	31	0
fighter	5	48	15	8	0	0	1	24

The last method that we tested was recognition based on keyed context patches. We trained and tested the system as described in Section 4. We obtained recognition performances of 99% on the black background images and 72% on the cluttered images, significantly higher overall than SIFT and PCA-SIFT. This is not entirely surprising, as the features used in this method were

Table 5. Error matrix for keyed context patches, black background

class name	index	samples	0	1	2	3	4	5
cup	0	48	48	0	0	0	0	0
toy-bear	1	48	0	48	0	0	0	0
sports-car	2	24	0	0	23	0	1	0
toy-rabbit	3	48	0	0	0	47	1	0
plane	4	48	0	0	0	0	47	1
fighter	5	48	0	0	0	0	0	48

Table 6. Error matrix for keyed context patches, cluttered background

class name	index	samples	0	1	2	3	4	5
cup	0	48	26	0	1	0	16	5
toy-bear	1	48	7	10	0	1	28	2
sports-car	2	24	0	0	23	0	1	0
toy-rabbit	3	48	1	0	1	37	8	1
plane	4	48	0	0	0	0	48	0
fighter	5	48	0	0	0	0	2	46

specifically designed for object recognition in a setup that is very similar to ours. Detailed error matrices for the two test sets can be seen in Tables 5 and 6.

However, a closer examination of the error matrices shows that while the keyed context patches outperform PCA-SIFT overall, PCA-SIFT performs better on some objects (the cup, the toy-bear and the toy-rabbit). Depending on the feature type, some objects are recognized easier than others. SIFT and PCA-SIFT perform better on objects with curved contours (the cup and the toy-bear), while the keyed context patches do better on objects with straight contours (the plane, the fighter and the sports-car). This is due to the underlying structure of each feature type, as well as the keypoints around which these features were extracted.

Since the two features that perform best, PCA-SIFT and keyed context patches, are uncorrelated, it is very likely that combining the two features would lead to an even better recognition performance. This could be done either at object hypothesis level, by analyzing confidence scores of the two recognition methods, or at feature level. The structure of the keyed context patch recognition system is such that it would easily permit the incorporation of a new feature [13].

6 Conclusions

In this paper we analyzed the effect that the use of different local image descriptors has on object recognition. We chose two features that performed best in previous image matching experiments, SIFT and PCA-SIFT, and a feature that performed very well in object recognition experiments, keyed context patches.

Our data set consisted of images of six objects taken around the viewing sphere on clean and cluttered backgrounds. All three features performed well

given the complexity of the images, with clean background performances that were significantly higher than cluttered background ones.

PCA-SIFT performed better than SIFT, confirming previous image matching results. The best overall results were obtained by the keyed context patches used in a hierarchical object recognition system. However, PCA-SIFT outperformed the keyed context patches on some objects, showing that the two features are uncorrelated.

In the future we plan to investigate ways of combining the two features in order to obtain even better recognition performance.

Acknowledgement

Many thanks to Randal Nelson from the University of Rochester for useful discussions and for reviewing an early version of this paper.

References

1. Mikolajczik, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1615–1630
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
3. Ke, Y., Sukhtankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: Computer Vision and Pattern Recognition, Washington, D.C. (2004) 511–517
4. Selinger, A., Nelson, R.C.: A perceptual grouping hierarchy for appearance-based 3d object recognition. Computer Vision and Image Understanding **76** (1999) 83–92
5. Dorko, G., Schmid, C.: Selection of scale-invariant parts for object class recognition. In: 9th International Conference on Computer Vision, Nice, France (2003) 634–640
6. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Computer Vision and Pattern Recognition, Madison, Wisconsin (2002) 264–271
7. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation by image exploration. In: 8th European Conference on Computer Vision, Prague, Czech Republic (2004) 40–54
8. Leibe, B., Schiele, B.: Interleaved object categorization and segmentation. In: 14th British Machine Vision Conference, Norwich, UK (2003) 759–768
9. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991) 891–906
10. Koenderink, J., van Doorn, A.: Representation of local geometry in the visual system. Biological Cybernetics **55** (1987) 367–375
11. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets. In: 7th European Conference on Computer Vision, Copenhagen, Denmark (2002) 414–431
12. Van Gool, L., Moons, T., Ungureanu, D.: Affine / photometric invariants for planar intensity patterns. In: 4th European Conference on Computer Vision, Cambridge, UK (1996) 642–651
13. Nelson, R., Selinger, A.: A cubist approach to object recognition. In: 6th International Conference on Computer Vision, Bombay, India (1998) 614–621
14. Nelson, R.C.: Finding line segments by stick growing. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 519–523

Improving Spatiotemporal Inpainting with Layer Appearance Models

Thommen Korah and Christopher Rasmussen

Dept. Computer and Information Sciences
University of Delaware
Newark, DE 19716
{korah, cer}@cis.udel.edu

Abstract. The problem of removing blemishes in mosaics of building facades caused by foreground objects such as trees may be framed in terms of inpainting. Affected regions are first automatically segmented and then inpainted away using a combination of cues from unoccluded, temporally adjacent views of the same building patch, as well as surrounding unoccluded patches in the same frame. Discriminating the building layer from those containing foreground features is most directly accomplished through parallax due to camera motion over the sequence. However, the intricacy of tree silhouettes often complicates accurate motion-based segmentation, especially along their narrower branches. In this work we describe methods for automatically training appearance-based classifiers from a coarse motion-based segmentation to recognize foreground patches in static imagery and thereby improve the quality of the final mosaic. A local technique for photometric adjustment of inpainted patches which compensates for exposure variations between frames is also discussed.

1 Introduction

For a task such as vision-based architectural modeling, one subgoal is to obtain "clean" texture maps of the planar building faces with foreground objects removed. Image/video inpainting or completion [1, 2, 3, 4], a method for image restoration or object removal, suggests a way to remove larger foreground elements by interpolating building features spatially and/or temporally. Typically, the region to be filled is user-specified, but in previous work [5] we showed how problem areas could be automatically identified using motion cues. In that work, we observed that pure spatial inpainting is strictly necessary only where the background is never seen for the entire sequence, and that median filtering suffices when it is present in a majority of views. Our major contributions were (1) how to find the *holes* to be filled and (2) how to use building regions visible in a non-zero *minority* of views to constrain what should be painted there. By combining spatial information from pixels in a partially-completed mosaic with the temporal cues provided by images in the *timeline*, or sequence of images captured, sequences that present significant difficulties for existing background-subtraction techniques could be well-handled.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 718–730, 2006.
© Springer-Verlag Berlin Heidelberg 2006

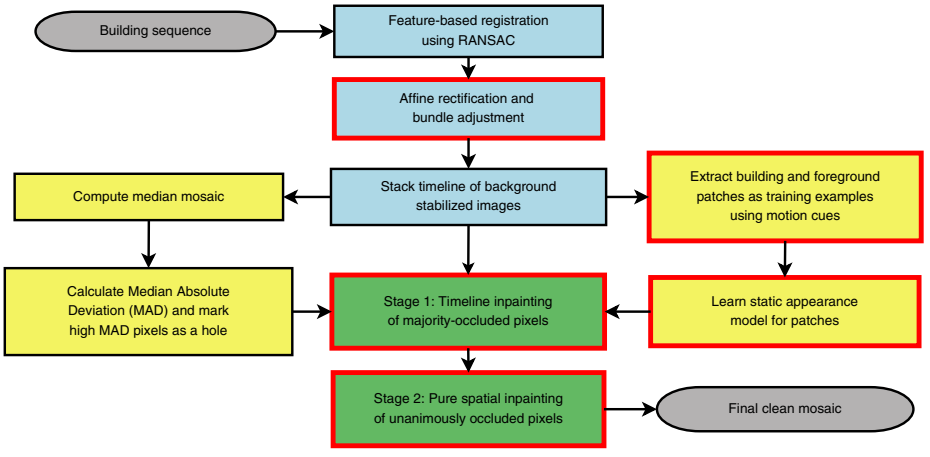


Fig. 1. System diagram for recovering clean texture maps. The thick red boxes indicate where the novel techniques discussed in this work fit into the overall framework.

This paper extends our previous work in several significant ways to achieve considerably improved results on difficult sets of images. First, we have found that motion cues and inpainting alone have shortcomings in eliminating certain foreground elements such as leaves and thin branches. An inability to recognize patches in the timeline containing such features tended to introduce subtle artifacts into the final mosaic, primarily because our method for measuring motion energy was not fine-grained enough to capture high-frequency occluders.

In [6] we enhanced the algorithm with a PCA-based measure of *appearance* similarity for patch selection in the spatiotemporal inpainting algorithm. By learning a local model of the building appearance from a set of exemplar patches, inpainted regions were virtually indistinguishable from the previous SSD-based method and filled in more than an order-of-magnitude faster–a key improvement when dealing with long sequences and panoramas. While the focus of that work was efficiency, appearance cues (vs. solely motion-based ones) suggested additional avenues for increasing the quality of foreground-background discrimination that we explore further in this paper. In the next sections we will briefly review our basic algorithm from [5], which we call *timeline inpainting*, describe our procedures for patch classification and photometric alignment, and present results for several different building sequences.

2 Timeline Inpainting

2.1 System Framework

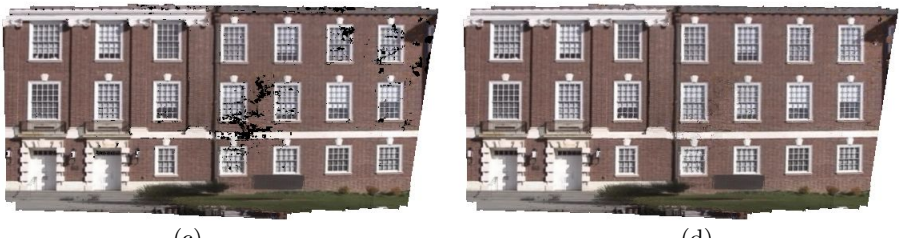
Figure 1 shows a system diagram of the timeline inpainting framework introduced in previous work [5]. Given a building sequence, it outputs a texture map of the facade with all foreground elements removed - as illustrated by Fig. 2



Every alternate frame from 9 to 17 in the 18 frame sequence



(a) (b)



(c) (d)

Fig. 2. Result from various stages of our algorithm for the Building A sequence (top row). The central windows are occluded in almost all frames making the case for inpainting. (a) Median mosaic. Foreground pixels are intermingled with the background when they are in the majority of frames. (b) MAD outliers that will be fed to the inpainting algorithm for filling. (c) Result after timeline inpainting (Stage 1). (d) Result after spatial inpainting (Stage 2).

containing the result from various stages of this algorithm. The boxes in the system diagram are shaded according to its function, and the thick red boxes indicate where the novel techniques introduced in this work fit in. The pipeline begins by computing the dominant planar motion (assumed to belong to the building facade) between successive pairs of images $\mathbf{I}_t, \mathbf{I}_{t+1}$ in a sequence of N frames. This sequence is expected to be captured from a robot platform while moving around the building perimeter (1-D scanning motion). A line detector is applied on the warped images to compute the rectifying homography. Finally, a full bundle adjustment is carried out to compute a single frame-to-mosaic homography for each frame. The result is a stack of background stabilized images $\mathbf{W}_t$.

Each location $\mathbf{p} = (x, y)$ in the mosaic reference frame has a set of pixels from the background stabilized images $\{\mathbf{W}_t(\mathbf{p})\}$ associated with it which we call its *timeline* $\mathcal{T}(\mathbf{p})$. Intuitively, since all pixels on the building facade exhibit the dominant motion, they should appear stationary in the mosaic whereas foreground objects such as trees and signs move due to parallax. Given that each $\mathcal{T}(\mathbf{p})$ contains an unknown mixture of background and foreground object

pixels, our goal is to correctly pick or estimate each background pixel $\mathbf{M}(\mathbf{p})$ where $|\mathcal{T}(\mathbf{p})| > 0$, forming a timeline mosaic $\mathbf{M}_{time}$. It is also important to identify those pixels where the background was never imaged (unanimously occluded), in which case regular image completion needs to be done based on contextual information from the rest of the mosaic. The result of the second stage would be the required texture map of the building mosaic $\mathbf{M}$. These two stages are shown in the green boxes forming the central component of our algorithm.

The boxes to the left of the main pipeline indicate that the median mosaic $\mathbf{M}_{med}$ is used as an initial robust estimator. The temporal median belongs to the background only if it was imaged in a majority ($> 50\%$) of frames. We use the median absolute deviation (MAD) [7] metric to threshold out outlier pixels. These are marked as holes to be inpainted over.

The key contribution of this paper is shown in the yellow boxes to the right of the central pipeline. Most exemplar-based methods [8, 9] use the SSD as the distance function $d(\cdot, \cdot)$ between two image patches. For large search regions (as typically occurs with panoramas or videos), this could be very computationally expensive. We therefore try to "learn" a more compact representation of building and foreground patches to measure building likelihood. Since appearance models can vary with location or season, motion cues are exploited to bootstrap the learning process by generating positive and negative examples of building patches. A pixel-wise Gaussian color model is also learnt to complement the coarser patch-based models. The techniques will be elucidated in greater detail in the following sections.

We first discuss some other improvements we've made to the overall framework including patch selection by better fusion of motion and appearance likelihoods, radiometric alignment of patches from distant frames, and incorporating domain knowledge to assist Stage 2 of spatial inpainting. We will then describe the appearance models used.

2.2 Patch Selection

Consider a patch $\Psi_{\hat{\mathbf{p}}}$ on the fill-front boundary of the timeline mosaic $\mathbf{M}_{time}$ that is the next to be inpainted. Pixels in its unfilled part $\Psi_{\hat{\mathbf{p}}} \cap \Omega$ (where Ω is the hole) will come from the corresponding part of one timeline patch $\Psi^*_{\hat{\mathbf{p}}} \cap \Omega$. The building likelihood of each timeline patch in $\Psi^t_{\hat{\mathbf{p}}}$ is jointly measured by the appearance and motion energy as

$$B(\Psi^t_{\hat{\mathbf{p}}}) = p_{app}(\Psi^t_{\hat{\mathbf{p}}})\bar{p}_{motion}(\Psi^t_{\hat{\mathbf{p}}}). \tag{1}$$

Pixels are then copied from $\Psi^*_{\hat{\mathbf{p}}}$ determined by $* = \text{argmax}_t\, B(\Psi^t_{\hat{\mathbf{p}}})$. The exact definition of the terms on the right side will be deferred until the next section, but suffice it to say that they represent the likelihood of a patch belonging to either the building or the foreground based on appearance and motion cues respectively. To prevent copying patches from timelines where the background was never imaged, we set appearance and motion thresholds T_{app} and T_{motion}.



Without correction



With correction

Fig. 3. Results of photometric correction during inpainting on Building C sequence

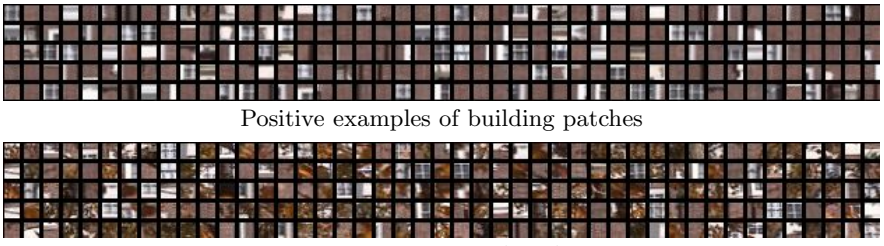
T_{app} varies with the appearance model used and is determined by cross-validation experiments on the training set. T_{motion} is set to 0.8 implying that patches with more than 80% of pixels above the motion threshold τ_{motion} will not to be included in the computation of (1). The learnt models will be shown to be discriminative with well separated decision boundaries, thus easing the task of setting these thresholds.

Since (1) operates on patches, it does not guarantee against blemishes in the mosaic that occur when tiny fragments of foreground pixels are copied over. Thus a per-pixel decision is also made before copying patches from the timeline - once again based on appearance and motion. The Gaussian color model is used to threshold out bad pixels based on the RGB values, while pixels with motion energy below τ_{motion} are also not copied to $\mathbf{M}_{time}$. This combined framework allows us to find the right balance between spatial and temporal cues at either fine-grained or coarse-grained resolutions.

2.3 Photometric Alignment

For a sequence with significant variations in intensity, either due to lighting changes in the scene or automatic camera controls, the seams between overlapping patches may be pronounced. Graph-cut and gradient-domain algorithms [10, 11] have been used to minimize these effects in mosaicing and texture synthesis. Since most of the photometric variations in our sequence arise due to varying camera parameters, we experimented with exactly recovering the camera response curves to compensate for these radiometric misalignments [12]. However this proved very sensitive to misalignments and foreground objects. Noting that an affine transformation across each RGB channel is able to fully account for contrast and brightness changes [13, 14], we simply use a multiplicative term λ_k that represents the contrast change across the patch. When pixels from the best chosen patch $\Psi^*_{\hat{\mathbf{p}}}$ are to be copied into the timeline mosaic $\mathbf{M}_{time}$, λ_k is estimated by least squares minimization over the overlapping pixels. This correction is applied before the missing pixels are filled in.

Figure 3 focuses on the result of inpainting with and without photometric alignment on a section of the Building C sequence. When compared to [5], the mosaics generated using this technique appear much more consistent and visually pleasing. Even though we only model contrast changes, it is sufficient when applied to a small patch and is able to propagate to a much larger scale.



Positive examples of building patches

Negative examples from RANSAC outliers

Fig. 4. Instances of 11×11 patches from the Building A sequence used for training the classifier

2.4 Heuristics for Spatial Inpainting in Stage 2

Mosaic pixels that were never imaged in the timeline are detected in Stage 1 and marked as a hole - to be completed in Stage 2 by a general spatial inpainting algorithm. Given that most of the background has been recovered in Stage 1, only a small fraction of pixels require conventional inpainting. We use the algorithm of [2] with a few heuristics derived from domain knowledge. Firstly, we search within the warped sequence $\mathbf{W}$ rather than the result of Stage 1 to improve the likelihood of finding a good match. Secondly, since building facades exhibit grid-like patterns, we limit the SSD search to lie within a horizontal and vertical grid around the target patch. This serves to speed up the search through the sequence and reduce the chances of picking a wrong patch to copy into the hole.

3 Measuring Building Likelihood

3.1 Appearance Modeling

The temporal information available from the timeline has already limited the candidate pixels that can be copied to a small set of aligned image patches in the stack $\mathbf{W}$. It was suggested in our previous work [6] that the appearance matching problem could be reformulated as classifying patches in a timeline as building or foreground. Compared to the exhaustive search over Φ using SSD, this could potentially be much more efficient. We now explain our design of a classifier that can disambiguate between building and foreground patches.

Training Set. Most classification and semantic recognition algorithms [15, 16, 17] rely on supervised training with a user manually marking out positive and negative examples. Learning from a universal set of examples does not seem appropriate to our task as the nature of building and non-building patches could vary with location or even seasons. We instead use motion cues to generate training examples specific to the particular sequence - thus bootstrapping the learning for static patch classification. Positive examples of building patches are selected from the MAD inliers Φ by extracting $n \times n$ patches spaced by m pixels on a regular grid (patches overlap when $m < n$). The patch size for learning

and inpainting n was typically 9 or 11, and the spacing varied from $m = 3$ to $m = n$. In addition, we detect Harris corners on the MAD inliers and add them to the positive examples, with the hope of capturing the most important building features also.

The negative examples belonging to trees, grass and so on are harder to extract. Rather than manually demarcate regions, we use the RANSAC outliers from the image registration and mosaicing stage as possible examples of non-building patches. This assumption is reasonable if the dominant motion from frame-to-frame is that of the building. Even if a small set of RANSAC outliers fall on the building - as it does in our case, the classification algorithm should be robust to these incorrect labellings. Figure 4 shows a few examples of 11×11 patches that were used for training. In a similar vein, if the MAD inliers alone are unable to capture the salient building features due to excessive occlusion, we add the RANSAC inliers to the training set as positive examples - albeit at the risk of introducing more errors into the training set.

Modeling Pixel Color. An obvious first step is to experiment with the discriminative ability of pixel color in separating the background from the foreground. If indeed a statistical model can be learnt from the RGB values, it would have the advantage of being spatially and temporally fine-grained. Tracking and segmentation algorithms [18] have used Gaussian Mixture Models to model color distributions. We take the set of RGB values from the training set patches to model the foreground (F) and background (B) as two Gaussian distributions described by means μ_f, μ_b and covariances Σ_f, Σ_b. The computation of the foreground likelihood for pixel y_i can now be framed as

$$p(y_i|F) = \frac{1}{Z_f} \exp\{-\frac{1}{2}(y_i - \mu_f)^T \Sigma_f^{-1}(y_i - \mu_f)\} + \epsilon \tag{2}$$

where $Z_f = (2 * \pi)^{3/2} \Sigma_f^{1/2}$. The background likelihood is also computed in a similar manner. Given the RGB values of a pixel, the probability of it belonging to the building background is computed as $P(y_i = B) = \frac{p(y_i|B)}{p(y_i|B)+p(y_i|F)}$. Despite the lack of context, it provides a strong cue which is first used to refine any obvious errors in the thresholded MAD inliers, especially in the homogeneous regions.

Visual Features. Given several labeled image patches belonging to the building and foreground, we wish to build an empirical model of building and non-building features for the particular sequence. Our previous technique [6] used only color and intensity of a patch to measure its appearance similarity to other building patches. The $n \times n$ size patches are reduced to a low-dimensional space using Principal Component Analysis (PCA) which maximizes the scatter of all the training samples. However, PCA has various limitations (Gaussian distribution, orthogonal linear combinations etc.), which could make it less applicable despite its simplicity. We observe that building patches are likely to contain prominent linear structures with a majority of them oriented horizontally or vertically. In

addition, there could be texture variation in the brick or leaf that color alone might not capture - even at the patch level.

Filter banks have been widely used for texture recognition [19, 15] as well as object/scene categorization [16, 17]. We employ the common Base Filter Set (BFS) used by [15] with 34 filters (8 orientations at 2 scales for 2 oriented filters, plus 2 isotropic). Color patches are converted to grayscale before the convolution is performed. In addition, we append the mean R, G, and B color values across each channel to obtain a 37-dimensional feature vector. For better classification results, the input attributes were mapped to be within [-1,1] and the responses normalized to prevent some attributes from dominating over others.

Appearance-Based Building Likelihood. The classifier used should be robust to outliers in the training set and also generalize well over the training examples without being too sensitive to the number of patches used for training. We explore the two methods described below.

Nearest neighbor classifier: Given a test patch $\Psi_{\mathbf{y}}$, we can classify it as belonging to class $\hat{\nu}$ that has the maximum posterior probability. The patch $\Psi_{\mathbf{y}}$ is first projected into the k-dimensional feature space. Let $(< \mathbf{x}_1, V(\mathbf{x}_1) > \ldots < \mathbf{x}_N, V(\mathbf{x}_N) >)$ be the N nearest neighbors and their associated labels from the training examples. Then we return a distance weighted likelihood

$$p_{app}(\Psi_{\mathbf{y}}) = \frac{\sum_{i=1}^{N} w_i(\Psi_{\mathbf{y}}, \mathbf{x}_i)\delta(Building, V(\mathbf{x}_i))}{\sum_{i=1}^{N} w_i(\Psi_{\mathbf{y}}, \mathbf{x}_i)}$$

where $w(\cdot, \cdot)$ is the reciprocal of the Euclidean distance between the two patches and $\delta(a, b) = 1$ if $a = b$ and 0 otherwise.

Support Vector Machine: A Support Vector Machine (SVM) [20] classifier was trained on the set of patches to learn a model for building and foreground appearance. The 37-dimensional response of the filter bank was fed as the feature vector to SVM. The data was normalized for more accurate results. We used SVM with an RBF kernel to map the training vectors to a higher dimensional space - the best γ being chosen by cross-validation experiments. The freely available SVM-Light package was used for training and classification.

3.2 Motion Cues

The appearance likelihood method is coarse-grained since it operates on patches using the whole neighborhood of pixels for support. This can result in patches with a very small fraction of foreground pixels to be classified as building - a common problem around the edges of leaves or branches. The color models, in spite of working at pixel resolution, are not reliable enough as they completely disregard context. To complement this, we employ motion cues that are spatially fine-grained but temporally coarse.

The intersection of a pair of successive, thresholded difference images was suggested in [21] as a method for identifying foreground pixels. By converting

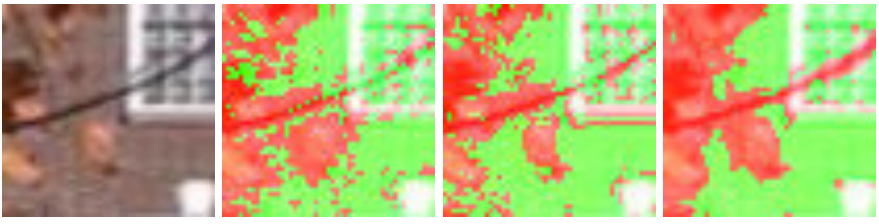


Fig. 5. Comparison of pixel classification on zoomed in section of a frame based on PCA and filter banks(FB) with k-nearest neighbor (KNN,$k = 10$) and SVM. Green shade indicates building while red regions indicate pixels classified as foreground.

the warped images to grayscale and scaling their intensity values to $[0, 1]$ to get $\{\mathbf{W}'_t\}$, we can adapt this approach to define a motion energy or *foreground image* at time t. Also, $\bar{p}_{motion}(\Psi^t_{\mathbf{p}})$ for a patch is the fraction of pixels in $\Psi^t_{\mathbf{p}}$ with background likelihood above a minimum threshold τ_{motion}.

4 Results

4.1 Classification

The quality of the static classifier in distinguishing between building and non-building pixels based on the local statistics in a patch would have a direct impact on the spatiotemporal inpainting approach of [6], not only in preventing small foreground pixels from bleeding into the final mosaic, but also to pull in building patches that the motion cues would otherwise discard due to its temporal coarseness.

Figure 5 zooms in on a small section of a frame and the corresponding result of classification based on PCA, filter banks and SVM. The classification is done independently on patches centered around every pixel. The training for Building A was done with a total of 2908 patches, out of which 2169 were positive examples from MAD and RANSAC inliers. The decision to classify a pixel as foreground or background with k-nearest neighbor was made by setting a threshold of 0.8 on p_{app}. These numbers were arrived at after running the leave-one-out test with a series of parameters and the model with the least percentage of error was chosen.

A few key factors can be noted. All methods seem to detect most of the foreground or tree pixels. PCA essentially works on the RGB color values and might not be able to pick up some of the high frequency variations that the filter bank can. Texture-based classification is thus able to do marginally better on some of the tiny leaves or thin branches that occlude the building. SVM seems the cleanest among all three methods for segmentation. It can generalize well over the training set with several examples and the classification is done in fixed time. In contrast, nearest neighbor approaches become very inefficient as we add more patches.

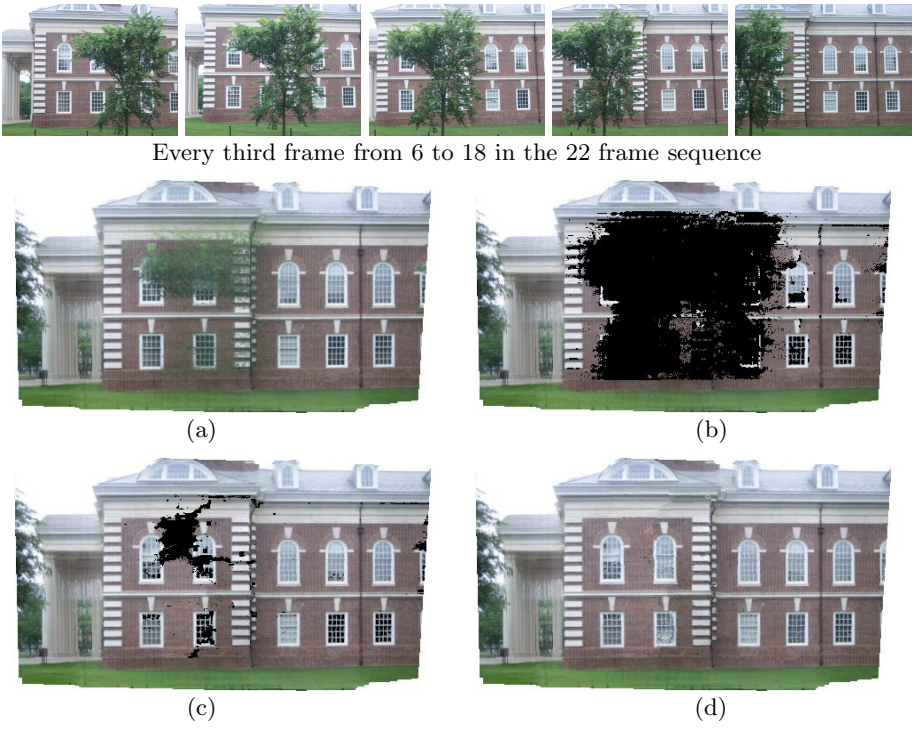


Fig. 6. Building B sequence. (a) Median mosaic. (b) MAD outliers refined by color model that will be fed to the inpainting algorithm for filling. (c) Result after timeline inpainting (Stage 1). (d) Result after spatial inpainting (Stage 2).

For quantitative results, we ran a leave-one-out test on the 2908 patches for Building A. The best result using PCA was 15.1% error with 10-nearest neighbors and a distance threshold of 0.85. The lowest error using filter banks was 11.2% under the same settings. The best accuracy by far was obtained with SVM which misclassified only 3.6% of the training examples. On closer inspection, it was observed that most of these errors boiled down to incorrect labels in the training examples itself.

4.2 Spatiotemporal Inpainting

Each of the appearance models as well as photometric alignment was integrated into the inpainting framework of [6] for comparison. For lack of space, we only show results using SVM learnt on the filter responses. Similar to Fig. 2, figures 6 and 7 illustrate the output at various stages for two other building sequences.

Compared to the SSD metric used in [5], the three appearance models and classification results demonstrate the feasibility of using them to make a hard decision of background or foreground. Contrastingly, this is not possible using SSD and one can only rely on the motion/parallax information that requires a



Every alternate frame from 6 to 14 in an 18 frame sequence



(a) (b)

Fig. 7. Building C sequence. (a) Median mosaic. (b) Result after spatial inpainting (Stage 2).

background pixel to be imaged in three consecutive frames. The classification approaches are clearly much more efficient (SSD required 778 seconds) since the slowest of the 3 methods (FB+KNN) required only 115.3 seconds with definite improvements in the quality of the mosaic.

5 Conclusion

We have described a method of training appearance-based classifiers from a coarse RANSAC-based segmentation to recognize static imagery. The primary motivation behind this work was to identify high frequency features that motion-based approaches alone could not consistently capture. However, we use motion to bootstrap the learning and generalize over the training examples. We have applied the results of our models to both static scene segmentation as well as inpainting to recover texture maps of occluded building facades. Various types of visual features - both intensity-based and texture-based - were proposed in order to learn a low-dimensional representation of image patches to aid in accurate classification of image patches.

Inpainting a building facade is very hard due to the high amount of structure accentuating slight errors or misalignments. Most existing algorithms would not be able to recover the whole facade behind the tree - and we have shown promising results using appearance and motion cues. From the results at various stages of the process, it is obvious that Stage 2 spatial inpainting is the weakest. This is because most inpainting algorithms rely on greedy matching of patches. A single bad choice can propagate errors without any backtracking procedure. For example, the second window from the left in the upper story of Building B reveals an error in the filling in. Our higher level semantic knowledge suggests that a small white patch should be copied on the upper left edge of the window,

but this kind of reasoning is not built into the inpainting. Our future work involves incorporating such abilities into the hole-filling. At a lower level, we wish to experiment with gradient-domain methods [11] for blending patches and even building entities like whole windows or doors. If the higher-level module reasons from a single image that a window is completely occluded by a tree, we would like to first determine where the window ought to appear and seamlessly copy a whole window from a different part of the image.

References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH. (2000) 417–424
2. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Processing **13** (2004)
3. Jia, J., Wu, T., Tai, Y., Tang, C.: Video repairing: Inference of foreground and background under severe occlusion. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2004)
4. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2004)
5. Rasmussen, C., Korah, T.: Spatiotemporal inpainting for recovering texture maps of partially occluded building facades. In: IEEE Int. Conf. on Image Processing. (2005)
6. Korah, T., Rasmussen, C.: Pca-based recognition for efficient inpainting. In: Proc. Asian Conf. Computer Vision. (2006)
7. Tommasini, T., Fusiello, A., Trucco, E., Roberto, V.: Making good features to track better. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (1998) 178–183
8. Efros, A., Freeman, W.: Image quilting for texture synthesis and transfer. In: SIGGRAPH. (2001)
9. Bornard, R., Lecan, E., Laborelli, L., Chenot, J.H.: Missing data correction in still images and image sequences. In: ACM Multimedia. (2002)
10. Szeliski, R.: Video mosaics for virtual environments. IEEE Computer Graphics and Applications **16** (1996) 22–30
11. Prez, P., Gangnet, M., Blake, A.: Poisson image editing. In: ACM Transactions on Graphics (SIGGRAPH'03). (2003) 313–318
12. Kim, S.J., Pollefeys, M.: Radiometric alignment of image sequences. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2004) 645–651
13. Capel, D., Zisserman, A.: Computer vision applied to super resolution. IEEE Signal Processing Magazine **20** (2003) 75–86
14. Jin, H., Favaro, P., Soatto, S.: Real-time feature tracking and outlier rejection with changes in illumination. In: Proc. Int. Conf. Computer Vision. (2001) 684–689
15. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis **62** (2005) 61–81
16. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 2. (2005)
17. Lu, L., Toyama, K., Hager, G.D.: A two level approach for scene recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2005) 688–695

18. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 747–757
19. Leung, T.K., Malik, J.: Recognizing surfaces using three-dimensional textons. In: ICCV. (1999) 1010–1017
20. Joachims, T.: Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods: Support Vector Learning. MIT Press (1999)
21. Toyama, K., Krumm, J., Brumitt, B., Meyers:, B.: Principles and practice of background maintenance. In: Proc. Int. Conf. Computer Vision. (1999)

Edge Detection Using a Complex-Valued Directional Vector Representation

Sung Bae Kim[1] and Rae-Hong Park[1,2]

[1] Department of Electronic Engineering, Sogang University
C.P.O. Box 1142, Seoul 100-611, Korea
{sbkim, rhpark}@sogang.ac.kr
[2] Interdisciplinary Program of Integrated Biotechnology, Sogang University

Abstract. Edge detection is a fundamental step in computer vision and image analysis. A new method using the modified Laplacian and position mask is proposed for edge detection of both gray-evel and color images. A directional field is computed by the proposed masks and edges are detected by additional simple mask operations. Edge detection results by the proposed masks are shown in comparison with those by Sobel, Laplacian of Gaussian (LoG), and Canny operators.

Keywords: Edge Detection, Modified Laplacian, Directional Field, Position Mask.

1 Introduction

Edges characterize boundaries of objects and are therefore of fundamental importance in computer vision and image analysis. Generally edges correspond to abrupt discontinuities in physical quantities such as gray-evel, color, texture, motion, and so on [1][2]. Most edge detection methods compute first-order derivatives followed by some thresholding. There are many small-sized edge masks, such as Prewitt, Sobel, Frei-Chen, and Roberts masks, which simply approximate the first derivative of the single channel (gray-level) image [2]. Sometimes detected edges using first-order derivatives are not satisfactory due to the tradeoff between thickness of detected edges and the detection rate, where the detection rate is defined by the ratio of the number of correctly detected edges to that of ground truth edges. These masks are simpler than the Laplacian of Gaussian (LoG) mask and thus, have been commonly used for real-time implementations. Finding zero-crossings of the LoG-filtered image is an example of edge detection using the second-order derivatives [3][4][5]. By LoG mask operations, thin edge lines that compose of closed contours are detected and relatively accurate edge positions can be found, which is dependent on the scale value of a Gaussian smoothing filter. However, some of detected edges are not boundaries of real objects (sometimes closed contours appear in the background due to the small variation of intensity or color) and thus a non-edge rejection process must be followed. Canny presented an optimal edge detection method that consists of a series of steps: Gaussian filtering, gradient operation, non-maxima suppression, and hysteresis thresholding [6].

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 731 – 740, 2006.
© Springer-Verlag Berlin Heidelberg 2006

A desirable edge map of an image is to be detected as one-pixel-thick edge lines having a high detection rate and detection accuracy, where the detection accuracy is represented in terms of the similarity between detected edge positions and the ground truth edge positions. LoG edge detection can produce one-pixel-thick edge lines with a relatively high detection accuracy compared with other edge detection methods. However, it cannot give directional information of edges. The directional information of edges in a gray-level image can be defined as the difference of intensity as in the edge detection methods based on the first-order derivative. However, the directional information cannot be defined in LoG edge detection that is based on a two-dimensional (2-D) isotropic approximation of the second-order derivative of a Gaussian-smoothed image.

In this paper a new approach to edge detection is presented. A modified form of the Laplacian and a complex-valued position mask are proposed and their application to edge detection of gray-level and color images is presented. The position mask is a small odd-sized square mask (e.g., 3×3, 5×5, or 7×7), which is suitable for machine vision and inspection applications where fast implementation is desirable. Using both masks, a directional field is computed and then edges are detected by simple additional mask operations, which is similar to finding zero-crossings in LoG edge detection. This complex-valued directional field has the direction information that is related to the directional absolute intensity differences, which are defined between the center pixel of the mask and its neighboring pixels. Using these properties, the proposed method detects edges as one-pixel-thick lines while non-edge pixels are rejected by thresholding of the magnitude.

The rest of the paper is structured as follows. Section 2 presents the proposed edge detection method and Section 3 shows experimental results and discussions. Finally, Section 4 gives conclusions of the paper.

2 Proposed Edge Detection Using the Modified Laplacian and Position Mask

In this section, a new approach to edge detection is presented. The Laplacian mask is a non-directional one, in which four directional approximations of the second-order derivative are combined. Also it can be represented as eight directional gradients. These characteristics are used to propose the modified Laplacian mask.

The overall procedure of the proposed edge detection is similar to finding zero-crossings in LoG edge detection. Fig. 1 shows the overall block diagram of the proposed edge detection algorithm. Each block in Fig. 1 is presented in the following subsections.

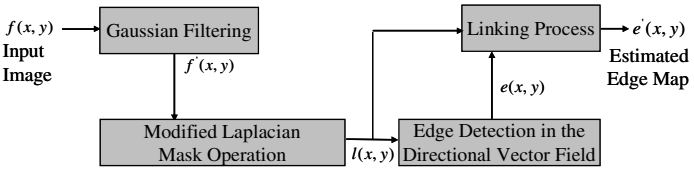


Fig. 1. Overall block diagram of the proposed edge detection method

2.1 Gaussian Filtering

Since the second-order derivative of a signal is more sensitive to noise or small variations than the first-order derivative, edge detection based on the second-order derivative accompanies a smoothing filter to effectively suppress noise components and small variations. A uniform filter or a Gaussian filter has been commonly used as a smoothing filter to reduce noise, where amount of noise reduction depends on the size of a filter. The Gaussian smoothing filter is employed in the proposed edge detection algorithm, which is defined as

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where σ denotes the scale parameter of a Gaussian filter. In a 2-D digital image, a Gaussian filter can be defined as a digital mask within the rectangular or circle region whose size is related to the value of σ, in which filtering can be easily realized by convolution with a Gaussian mask. The mask size of a smoothing filter should be as small as possible to decrease the image blurring.

2.2 Modified Laplacian and Position Mask

The conventional Laplacian mask for a 2-D digital image is defined as a linear combination of four directional approximations of the second-order derivative. The one-dimensional (1-D) approximation of the second-order derivative is given by

$$L(x) = \frac{\partial G(x)}{\partial x} \approx G(x+1) - G(x) = 2f(x) - (f(x-1) + f(x+1)) \tag{2}$$

where $f(x)$ represents an arbitrary 1-D signal and $G(x)$ is a gradient of $f(x)$.

Let $N_M(x, y)$ be a set consisting of neighboring pixels of the center pixel (x, y), defined as

$$N_M(x, y) = \left\{ (x+m, y+n) \middle| -\frac{M-1}{2} \le m, n \le \frac{M-1}{2}, m \ne 0, n \ne 0 \right\} \tag{3}$$

where the square mask size M is assumed to be odd. Fig. 2(a) shows the 3×3 Laplacian mask which is regarded as a combination of four directional second-order derivatives. Equivalently, it can be regarded as a combination of eight differences $d_{x,y}(m,n)$ defined between the intensity at the center pixel (x, y) and that at a neighboring pixel at $(x+m, y+n)$, which is shown in Fig. 2(b) and expressed as

$$L(x, y) = \sum_m \sum_n d_{x,y}(m,n) = \sum_m \sum_n (f(x, y) - f(x+m, y+n)) \ . \tag{4}$$

The differences have their own directions, similar to directional gradients, however the direction information in the gradient is not directly applicable to the Laplacian, which is isotropic with all the weights at neighboring pixels equal to -1. Note that the direction of the gradient is defined as a difference along the specified direction and therefore multiple directional masks are commonly defined to detect edges along different directions. As mentioned earlier, the Laplacian mask in Fig. 2(a) has no directional information in itself. However, from equation (4), each of differences can

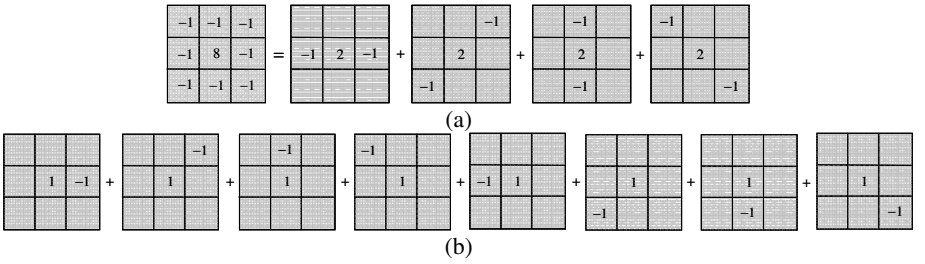


Fig. 2. Two interpretations of the 3×3 Laplacian mask. (a) Combination of 4-directional second-order derivatives, (b) Combination of 8-directional first-order derivatives.

be given its own directionality. If each position of neighboring pixels is represented as a vector in the 2-D mask and the difference (positive or negative) is replaced by the absolute difference, it is possible to define the direction at the center pixel (x, y) as the one showing a dominant intensity variation between the center pixel and neighboring pixels. Let $p_{m,n}$ be a normalized directional representation, in complex form, of the neighboring pixel (m,n) in a 2-D mask [2], which is represented as

$$p_{m,n} = \frac{1}{|m + jn|}(m + jn), \qquad -\frac{M-1}{2} \le m, n \le \frac{M-1}{2}, m \neq 0, n \neq 0 \tag{5}$$

where $j = (-1)^{1/2}$. Figs. 3(a) and 3(b) show examples of the 3×3 and 5×5 complex-valued position mask, respectively. The normalized complex-value $p_{m,n}$ represents the directional information in complex-valued vector form, which is dependent on the relative position of the neighboring pixel (m,n). Using the modified Laplacian, the complex-valued directional field $l(x,y)$ can be computed as a combination of eight complex-values position mask weighted by the directional absolute differences

$$l(x, y) = \sum_m \sum_n \left| d_{x,y}(m,n) \right| p_{m,n} \cdot \tag{6}$$

Since $l(x,y)$ is a linear combination of eight directional absolute differences weighted by normalized complex-valued coefficients in the position mask, the direction specified by an argument of $l(x,y)$ can represent the direction that is related to the variation of intensity. Also, this procedure can be easily extended to a larger odd-sized window, as shown in Fig. 3(b) as an example of a 5×5 mask.

In color images, e.g., in RGB color space, $\mathbf{C}_{x,y} = [R_{x,y} \quad G_{x,y} \quad B_{x,y}]^t$ represents the 3×1 color vector [7][8][9], the modified Laplacian can be easily generalized to color images as

$$l(x, y) = \sum_m \sum_n d^c_{x,y}(m,n)\, p_{m,n} \tag{7}$$

where $d^c_{x,y}(m,n) = \sqrt{(\mathbf{C}_{m,n} - \mathbf{C}_{x,y})^t (\mathbf{C}_{m,n} - \mathbf{C}_{x,y})}$ denotes the absolute difference in RGB color space.

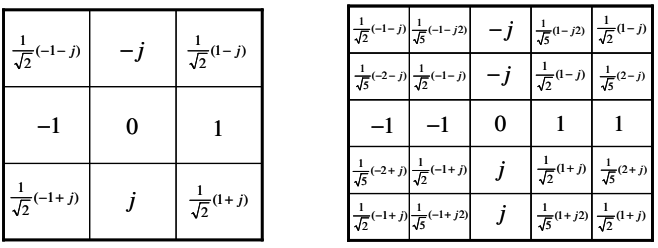


Fig. 3. Position mask. (a) 3×3 $\mathbf{p}_3$ mask, (b) 5×5 $\mathbf{p}_5$ mask

The complex-valued directional field $l(x, y)$ computed using the modified Laplacian and the complex-valued position mask has both the magnitude and directional information at each pixel. The magnitude of $l(x, y)$ is therefore similar to that of the Laplacian. The direction of $l(x, y)$ is similar to the gradient direction in a flat region. But if there is an estimated edge line inside the mask with $(x+m, y+n)$ on the estimated edge line, the absolute differences multiplied by $p_{m,n}$ are larger than the other absolute differences. As a result, the argument of the directional field $l(x, y)$ is almost or exactly perpendicular to the estimated edge line, which is similar to the gradient-based edge detection cases. If the center pixel is on an edge line, the two angles of $l(x, y)$ computed at two such neighboring points (opposite-directed pair of neighboring pixels) are opposite to each other, where the angle of the complex number is defined by the arctangent of the ratio of the imaginary part to the real part. The complex-values $l(x, y)$ computed at opposite-directed neighboring pixels have large absolute differences directed to the edge pixel located at the center of the mask and the directions represented as the complex-valued coefficients of the position mask. Then, magnitude of $l(x, y)$ at that position is close to zero because $l(x, y)$ at such two neighboring points have the similar large magnitudes with different sign. Finding the opposite-directed pair of neighboring pixels in the proposed edge detection method is similar to finding "zero-crossing" in LoG edge detection, implementation of which is described in subsection 2.3. Fig. 4 shows an example of a synthetic gray-level image, whose intensities are 40 and 180 inside and outside the

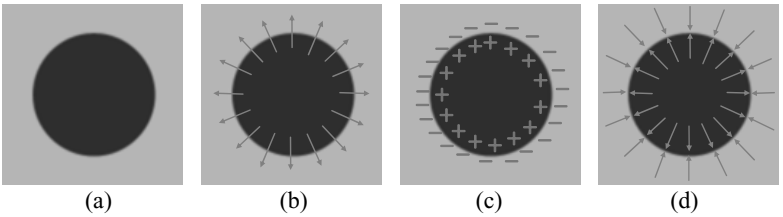


Fig. 4. Edge detection results (128×128, Synthetic). (a) Synthetic image, results of (b) Gradient mask, (c) Laplacian mask, (d) The proposed masks.

circle, respectively, and edge detection concepts by Sobel gradient masks, Laplacian mask, and the proposed edge detection masks.

2.3 Edge Detection in the Directional Field

In Section 2.2, the directional field is obtained by the modified Laplacian and position mask. In the proposed edge detection method, edges can be detected as pixels that have an opposite-directed pair of neighboring pixels facing each other with a relatively high magnitude of the directional field. For a pixel at (x, y), there are four types of opposite-directed pairs of neighboring pixels facing each other: horizontal, diagonal, vertical, and anti-diagonal pairs, which is shown in Fig. 5.

Feature matching can be used for finding the four cases in Fig. 5. Since the complex-valued directional field can be regarded as a 2×1 vector field, feature matching can be computed by an inner product with four possible pairs of complex-valued feature vectors. A prototype of the complex-valued feature in Fig. 5 can be written as

$$c_a(m,n) = \begin{cases} -\dfrac{m}{\sqrt{m^2+n^2}} - j\dfrac{n}{\sqrt{m^2+n^2}}, & \text{if} \quad (m,n) = (p,q) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where α is h, d, v, or ad, which signify horizontal, diagonal, vertical, and anti-diagonal, respectively, and $(p,q) = (-1,0)$ or $(1,0)$ when $\alpha = h$, $(p,q) = (-1,-1)$ or $(1,1)$ when $\alpha = d$, $(p,q) = (0,-1)$ or $(0,1)$ when $\alpha = v$, $(p,q) = (1,-1)$ or $(-1,1)$ when $\alpha = ad$, with $(0,0)$ corresponding to the center of the mask. Then, feature matching can be defined as an inner product in complex form

$$l(x+m, y+n)c_\alpha^*(x+m, y+n) \tag{9}$$

where c^* denotes complex conjugate of c Since $c_\alpha(m,n)$ is a pair of complex values, each pair of the four cases has two complex-values to be used to calculate the inner product. A pixel is declared as an edge if it satisfies the conditions that real parts of the inner product of the two complex-values must be positive (the two $l(x, y)$ s at the opposite-directed neighboring pixels must face each other. In other words, the directional field in a mask should be shaped like the one of the four cases in Fig. 5) and greater than the threshold, which can be expressed as

$$\forall((m,n)\big|(m,n) = (p,q)), \quad \mathrm{Re}[l(x+m, y+n)c_\alpha^*(x+m, y+n)] > \text{threshold} \tag{10}$$

with $\mathrm{Re}[q]$ representing a real part of a complex value q. Normally, edges in real images are not detected ideally, therefore a distance between the pixels of the opposite-directed pair of neighboring pixels can be one, two (horizontally or vertically), $\sqrt{2}$, or $2\sqrt{2}$ (diagonally or anti-diagonally) in a 3×3 mask. If it is equal to one or $\sqrt{2}$, edges can be detected as two-pixel-thick lines because the edge condition can be satisfied at two adjacent pixels that are aligned perpendicular to the edge line. For one-pixel-thick edge detection, detected edge pixels should be compared with the

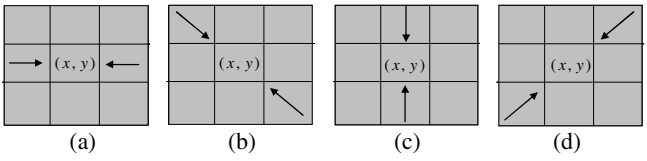


Fig. 5. Four types of the opposite-directed neighboring field. (a) Horizontal, (b) Diagonal, (c) Vertical, (d) Anti-diagonal.

previously-scanned opposite-directed pair of neighboring pixels of the best matched feature to find the pixel at which a larger change occurs. The pixel with a smaller magnitude of $l(x, y)$ should be detected as an edge for more accurate edge position, with the other pixels changed to as non-edge pixels.

2.4 Linking Process

Some missing edge pixels, whose neighboring pixels are opposite-directed but not detected as edges by thresholding, should be connected to previously-detected edges. To link disconnected edges, the angle at an edge pixel, i.e., the angle of the estimated edge line, must be computed using $l(x, y)$ at neighboring pixels. To compute the angle of the edge line at a disconnected edge pixel, a complex-value $t_l(x, y)$ is computed as

$$t_l(x, y) = \sum_m \sum_n s[l(m, n)]\, l(m, n) \text{ ,. } s[l(x, y)] = \begin{cases} 1 & \text{if } \operatorname{Re}[l(x, y)] > 0 \\ 0 & \text{if } l(x, y) = 0 \\ -1 & \text{otherwise} \end{cases} . \tag{11}$$

Since there are two directions perpendicular to the direction of the estimated edge lines, $s[l(x, y)]$ is defined to select one of two opposite angles. Then, the angle of the edge line is estimated as $\arg(t_l(x, y)) + 90^\circ$, where $\arg(t_l)$ represents an argument (or angle) of a complex value t_l. Then, the non-edge pixel, which is connected to the previously-detected edge pixel, can be changed to an edge pixel, if the condition in equation (10) is satisfied with a lower threshold. To find the next edge points and link them to the other edge points, above-mentioned procedure is repeated.

3 Experimental Results and Discussions

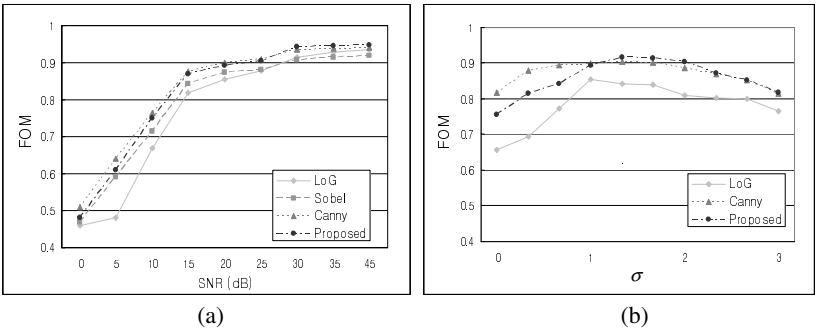
In order to compare the performance of edge detection, edge detection functions of LoG, Sobel, and Canny in MATLAB 7.01 are used, all of which are for gray-evel images. Threshold values, parameters in edge detection, are selected in such a way that the total number of edge pixels detected by each method is similar to each other for reasonable comparison.

A quantitative performance of the figure of merit (FOM) [10] is also used for performance evaluation of each detection method, which is defined as

$$FOM = \frac{1}{\max(I_D, I_I)} \sum_{i=1}^{I_D} \frac{1}{1+\beta(d_i)^2} \tag{12}$$

where I_D and I_I represent the number of estimated edges and the number of ground truth edges, respectively, β denotes a positive calibration constant (generally, equal to $1/9$), and d_i signifies the distance from the ground truth edges to the i^{th} estimated edge. FOM always has a positive value from 0 to 1. To compare the quantitative performance of the robustness to noise and Gaussian smoothing, two plots are shown using respective edge detection results.

A ground truth edge map is required to evaluate the FOM. Therefore, a synthetic image is used, which contains a circle whose edges cover all directions. Fig. 6 shows two plots: Fig. 6(a) shows an FOM graph as a function of the signal to noise ratio (SNR) whereas Fig. 6(b) illustrates an FOM graph as a function of a scale parameter σ of a Gaussian smoothing filter with fixed SNR (= 15 dB). FOM largely depends on the parameters of each edge detector. In Fig. 6(a), each parameter (threshold value or σ of a smoothing filter) is selected in such a way that the total number of edge pixels detected by one method is similar to that by other methods (number of ground truth edge pixels of the synthetic circle image is 553 and the maximum difference of the numbers of edge pixels detected by different edge detectors is less than 20). Fig. 6(a) shows that the proposed method is a little bit more sensitive to noise than Canny's method when the SNR is less than 25 dB, however, less sensitive than the other methods (Sobel and LoG). When the SNR is larger than 25 dB, the proposed method is better than any other methods. In Fig. 6(b), to compare the effectiveness of the Gaussian filtering to noisy images, the SNR is set to 15 dB and the other parameters are selected in the same way as in Fig. 6(a). Fig. 6(b) shows that the proposed method is more robust to blurring than LoG, however, less robust than Canny's method. Note that in general edge detectors based on the second-order derivative are sensitive to noise. However, the proposed method shows a relatively better performance than LoG and for some σ value better performance than even Canny's method.



(a) (b)

Fig. 6. FOM plots. (a) FOM as a function of the SNR of the input image (additive Gaussian noise case), (b) FOM plot as a function of the scale parameter σ of the Gaussian smoothing filter (SNR=15 dB).



Fig. 7. Estimated edge maps (256×256, Lena image). (a) Zero-crossings of LoG ($\sigma = 0.7$, number of edge pixels=5233). (b) Sobel operator (number of edge pixels=5298). (c) Canny operator ($\sigma = 0.7$, number of edge pixels=5291). (d) Proposed method ($\sigma = 0.7$, number of edge pixels=5237).

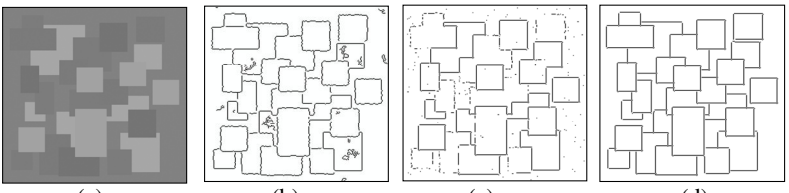


Fig. 8. Estimated edge maps (256×256, Synthetic). (a) Synthetic color image. (b) Edge map of (a) by Canny operator without color information (number of edge pixels=3132). (c) Edge map of (a) by the proposed method without color information (number of edge pixels=3194). (d) Edge map of (a) by the proposed method with color information (number of edge pixels=3243).

To compare the performance for real images, the 256×256 gray-evel Lena image is used. Fig. 7 shows the edge maps detected by each method. Figs. 7(a), 7(b), 7(c), and 7(d) show edge detection results by LoG, Sobel, Canny, and the proposed method, respectively. As in the previous experiment, edge maps with similar numbers of edge points have been compared.

As pointed out in Section 2.2, the proposed edge detector can easily be applied to color images. Fig. 8(a) shows a synthetic color image, Fig. 8(b) shows edge detection results by Canny's method considering only intensity information of a color image, Fig. 8(c) shows edge detection results by the proposed method considering only intensity information of a color image, and Fig. 8(d) shows edge detection results by the proposed method with RGB color information. Fig. 8 shows that the proposed edge detection method is superior to LoG or Sobel. Compared with Canny's method, there are advantages and disadvantages. The proposed method is better than Canny's method in the sense that it can detect high-detailed edges, especially in eye and hair because it uses both magnitudes and direction information. Canny's method is better than the proposed method in the sense that it gives more straight edge lines with less disconnected edges.

The computation time for 256×256 gray-level Lena image of the proposed method (0.274 sec) is smaller than Canny's method (0.328 sec), whereas larger than Sobel (0.141 sec) or LoG (0.234 sec). In addition, the computation time of the proposed method using color information is about 1.7 times larger (0.466 sec) than that using

the intensity only. As shown in experimental results, the proposed method has good performance compared with the other methods and can be applied to various applications requiring fast implementation.

For some applications like object segmentation or recognition, edge magnitudes $l(x, y)$ can be ranked to detect well-focused objects in an image. Especially, for color segmentation, this approach can be easily generalized using the color information.

4 Conclusions

This paper proposes a new approach to edge detection using the modified Laplacian and position mask. Edge detection is based on utilizing the directional information in the complex-valued directional field. Edge detection results by the proposed method are compared with those of existing edge detection methods. The proposed method can detect detailed edges better than the existing methods and be easily generalized to color images, which is motivated by the color edge detection method using vector order statistics [9]. Simulation results show the effectiveness of the proposed algorithm. Further research will focus on the development of the robust extension of the proposed algorithm.

Acknowledgement

This work was supported by Brain Korea 21 Project.

References

1. Park, R.-H.: Complex-Valued Feature Masks by Directional Filtering of 3×3 Compass Feature Masks. Int. J. Pattern Analysis & Applications, 5 (2002) 363–368
2. Vliet, L. J., Young, I. T., Beckers, G. L.: An Edge Detection Model Based on Non-Linear Laplace Filtering. Int. J. Pattern Recog. and Artificial Intell., 25 2 (1998) 63–73
3. Jane, R., Kasturi, R., Schunck, B.G.: Machine Vision. McGraw-Hill, New York USA (1995)
4. Shapiro, L. G., Stockman, G. C.: Computer Vision. Prentice Hall, NJ USA (2000)
5. Marr, D., Hildreth, E.: Theory of Edge Detection. Proc. R. Soc. London UK, Ser. 207 (1980) 187–217
6. Canny, J.: A Computational Approach to Edge Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 8 6 (1986) 679–698
7. Koschan, A., Abidi, M.: Detection and Classification of Edges in Color Images, IEEE Magazine, 22 1 (2005) 64–73
8. Robinson, G. S.: Color Edge Detection. Proc. SPIE Symp. Advances Image Transmission Techniques, 87 (1976) 126–133
9. Trahanias, P. E., Venetsanopoulos, A. N.: Vector Order Statistics as Color Edge Detectors. IEEE Trans. Systems, Man, and Cybernetics, 26 1 (1996) 135–143
10. Abdou, I. E., Pratt, W. K.: Quantitative Design and Evaluation of Enhancement/ Thresholding Edge Detectors. Proc. IEEE, 69 5 (1979) 753–763

Analysis and Design of Graphical Password Techniques

Xiaoyuan Suo, Ying Zhu*, and G. Scott Owen

Department of Computer Science
Georgia State University
Atlanta, Georgia, USA
yzhu@cs.gsu.edu

Abstract. In recent years, graphical passwords have become a viable alternative to the traditional text-based passwords due to their superior ease of recall and potential large password space. However, there is very limited research in classifying and mathematically analyzing the graphical password techniques. In this paper, we conduct a comprehensive study of the existing graphical password techniques. We propose the first taxonomy for graphical passwords methods and discuss the important elements in designing them. We present a mathematical analysis of the graphical password space. Finally we present a new graphical password scheme.

1 Introduction

Human factors are often considered the weakest link in a computer security system. The vulnerabilities of the traditional text-based password have been well publicized and documented [1]. As a result, there is a growing interest in finding alternative authentication methods to text-based passwords. Although biometrics based authentication are considered very secure, currently they are still too expensive and not entirely reliable. Another alternative is graphical password, i.e., a technique that uses images in the authentication process to replace or supplement text-based passwords. The use of graphical password is partially motivated by the fact that humans can remember images better than text [2]. Therefore they have the potential to be more easily remembered as well as having a potentially larger password space. In addition to workstation and web log-in applications, graphical passwords have also been applied to ATM machines and mobile devices.

Although many graphical password schemes have been proposed [3], the graphical password development is still in a very early stage. There has not been a thorough analysis of graphical password methods in terms of design space, security issues, and usability issues. In this paper, we present a thorough analysis of the graphical password design space. Our main contribution is to propose the first taxonomy of graphical password techniques. For each category of graphical password techniques, we present a mathematical model to analyze its password space.

During the process of developing the graphical password taxonomy, we have identified a few gaps in the existing techniques. As a result, we have developed a novel graphical password scheme which is also presented in this paper.

* Corresponding author.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 741 – 749, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The rest of the paper is organized as follows. Section 2 will discuss the taxonomy of the graphical password techniques; section 3 will provide the evaluating factors in detail; section 4 will analyze several new applications based on the evaluating factors; in section 5 is the conclusion and possible future work.

2 A Taxonomy for Graphical Passwords

In this section, we will present our taxonomy for the graphical password techniques. Figure 1 illustrates the classification of the graphical password techniques and the important factors under each category. (Please refer to [3] for a comprehensive survey of the existing graphical password techniques.)

In general, the current graphical password schemes can be classified into two classes: recognition based and recall based techniques.

2.1 Recognition Based Techniques

Using recognition-based techniques, a user is presented with a set of images and the user passes the authentication by recognizing and identifying the images he or she selected during the registration stage. There are two important factors for designing recognition based graphical passwords: visualization and interaction.

Visualization. Visualization has three main parameters: content, spatial layout, and distraction images.

In recognition based techniques, the content of the visualization is a set of images, which can be random pictures [4], human faces [5], or users' favorite images.

Spatial layout is an important factor that affects usability. In most graphical password techniques, images are presented in a matrix. But in some schemes, images are randomly placed in a 2D space. When there are too many images on the screen, the display gets overly crowded and the usability will be severely decreased – finding a target image becomes difficult. In many cases, images are grouped in pages and users would need to "page down" during the authentication process. This often makes entering graphical passwords slower than text-based ones.

During the authentication stage, the user is asked to choose from a set of images, which contains a few pass-images and many distraction images. The (number of pass-images)/(number of distraction images) ratio greatly affects the size of the password space.

Interaction. During the authentication stage, the user can interact with the system either with a mouse, keyboard, or stylus pen. With mouse, there are two options. The user can either use the mouse to select a target image, or the target image is not visually selected. For example, the program can animate the image and when the target image hits an object, the user would click the mouse button. The interaction technique also greatly affects the security aspect of the password scheme. For example, key loggers are often used to secretly record users' passwords. The sound of keyboard typing can also be used to identify the keys being typed. In this regard,

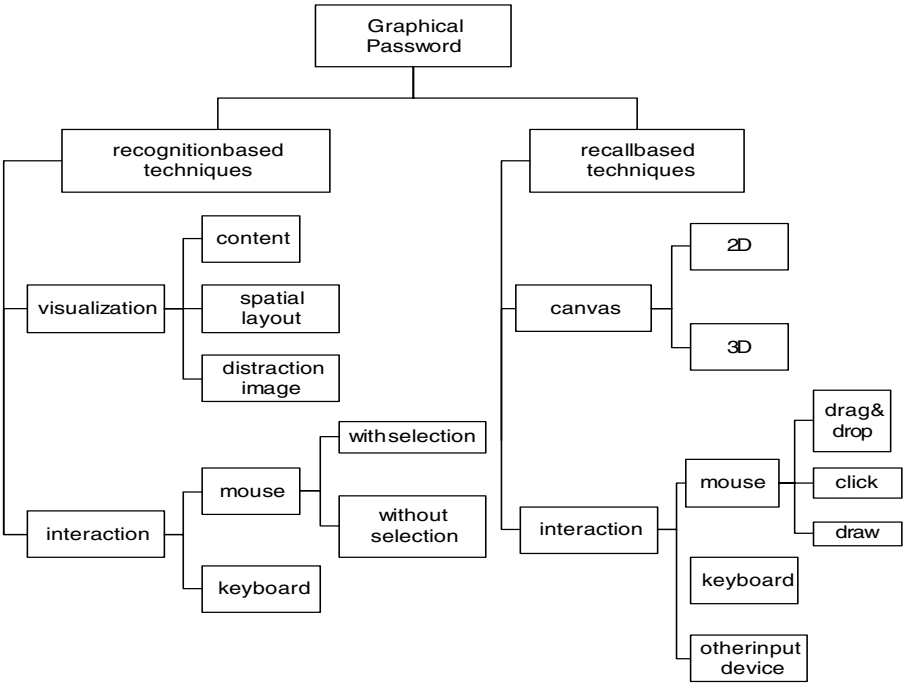


Fig. 1. A taxonomy of graphical passwords

mouse based interactions seem to be more resistant to such attacks, which is one of the strengths of graphical passwords.

2.2 Recall Based Techniques

Using recall-based techniques, a user is asked to reproduce something (e.g. a drawing or a sequence of actions) that he or she created or selected earlier during the registration stage. In this category, there are two important factors: canvas and interaction.

Canvas. Here we define canvas as a space where users can use input devices to draw, mark, or write things. Canvas can be either 2-D or 3-D, although we have not found any 3-D based graphical password techniques. For example, some graphical password schemes require users to recall and draw a picture on the canvas [6]. Other techniques require users to click on a sequence of pre-selected positions on an image [7]. In this case, the image is the canvas.

Simply put, recall based graphical password techniques can be defined as "repeating a sequence of actions". The canvas is where the sequence of user actions is visualized. The traditional text-based password method falls under this category because typing a password is "repeating a sequence of key strokes." The small text box that displays the (sometimes masked) password is a canvas. Therefore in our

taxonomy, the text-based password method is a special case of recall based graphical password techniques.

Interaction. Again, user interaction can be conducted through various input devices, such as a keyboard, mouse, or stylus pen. Other devices, such as a joystick may be used too, although we have not found any technique that uses it.

Typical recall based graphical passwords require users to either draw [6], or select (the pixels on the screen)[7]. We have proposed a new graphical password technique that requires drag and drop (see section 4 for details).

3 Analysis

In this section, we analyze the two main categories of graphical password techniques from several perspectives. We focus mainly on security and usability but also take into consideration system and communication issues. For security, we focus on password space and the strength of the password. For usability, we focus on the easiness of registration and authentication.

3.1 Recognition Based Techniques

Security. The password space of the recognition based techniques largely depends on the size of the content. Most recognition based techniques do not consider the order of the selection. They often involve many rounds of authentication with users going through several pages of images [5, 8].

Table 1. Important parameters in recognition based techniques

Description	Variable
Total number of pictures	n
Password length	l
Maximum password length	m

The password space for recognition based technique is given below, assuming a picture can be selected more than once.

$$password_space = \sum_{l=1}^{m} \binom{n+l-1}{n-1} = \sum_{l=1}^{m} \frac{(n+l-1)!}{l!(n-1)!}. \quad (1)$$

The chances of creating a weak password are high using recognition based passwords. The work by Davis, et al. [9] found obvious patterns among the PassFace password [5]. For example, most users tend to choose faces of people from the same race.

Random art [4] could be one solution for addressing the weak password, i.e. the user has no familiarity with any of the password images. However doing this may decrease usability by making the password hard to remember.

Usability. Content (images), spatial layout of the content, and input devices are all important factors that influence usability. For example, users' favorite pictures tend to be easier to remember but also easier to be guessed by attackers. Too many distraction pictures tend to slow down the authentication process. Several existing techniques are proven to have usability issues due to the crowded content arrangement [10].

Other issues. An overly large storage requirement is a significant issue for recognition based techniques, since the size of a typical picture is much larger than the equivalent text. In order to achieve the larger password space, thousands of pictures need to be stored at one time. Sending a large number of pictures over the network is also a problem for low speed networks.

3.2 Recall Based Techniques

Security. It is difficult to calculate the password space of a recall based technique, since there are many variations in recall based techniques. Here we demonstrate a general mathematical model. Some of the most important elements that determine the password space of a recall based technique are listed in table 2.

Table 2. Important parameters of recall based techniques

Description	Variable
Number of units (pixels) on canvas	n
Password length	l
Maximum password length	m

The maximum password space that a recall based technique can have is extremely large, since certain techniques require the user not only to have the proper shape of drawing or clicking, but also the proper sequence of drawing [6]. If the drawing allows the same unit (or pixel) to be chosen multiple times, the password space for a specific password of length l is:

$$password_space_\max = \sum_{l=1}^{m} n^l \tag{2}$$

If a password scheme does not allow the drawing to pass the same pixel multiple times, or if it requires mouse clicking to match the pre-registered sequence, the password space is the smallest:

$$password_space_\min = \sum_{l=1}^{n} \frac{n!}{(n-l)!}. \tag{3}$$

Thus the password space for recall based approach is:

$$\sum_{l=1}^{n} \frac{n!}{(n-l)!} \le pasword_space \le \sum_{l=1}^{m} n^l. \tag{4}$$

Usability. A major complaint in recall based graphical passwords is that it is difficult to draw shapes with a mouse. Most users are not familiar with using a mouse as a drawing tool. However, on mobile devices, a stylus pen is a good choice for such techniques.

4 The Design of New Graphical Password Techniques

In the process of developing the taxonomy and analyzing the design space of graphical password, we have identified a few gaps in the existing graphical password methods. This led to the design of a new graphical password scheme.

The existing graphical password methods are either recognition based or recall based. We have developed a graphical password technique that is a combination of both recognition and recall based techniques.

Design. Our graphical password scheme is called Recall-a-Formation (RAF). Two tables are involved -- a data table and an input table. Figure 2 shows a Java based interface for RAF – on the right hand side is a 8×8 data table and on the left hand side is the input table. The data table is the canvas where the user would recall and repeat a pre-registered formation of icons by first selecting the icons from the input table and then dragging and dropping them in the data table. In the input table, users need to recognize and select the target icons among the distraction icons – a recall based approach. The input table can be very large, containing pages of icons of different themes for users to choose from. In our prototype system, we implement four different themes: fruits, animals, stars, and system icons (figure 3).

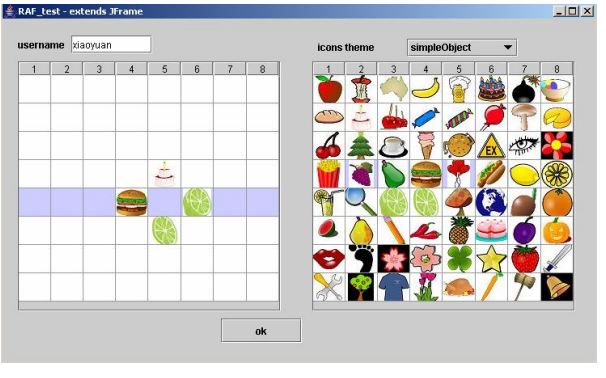


Fig. 2. RAF main interface

Password space analysis. Our implementation has an 8×8 input table, and an 8×8 data table. Since some of the data table cells may be empty, there are 65 different choices for each data table cell. The input table may have multiple pages. Therefore the password space of RAF is $64^{(64\times number_of_theme_pages)+1}$.

Now consider a text-based password. Suppose we use all 83 printable characters, and then the password space of an 8-character password is $8^{83} = 9.046 \times 10^{74}$. Even using only one theme page, the password space of RAF is $64^{65} = 2.5217 \times 10^{117}$, which is much larger than that of text-based password.



Fig. 3. Different themes of icons in the input table

Preliminary user study. We have conducted a preliminary user study among 30 different users. The users were between 20 to 30 years old and most were college students. We asked the users to use our RAF system to create a graphical password and at the same time chose a text password of no more than 64 digits. The users were asked to recall their password the next day using our system. Among the 30 users, 11 people remembered their exact formation and the exact icons they have chosen. Fifteen remembered more than half of the icons they inputted the day before. Among these 15 people, nine people recognized their icons exactly, but did not recall the formation. On the other hand, 100% of the 30 users remembered their exact text-based password. However, among the 30 text-based passwords, 29 were easy-to-remember words or numbers, such as users' names and birthday dates.

Among the 30 users, three people reported that they needed to write the RAF formation on a piece of paper to assist their memory. One person reported that she only inputted one object into the input table. Almost 90% of the users reported they preferred the single-object type theme to the multi-object type theme, since they could remember the formation better for a single-objected theme. Eighty percent of the users chose no more than four different objects in the whole input table, and the icons chosen were the simplest objects, i.e. simplest stars, circular objects, or their favorite item. Indeed the single objected theme, i.e. "star", was the most popular theme choice.

The user study has helped identify usability issues of the RAF method. Currently we are continuing to improve the usability of RAF to make it easier to remember.

5 Conclusion

In this paper we analyze the design space of graphical password techniques and propose the first taxonomy for these techniques. We discussed the security and usability issues of the two main categories of graphical password methods – recognition based and recall based graphical password techniques.

We also presented a novel graphical password scheme, called "Recall-a-Formation" (RAF). RAF is the first graphical password scheme that combines a recall based technique with a recognition based technique. We showed that a relatively small RAF configuration can achieve a much larger password space than a text-based password method. We are continuing to improve the usability of the RAF method.

Graphical password authentication has a lot of potential as an alternative to text based and biometrics based authentication. Our taxonomy and analysis provides new insight into the design space of graphical password methods, and we believe such insight will lead to more novel graphical password techniques.

Acknowledgment

We would like to thank Dr. Raheem A. Beyah for his helpful discussions.

References

1. Adams, A., Sasse, M. A.: Users are not the enemy: why users compromise computer security mechanisms and how to take remedial measures, Communications of the ACM, Vol. 42. (1999):41-6
2. Shepard, R.N.: Recognition memory for words, sentences, and pictures, Journal of Verbal Learning and Verbal Behavior 6 (1967):156-63
3. Suo, X., Zhu, Y., Owen, G.S.: Graphical password: a survey, in Proceedings of the Annual Computer Security Applications Conference (ACSAC), IEEE (2005)
4. Dhamija, R.,: Hash visualization in user authentication, in Proceedings of the ACM Conference On Human Factors In Computer Systems (2000)
5. RealUser,: www.realuser.com, (last retrieved in August 2006)
6. Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K., Rubin, A. D.: The design and analysis of graphical passwords, in Proceedings of the 8th USENIX Security Symposium (1999)
7. Wiedenbeck, S., Waters, J., Birget, J. C., Brodskiy, A., Memon, N.: PassPoints: design and longitudinal evaluation of a graphical password system, International Journal of Human Computer Studies. International Journal of Human-Computer Studies, 63 (2005) 102-127
8. Wayne J.: Authenticating mobile device users through image selection, in Advances in Learning, Commerce and Security, K. Morgan & M. J. Spector (Editors), WIT Press (2004)
9. Davis, D., Monrose, F., Reiter, M. K.: On user choice in graphical password schemes, in Proceedings of the 13th USENIX Security Symposium (2004)

10. Wiedenbeck, S., Waters, J., Sobrado, L., Birget, J.C.: Design and evaluation of a shoulder-surfing resistant graphical password scheme', in Proceedings of Advanced Visual Interfaces (AVI) (2006)
11. Syukri, A. F., Okamoto, E., Mambo, M.: A user identification system using signature written with mouse, in Proceedings of the 3rd Australasian Conference on Information Security and Privacy (1998)
12. Dourish, P., Redmiles, D.: An approach to usable security based on event monitoring and visualization, in Proceedings of New Security Paradigms Workshop (2002)

Viewing Scenes Occluded by Smoke

Arturo Donate and Eraldo Ribeiro

Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32905, USA
{adonate, eribeiro}@cs.fit.edu

Abstract. In this paper, we focus on the problem of reconstructing images of scenes occluded by thick smoke. We propose a simple and effective algorithm that creates a single clear image of the scene given only a video sequence as input. Our method is based on two key observations. First, an increase in smoke density induces a decrease in both image contrast and color saturation. Measuring the decay of the high-frequency content in each video frame provides an effective way of quantifying the amount of contrast reduction. Secondly, the dynamic nature of the smoke causes the scene to be partially visible at times. By dividing the video sequence into subregions, our method is able to select the subregion-frame containing the least amount of smoke occlusion over time. Current experiments on different data sets show very promising results.

1 Introduction

In this paper, we focus on the problem of image recovery in static scenes containing large amounts of thick smoke. Our goal is to generate a single clear image of the scene given only a video as input. We assume that no previous knowledge of the underlying scene is available, and both the camera and the observed scene are static. We base our study on two observations. First, an increase in smoke density induces a decrease in the contrast of images. Secondly, the dynamic nature of the smoke causes the scene to be partially visible at times in different parts of the image. Under these conditions, the smoke will considerably reduce the visibility of the scene. This scenario occurs in various real-world situations such as explosions, fire-related incidents, and armed conflicts.

We propose an algorithm that effectively reconstructs a clear image of the occluded scene. Our algorithm begins by separating video frames into subregions. Each subregion is analyzed in the frequency domain as well as the saturation channel of the HSV colorspace. We use a highpass filter to compare high-frequency information about each frame. Our experiments show that measuring the decay of high-frequency content as well as the reduction in chrominance of an image provides a successful approach to determining the level of smoke occlusion in each frame. For each subregion, the algorithm finds the frame containing the least amount of occlusion. The final reconstructed image is a mosaic of all the clear frames. This paper was inspired by the work of Efros *et al.* [1] on removing image distortion caused by water waves.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 750–759, 2006.
© Springer-Verlag Berlin Heidelberg 2006



Fig. 1. Subset of frames from an experimental video sequence

The method is tested on different data sets created in the laboratory, each containing varying amounts of smoke as well as different number of image frames in order to present the robustness of the method. Current results show great success in generating clear images of the underlying scenes.

The remainder of the paper is organized as follows. Section 2 provides a review of the literature. Section 3 describes some of the effects of smoke occlusion in videos. An overview of the method is presented in Section 4. Experiments and results are shown in Section 5. Finally, Section 6 concludes the paper and presents directions for future research.

2 Related Work

The computer vision literature directly related to the problem of recovering a clear image of a smoke-occluded scene is somewhat limited. Most of the related works address the problems of smoke and fire detection in both digital [2] and satellite imagery [3,4], removal of weathering effects [5,6,7,8,9,10], and smoke synthesis [11]. The computer graphics community, on the other hand, has mostly addressed the problem of modeling and rendering realistic smoke.

Recently, there has been significant interest in computer vision methods for removing weathering effects such as fog and haze from images. An elegant modeling of the effects of weather conditions in images was proposed by Narasimhan and Nayar [5,6]. Their work is based on physical models that describe how light, colors, and contrast behave under certain weather conditions. In [5], Narasimhan and Nayar present two physics-based models that can be used for contrast restoration in images containing uniformly bad weather. The attenuation model describes how light weakens as it travels from a point on the scene to the observer, and the airlight model measures how the atmosphere can reflect environmental illuminations to the observer. These models provide a way of quantifying the decay in contrast of images with poor visibility conditions due to weather effects. In [6], Narasimhan and Nayar use the models introduced in [5] for removing weather effects such as fog and haze from a single image of a scene without

precise knowledge of the weather, and with minimal user input. Similarly, in [8] Narasimhan and Nayar present models for extracting the 3D structure of a scene occluded by bad weather. In the work presented in [10], Shwartz *et al.* approach the problem of blindly recovering the parameters needed for separating airlight from other measurements. Here, the authors describe a method that successfully recover the contrast of a hazy or foggy image with no user interaction and without having the sky in the image frame.

Other related problem is the automatic detection of smoke and fire in video sequences and satellite images. Treyin *et al.* [2] developed a real-time image-based smoke detection system that uses measurements on edge sharpness. The authors relate the effects of smoke's varying transparency to changes in image edge sharpness and saturation values of colors in the scene using a wavelet decomposition technique. The method requires the availability of a clean view of the observed scene. The problem of detecting smoke and fire in satellite imagery was addressed by [4,3]. Chung and Le [4] studied the possibility of automatic detection forest fire smoke plumes in satellite images. Farser *et al.* [3] use a neural network to analyze satellite imagery in order to classify the scenes into smoke, cloud, or clear background, in an attempt to automatically detect forest fires.

3 Effects of Smoke in Videos

In this section, we discuss some of the main effects of smoke in video sequences. Here, we focus on three main effects. First, we discuss how smoke occlusion causes variations in color saturation. Secondly, we study the effects of smoke on the image's frequency domain representation. Finally, smoke motion gives rise to image edges of weak intensity while weakening the response of static scene edges. Our method combines measurements of color saturation, frequency-domain energy, and edge maps to help select the best set of subregions of a frame sequence that can be stitched into a single clear view of the observed scene. In the analysis that follows, we assume that we have a video of a static scene occluded with a layer of moving smoke of varying thickness levels.

Smoke transparency and color. Smoke transparency will allow the underlying captured scene to be visible sporadically over time depending on the smoke thickness. Previous works have shown that variations in smoke transparency can be directly related to variations in both color contrast and color saturation. For example, Narasimhan and Nayar [5,6] provide color chrominance models that explain how these changes can be used to restore weathered images containing fog and haze. A decrease in chrominance values (i.e., hue and saturation components) has also been associated with the presence of smoke in images [2]. Figure 2 shows a plot of the total image saturation for a series of video frames with increasing levels of smoke occlusion. In the figure, the total saturation value decreases with the increase of smoke occlusion. The original video frames are also shown in the figure.

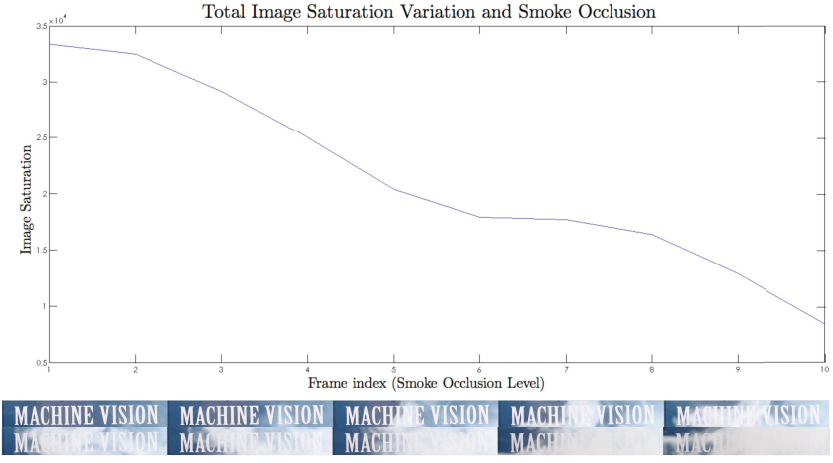


Fig. 2. Decreasing saturation due to smoke occlusion

Frequency domain analysis. The presence of the smoke in the observed scene causes a decrease in the color contrast of the video frames. The contrast reduction effect translates to a corresponding reduction in the image's high-frequency content [2,5,6,7,9]. The varying degrees of smoke thickness correspond to changes in both image transparency and contrast variation. This suggests that the analysis of the high-frequency content in the power spectrum of each frame in the video can provide a way to determine the level of smoke occlusion in each frame relative to the other frames of the video. The high-frequency power spectrum of an image can be computed by:

$$P'(u,v) = \|F(u,v)H(u,v)\|^2 \tag{1}$$

where F the Fourier transform of the image $I(x,y)$ defined by:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x,v)e^{-2\pi j(ux+vy)}dxdy \tag{2}$$

and H is a high-pass filter. In this paper, we use a simple Butterworth high-pass filter [12] with cut-frequency D_0:

$$H(u,v) = \frac{1}{1+[D_0/D(u,v)]^2} \tag{3}$$

Finally, we can express the remaining high-frequencies in polar coordinates using the function $P(r,\theta)$, where P is the power spectrum function, and r and θ are the polar coordinate parameters [12]. One-dimensional histograms of the polar power spectrum can then be formed by:

$$S(r) = \sum_{\theta=0}^{\pi} S(r,\theta) \tag{4}$$

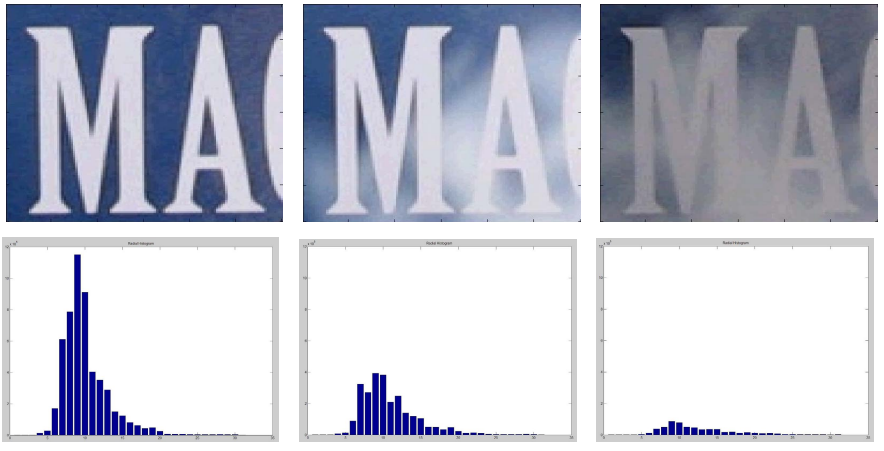


Fig. 3. Images with increasing levels of occlusion and their corresponding angular frequency histograms. The reduction in contrast causes a drop in the high-frequency content.

The total amount of high-frequency energy for each video frame is obtained by summing the bins of these histograms. Figure 3 illustrates this concept.

It is worth mentioning that measuring the decay of high-frequency content alone will not work well for quantifying the level of contrast reduction in images of different scenes. As pointed out by Field and Brady [13], a simple increase in the number of edges will result in an increase of high-frequency content, with no visual change in the degree of contrast in the images. This limitation may be overcome by using phase coherence measurements as described in [14]. A discussion on using the advantages of phase coherence is beyond the scope of this paper.

Edge accumulation map. The effects of chrominance and high-frequency variation represent measurements of the presence of smoke in an image. However, in the case of smoke occlusion, the high-frequency content attenuation is not caused by actual blur of edges. As a result, image edges will not suffer from localization uncertainty but they will appear weaker in some regions across the image. Additionally, for static scenes, some noisy edges will be temporally created by the gradient between the smoke cloud and a flat image background. These weak edges are likely to increase the image's high-frequency content. Here, we suggest the creation of an edge map obtained from the accumulation of edge evidence over the whole frame sequence. The information produced by this edge map will be used by our algorithm and it will be discussed later in this paper. An example of the accumulated edge map is shown in Figure 4.



Fig. 4. Accumulated edge map of static and dynamic "smokey" edges. Weaker edges are produced by the moving smoke.

4 Method Overview

In this section, we describe the details of our method. The main goal of our algorithm is to create a single clear image of an observed scene originally occluded by smoke given a video as input. This is accomplished by dividing all video frames into a number of non- overlapping subregions. These subregions will become "smaller videos" containing the same number of frames as the original video. The algorithm analyzes each subregion separately to select the single subregion-frame containing the least amount of smoke. In order to measure the relative amount of smoke in the image, we propose a function that combines chrominance information and frequency-domain measurements.

We commence by considering a video of N frames divided into K subregions. Our method of reconstructing the scene analyzes each subregion individually to select the frame containing the least amount of smoke occlusion. Let $q(\mathbf{x})$ be a function that measures the quality of a given image frame $\mathbf{x}$ such that an increase in the amount of smoke occlusion in $\mathbf{x}$ causes a reduction in the value of $q(\mathbf{x})$. We define $q(\mathbf{x})$ as follows:

$$q(\mathbf{x}) = \left\| \sum_{r} S(r) \right\|^{2p} \times \sum \text{saturation}(\mathbf{x}) \tag{5}$$

where

$$p = \begin{cases} 1, \text{edge}(\mathbf{x}) = 1 \\ 0, \text{otherwise} \end{cases}$$

Equation 5 has two main components. The first component corresponds to the total energy of the radial power spectrum representation of the image after the application of a high-pass filter as given by Equation 4. The second component corresponds to the total sum of the values in the saturation channel of the image. Here, we make two critical observations about the interplay between these two components. First, as the amount of smoke occlusion increases, the saturation values of the images decrease. While this property provides a simple approach for detecting smoke, it does not work well if the original colors in the scene have low saturation values. On the other hand, if the underlying scene contains no edge gradients, image frames with smoke will contain more high-frequency energy than clear frames (i.e., frames without smoke). This is due to the fact that

Algorithm 1. Smoke-occluded scene reconstruction

Given N-frame video sequence:

```
Divide video into K subregions
for Every subregion do
    Build an edgemap
    for Every frame in the subregion do
        Calculate high-frequency from polar histograms
        Calculate the sum of the values in the saturation channel
        Calculate a quality measurement from frequency and saturation measurements
    end for
    Extract frame with greatest quality measurement
end for
Generate reconstructed image I from retrieved subregions
```

partial smoke coverage will form a gradient between the boundary of the smoke and the flat background of a video frame. As a result, measuring smoke occlusion using high-frequency content only can produce erroneous results when analyzing image subregions with no edge gradients on the original image. We address this problem by adding the exponential term p in the first component of Equation 5. This term determines whether the high-frequency measurements should be taken into consideration or not. We set p to one if the analyzed subregion contains any edges in the accumulated edge map as described previously in this paper, and zero otherwise. The idea of this term is to allow the influence of high frequency measurements only when the subregion being analyzed contain static edges. If no edges are present, p will be set to zero and the quality of the frame will depend solely on color saturation. With $q(\mathbf{x})$ defined, the problem of finding the clearest frame in a subregion can be posed as a subregion-frame selection algorithm such that:

$$i' = \arg\max_{i} q(\mathbf{s}_i) \tag{6}$$

where $\mathbf{s}$ is the subregion containing the N frames ($\mathbf{s}_i \in \mathbf{s}$), and i' is the index corresponding to the frame in $\mathbf{s}$ with the least amount of smoke occlusion. Therefore, the problem now becomes a selection of subregion with maximum quality given by the measurement $q(\mathbf{x})$. The main steps of this method are listed in Algorithm 1. This process is applied to each subregion of the video. Once the "smokeless" subregion selection process is completed, a mosaic is formed from all the selected frames to create the final resulting image of the observed scene. The algorithm does not require any previous knowledge of the underlying scene.

5 Experiments and Results

In this section, we present experimental results on two different data sets recorded in our laboratory. Our objective is to show the effectiveness of the proposed method in selecting the cleanest frame- regions in the videos.



(a) original frame sequence



(b) mean



(c) median



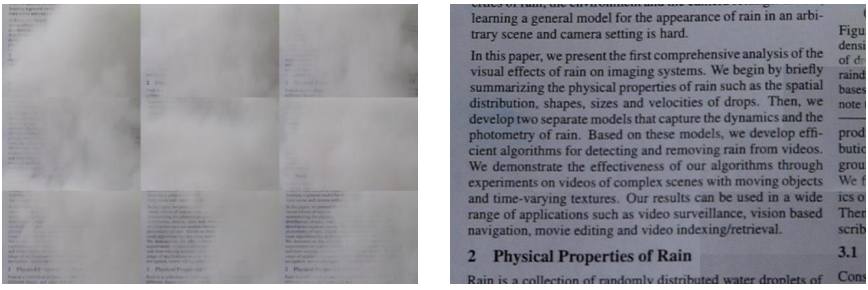
(d) our method

Fig. 5. Subset of frames showing occlusion due to smoke with reconstructions using (a) a simple temporal average pixel value method, (b) the median over time, and (c) our method

In the experiments, we analyze three 100-frame video sequences containing varying amounts of smoke occlusion. Some frames contain a small amount of smoke, while other frames contain very thick layers of smoke that completely obscure the observed scene. The first dataset consists of a video of a machine vision book with an overall lighter quantity of smoke occlusion. A selection of the frames in this dataset is shown in Figure 5(a). One of the simplest ways to approach this problem is to calculate the mean (or median) of each pixel over time. This is a very simplistic approach that will result in poorly reconstructed images, especially in the case of video sequences containing scene occlusions for the majority of the frames due to heavy clouds of smoke (as seen in Figure 5). We present the results for the mean and median calculations only as an example.

The second dataset is a video of a printed section of text from the paper in [7]. The frames in this dataset contain smoke in larger amounts and of larger density than the first dataset. Here, the scene was under occlusion for approximately 95% of the video, and each section of the scene was only visible for a few frames over the entire sequence. The purpose of this experiment was to show that increasing the number of occluded frames has no ill effect on the algorithm, as long as each subregion of the scene is clearly visible in at least one frame of the video. Additionally, these video frames contain mostly frequency information (i.e., saturation values over the frames were relatively low).

The third dataset is a video containing the largest amount of smoke occlusion. This time, the camera was placed further back in order to cover a larger area of the scene. The camera position allowed for capture of larger textured and untextured regions in the same scene. Results are shown in Figure 7. In this dataset, there is only a single frame that does not contain smoke. The idea behind this experiment is to show the effectiveness of the method to find a clear



(a) (b)

Fig. 6. Subset of video frames from paper experiment. (a) Sample of original video frames. (b) Final reconstructed image.



(a) (b)

Fig. 7. Dataset containing nearly-complete occlusion. (a) Original video frames. (b) Final reconstructed image.

frame of the scene regardless of the amount of occlusion in the other frames. Although the complexity of this dataset is greater than the previous ones, the algorithm is equally successful in generating an accurate reconstruction.

6 Conclusions

In this paper, we focus on the problem of image recovery in static scenes containing large amounts of smoke occlusion. Our main goal was to develop an algorithm to generate a single clear image of the scene given only a video as input, with no previous knowledge about the underlying scene.

Our analysis shows that smoke occlusion causes a reduction in the frequency content of images, as well as a reduction of chrominance values. Our proposed algorithm attempts to quantify the relative amount of occlusion in a given frame by measuring reductions in chrominance as well as high-frequency content. As a result, our algorithm successfully reconstructs clear images of occluded scenes using only a video sequence as input. The experiments presented here show the

feasibility of our approach. It is important to note that for a given subregion, if the input video does not contain a single clear view, then the algorithm will fail to generate a clear view of that subregion in the final reconstruction since it does not attempt to actually remove the smoke present in the scene.

We are currently working on an extension to the proposed method to work with a moving cameras. This is a much more complex problem as feature-based image alignment and mosaicking methods might are likely to fail due to possible erroneous feature detection on the moving smoke regions.

References

1. Efros, A., Isler, V., Shi, J., Visontai, M.: Seeing through water. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA (2004) 393–400
2. Treyin, B.U., Dedeoglu, Y., Cetin, A.E.: Wavelet based real-time smoke detection in video. In: European Signal Processing Conference. (2005)
3. Li, Z., Khananian, A., Fraser, R.H., Cihlar, J.: Automatic detection of fire smoke using artificial neural networks and threshold approaches applied to avhrr imagery. IEEE Transactions on Geoscience and Remote Sensing **39** (2001) 1859–1870
4. Chung, Y.S., Le, H.V.: Detection of forest-fire smoke plumes by satellite imagery. Atmos. Environ. **18** (1984) 2143–2151
5. Narasimhan, S.G., Nayar, S.K.: Contrast restoration of weather degraded images. IEEE PAMI **25** (2003) 713–724
6. Narasimhan, S.G., Nayar, S.K.: Interactive deweathering of an image using physical models. In: IEEE Workshop on Color and Photometric Methods in Computer Vision, In Conjunction with ICCV. (2003)
7. Garg, K., Nayar, S.K.: Detection and removal of rain from videos. In: International Conference on Computer Vision and Pattern Recognition. (2004) 528–535
8. Nayar, S.K., Narasimhan, S.G.: Vision in bad weather. In: International Conference on Computer Vision. (1999)
9. Schechner, Y.Y., Narasimhan, S.G., Nayar, S.K.: Polarization-based vision through haze. Applied Optics **42** (2003) 511–525
10. Shwartz, S., Namer, E., Schechner, Y.Y.: Blind haze separation. In: International Conference on Computer Vision and Pattern Recognition. (2006)
11. Fedkiw, R., Stam, J., Jensen, H.W.: Visual simulation of smoke. In: SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM Press (2001) 15–22
12. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, Reading, MA (1992)
13. Field, D., Brady, N.: Visual sensitivity, blur and the sources of variability in the amplitude spectra of natural scenes. Vision Research **37** (1997) 3367–3383
14. Wang, Z., Simoncelli, E.P.: Local phase coherence and the perception of blur. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA (2004)

A Non-photorealistic Rendering of Seurat's Pointillism

Hui-Lin Yang and Chuan-Kai Yang

National Taiwan University of Science and Technology, Taipei, 106, Taiwan, ROC

Abstract. In recent years, there has been a trend on simulating impressionism with computers. Among the various styles of impressionism, we are particularly interested in simulating the style of *pointillism*, especially the style presented by *Georges-Pierre Seurat*, as he was deemed the founder of pointillism. The reason that his style attracts us is twofold. First, the painting process of pointillism is extremely laborious, so simulating his painting style by computers is desired. Second, though several existing impressionism algorithms may approximate pointillism with point-like strokes, some delicate features frequently observed in Seurat's paintings are still not satisfactorily reflected by those general schemes. To achieve simulating Seurat's painting style, we made careful observations on all accessible Seurat's paintings and extract from them some important features, such as the few primitive colors, point sizes, and the effects of complementary colors and halos. These features have been successfully simulated and results are compared with not only Seurat's existing paintings, but also with previous attempted simulations.

1 Introduction

As one of the most famous non-photorealistic painting styles, *impressionism* concerns mainly the interaction between lights and shadows, or what the so called *impression* that a scene can bring to a person, without placing too much emphasis on the irrelevant details. Among the previously simulated impressionism styles, we are especially interested in *pointillism*, whose only painting primitive is *point*. As one of the most renowned painters in pointillism, Seurat, in particular, attracts our attention for two reasons. First, it is well known that a pointillism painting usually takes a very long time to finish, as it involves a very tedious process, which is potentially cut out for computers. Second, though generic techniques exist to approximate pointillism effects, such as [1,2,3,4], they are often too general to faithfully reflect all the features presented in Seurat's works. For example, there is a noticeable feature called *halo* that frequently appears in his paintings but has not been successfully simulated so far. To accurately emulate Seurat's painting style, we have made careful and detailed observations on numerous paintings of his in order to extract the common or important features that distinguish his work from others. To prove the success of our efforts, simulated images are presented at the end of the paper, together with their comparisons with the results produced by other existing techniques. Several Seurats' own paintings, after digitization, are also shown for serving as the comparison ground truth.

The rest of the paper is organized as follows. Section 2 reviews some existing works related to this study. Section 3 briefly describes what we observed from Seurat's paintings, i.e., the distinct features. Section 4 discusses how we simulate Seurat's painting

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 760–769, 2006.
© Springer-Verlag Berlin Heidelberg 2006

styles based on our observed features. Section 5 presents our simulation results, which are also compared against previous works, accompanied by existing Seurat's paintings for being the ground truth. Section 6 concludes this paper and hints for potential future directions.

2 Related Work

It has been more than a decade since the name of *non-photorealistic rendering* was first brought to the computer graphics community by Winkenback et al. [5] in 1994, and a huge number of papers have been generated since then. Due to the limit of space, readers who are interested in learning more information regarding NPR are referred to the website at http://www.red3d.com/cwr/npr/, written by Craig W. Reynolds, who did an excellent and substantial survey on many aspects of NPR. Since our main goal is to convert existing images into pointillistic ones, we will pay attentions only to those similar studies.

Probably the first work to simulate pointillistic effect, Hertzmann proposed to mimic the hand-painted features by curved brush strokes, and in particular, he attempted the pointillistic effects by placing dense circles with perturbed hue and saturation values [2]. A layered structure is employed to process an image layer by layer, with progressively smaller brushes. Litwinowicz also proposed to simulate impressionism by varying the attributes of strokes [4]. Similarly, Hays et al. also presented an elaborated scheme to control various attributes of strokes, which lends itself to stylized images or videos [1]. Hertzmann proposed a rather different approach for simulating a stylized painting by performing *image analogies* [6]; however, many believe that his method cannot really produce better results than with a more domain-specific approach.

Most of the aforementioned approaches concentrate on simulating the strokes, while leaving color variation to random perturbation. Luong et al.'s work [3] adopted a different approach, where they pick colors in an *isoluminant* manner, so that it matches well with our perception system. Similarly, Jing et al. [7] divided an input image into tiles, within which colors are also perturbed in a an isoluminant way.

3 Seurat's Pointillism

Georges-Pierre Seurat (December 2, 1859 – March 29, 1891) was deemed as the founder of *neoimpressionism*, *pointillism*, or *divisionism*, and in fact each of these names has its own reason, which will be made clear later.

Eleven Colors. Seurat was known for only using relatively few colors when composing his paintings. While some said that he used only eleven colors, so far to our knowledge, we cannot be sure of what these eleven colors are. However, the traditional rainbow colors are surely included as those colors were recognized long time ago.

Halo Effect. Figure 1 is an example for demonstrating the *halo* effect, while Figure 2 for the *crease edge* effect.



Fig. 1. Left: a partial image of Seurat's *Sunday afternoon on the island of La Grande Jatte* painting. Upper-right: an enlarged portion of the left image marked with red, where one can see how the halo (along the boundary of the lady's dress) and complementary colors (the red dots on the lady's dark dress) take places. Lower-right: another enlarged portion of the left image marked with green, where the green color is juxtaposed with its adjacent colors in Chevreul's color circle (see Figure 3), such as yellow and blue, to make the green color stick out.

Complementary Colors. Another important feature is to utilize complementary colors. The main purpose of such usage is to make the original colors more prominent after being contrasted with their complementary counterparts. Figure 1 gives a demonstration for this. Note that Seurat's concept of complementary colors may be different from ours, as shown by the Chevreul's color circle in Figure 3, which was believed to be used by Seurat.

Divisionism. As mentioned before, pointillism or divisionism, represents a desired color by putting two primitive colors together properly. Figure 2 also demonstrates an example of such.

4 Non-photorealistic Rendering of Seurat's Pointillism

4.1 Color Space Conversion

After careful and detailed examinations of many Seurat's paintings, now the task is to simulate these observed features by computers. Before illustrating our simulations step by step, one thing worth mentioning is the color space conversion. In this work we select the well-known $CIEL^*a^*b^*$ or $CIELAB$ color system as our internal color manipulation format, not only for its decoupling of hue, saturation and lightness, but also for its *perceptual uniformness*, which means that the same amount of value change would result the same amount of visual perception change.

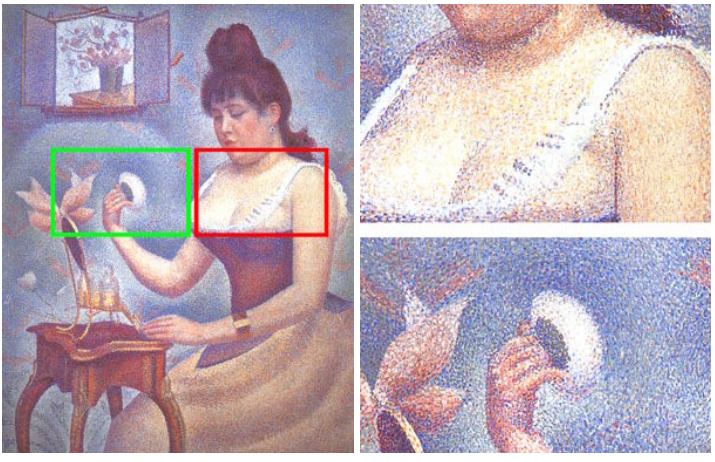


Fig. 2. Left: the image of Seurat's *young woman powdering herself*. Upper-right: an enlarged portion of the left image marked with red, where one can see the crease edge between the woman's body and her left arm being strengthened. Lower-right: another enlarged portion of the left image marked with green, where one can see the halo effect around the powder puff on the women's right hand.

4.2 Layering Structure of Points

Currently there are overall four layers in our implementation. The first, or the bottom layer is to serve as the background. Considering that a background color is often used to set the basic tone, we initially perform a *segmentation* process to partition a given image into regions of pixels with similar colors, then we locate the *brightest* point in each segment for using its color to color all the points falling within the same region.

The second layer, sitting on top of the first layer, is to add random but controlled variation, in terms of hues, to the previous points colored as backgrounds. The reason behind this randomness is apparent: it's natural that slight variation exists from stroke to stroke. The variation manifests itself in many ways, such as color, size, and shape, and we will discuss each of them in turn. In addition, to mimic the dense coverage observed from normal Seurat's paintings, this layer of points is independently added in by applying the Poisson disk method for another run.

We perform the operations of edge enhancements on the third layer. Through the use of *Canny edge detection* algorithm, we could identify essentially two kinds of edges: silhouette edge and crease edge. The distinction between the two is to be elaborated later. We then apply different edge enhancement schemes to strengthen these edges accordingly. Unlike what we did on the second layer, this enhancement process only replace the colors of the selected existing points from the second layer without generating new points.

The fourth layer, or the top layer is used to deal with the complementary colors. As mentioned before, complementary colors, commonly used in Seurat's paintings, serve as contrasts to consequently enhance the original colors. We selectively choose points from the second layer and substitute their original colors by their complementary

counterparts. As for how to find the complementary color of a given color will be explained shortly. And it could be inferred that the distribution of points in this layer should be sparser than previous layers.

4.3 Attributes of Points

As many classified Seurat's works into pointillism, it is self-evident that how to deal with points is of utmost importance.

Locations of Points. Among the many attributes of a point, such as position, color, size, and shape, position is no doubt one of the most important properties of a point. More concretely speaking, each point should be arranged in a proper location in the simulated painting so that the entire set of points as a whole are distributed uniformly with desired spacing among them. To have a simple and efficient solution, we adopt the idea of *Poisson disks* [8] to place points evenly within the entire image, while the pairwise distance between any two points is at least with some prespecified amount. Note here the size of a point in general is larger than a pixel, as we will associate a mask with each such point for simulating the real point strokes used during the painting process.

Colors of Points. Following the description of the four-layer structure of points, here comes the most important aspect of a point, i.e., the color. We first explain how color perturbation is realized on the second layer of points. The central idea is to perform the modulation in the $CIELAB$ domain, as the same amount of change in values leads to the same amount of alteration in visual perception in this color system. For a given color, represented as (L^*, a^*, b^*) in its $CIELAB$ representation, we define three thresholds, one for each of the three channels in $CIELAB$, respectively, to limit the range of variations, thus forming a 3D region of variation. Furthermore, such a variation distribution should have its peak centered at the original color (L^*, a^*, b^*) to be perturbed, therefore we enforce a *tri-normal* or *3D Gaussian* distribution on top of the 3D variation range. In practice, this tri-normal distribution could be implemented by the *Box-Muller* method [9].

Sizes of Points. To have a faithful simulation, we begin our study by making careful observations on Seurat's numerous paintings. In particular, we have to infer the point size used in his paintings in the real world, and calculate the corresponding point size on the image to be simulated and displayed on our screen with a desired resolution. This inference could be easily done by measuring the point sizes from the downsized reproductions of Seurat's paintings whose real size information is available. In our implementation, the side length of a point size is set to be 9 to 17 pixels.

Shapes of Points. Once the location and size of a point are settled, the next property comes to play is shape. For each point to have a natural look, like what a painter would do with his point strokes, each point should not always bear the same look. A naive approach is to manually scribble a dozen of "pre-defined" shapes for being the *selection pool*, while each time a shape randomly selected from this pool is randomly perturbed to form the final shape. Another way is to start with a solid

circle and then randomly and gradually grow from its boundary to "attach" more pixels. Currently we adopt the naive approach for its simplicity and acceptable results.

4.4 Edge Enhancement

To simulate the edge enhancement effect regularly presented in Seurat's paintings, we first apply Canny edge detection [10] to identify potential edges. Two kinds of edges are further distinguished as *silhouette edges* and *crease edges*. The former usually represents the edges along the boundary, thus being more salient, whereas the latter mainly refers to the internal edges. In practice, we differentiate them by the length of edges, although more sophisticated schemes or user assistance could be adopted for a more precise classification.

Silhouette Edges. Once an edge is classed as a silhouette edge, we proceed by strengthening it to generate the so called *halo effect* mentioned previously. To do this, we first need to determine the affected region, which is in turns determined by an associated radius, in terms of pixels. We could identify such regions by "rolling" a circle with a specific radius along all silhouette edges. Next, we need to further distinguish the *bright side* and *dark side* from an edge, as we will enforce different enhancing policies for them.

To enhance an edge, the basic idea is to enlarge the contrast; or more specifically, to make the bright side brighter and at the same time the dark side darker. We also hope such a contrast enhancement gradually decay as we move farther away from the corresponding edge. This is achieved by applying the following non-linear functions:

$$y = \frac{a}{2}[2(x - \frac{1}{2})]^{b} + 1 - \frac{a}{2} \tag{1}$$

to the dark side and

$$y = \frac{a}{2}[-2(x - \frac{1}{2})]^{b} + 1 + \frac{a}{2} \tag{2}$$

to the bright side, where x and y represent the distance (normalized to be between 0 and 1) from the current point to its corresponding edge, and the adjusting ratio, respectively. This ratio, defined to be the new lightness value divided by the old lightness value, governs how the lightness should be changed for edge enhancement. The two constants, a and b, are used to control the non-linear behavior and the blending with other non-enhanced areas, and typical values for them are 0.5 and 0.8, respectively.

Crease Edges. The enhancement of crease edges is relatively simple. We do not distinguish between the dark side and bright side of an crease edge, but instead just apply a non-linear contrast enhancement scheme as the following:

$$y = \frac{1}{2}[2(x - \frac{1}{2})]^{d} + \frac{1}{2} \tag{3}$$

where x and y represent the original lightness value and adjusted lightness value, respectively. The term d is again used to control the non-linearity. The affected

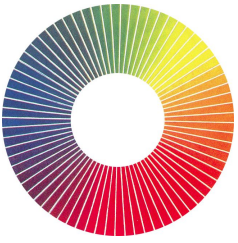


Fig. 3. The Chevreul's color circle

region, compared with that of the silhouette edge, is relatively small. A typical value of d is 0.5.

4.5 Complementary Colors and Color Dithering

As mentioned in the previous section, the existence of complementary colors is to enhance the original colors under contrast, thus being very important for numerous painting styles. One might surmise that finding the complementary color of a given color is trivial while it is not for at least the following two reasons. First, although modern color theory has enjoyed its success in many fields, it is in fact quite different from what traditional painters were using. For example, nowadays the color space of Hue, Saturation, and Value, or HSV color space for short, shows significant difference when compared with *Chevreul's color circle*, which is very likely the reference palette that Seurat adopted for his paintings. Therefore, it also indicates that to simulate the effect of complementary colors, instead of applying modern color theory such as the HSV model to find the color with the opposite hue, we should make use of Chevreul's color circle. Note that Chevreul's color circle, as shown in Figure 3, an extension of the eleven color model to contain 72 non-overlapping color ranges, is the fundamental color model that we use throughout for final color representation. Therefore a color, no matter coming from RGB or $CIELAB$, is converted to a representation by Chevreul's color circle before being put onto the simulated image. Such a conversion is done by first matching the color against all 72 color ranges, in terms of their *hue* values of the HSV model. Note that such a procedure is not only for finding a 72 color representation for a color, but also for locating the complementary color of a color in Chevreul's color circle. Also notice that complementary colors should appear with a relatively lower frequency so that they will not offset the original tones. And this could be achieved by selectively replacing a color in the second layer by its complementary counterpart, together with the checks to make sure such a color inversion will happen uniformly within an image.

However, as the gist of pointillism is not through color mixture but color juxtaposition, we thereby borrow the old technique of *halftoning* or *dithering* for simulating this effect. For example, assuming color a is to be approximated by a weighted average of color b and c, with the corresponding weighting factors to be 1 and 1, respectively. Instead of mixing the color before applying it, we approximate the original color with a 2×2 pattern by placing color b in two of the four regions while color c in the rest two regions.

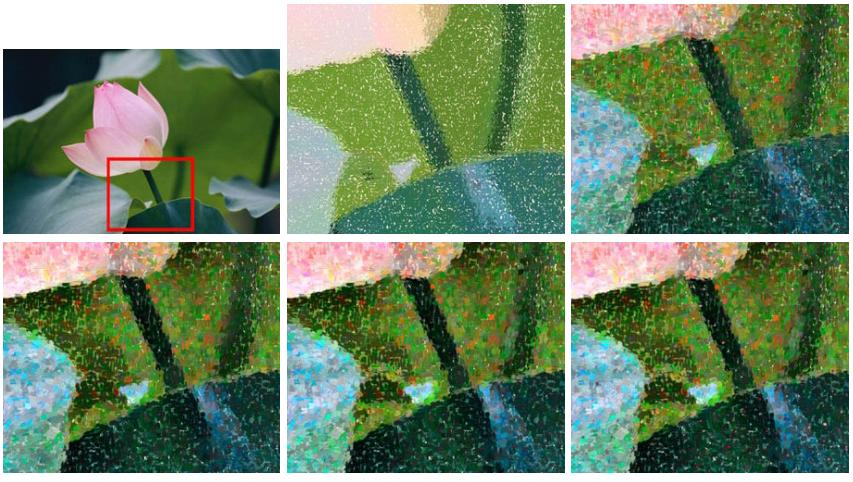


Fig. 4. From left to right and from top to bottom. 1. The source image of a pink flower. 2. The corresponding background layer image for an enlarged portion marked by red in the source image. 3. The previous image added with points of randomized colors. 4. The previous image added with silhouette edge enhancement. 5. The previous image added with crease edge enhancement. 6. The previous image added with complementary colors, thus the final image.

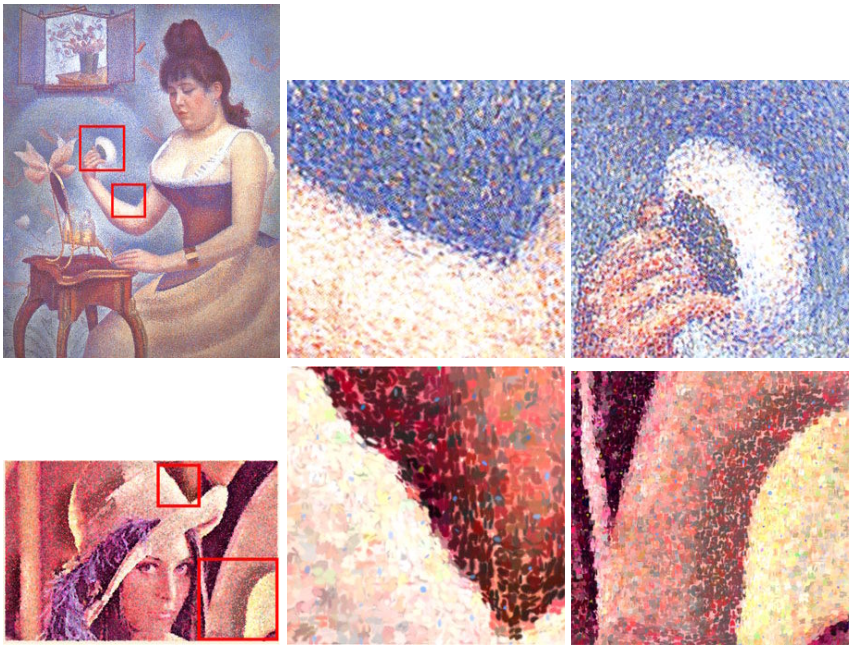


Fig. 5. From left to right and from top to bottom. 1. Seurat's painting. 2. An enlarged portion of the painting marked by red. 3. Another enlarged portion. 4. A simulated result of our system. 5. An enlarged portion of of our result marked by red. 6. Another enlarged portion.

5 Results

We have conducted our experiments on a Pentium IV 2.8GHz machine with 512MBytes memory running on a Windows XP operating system.

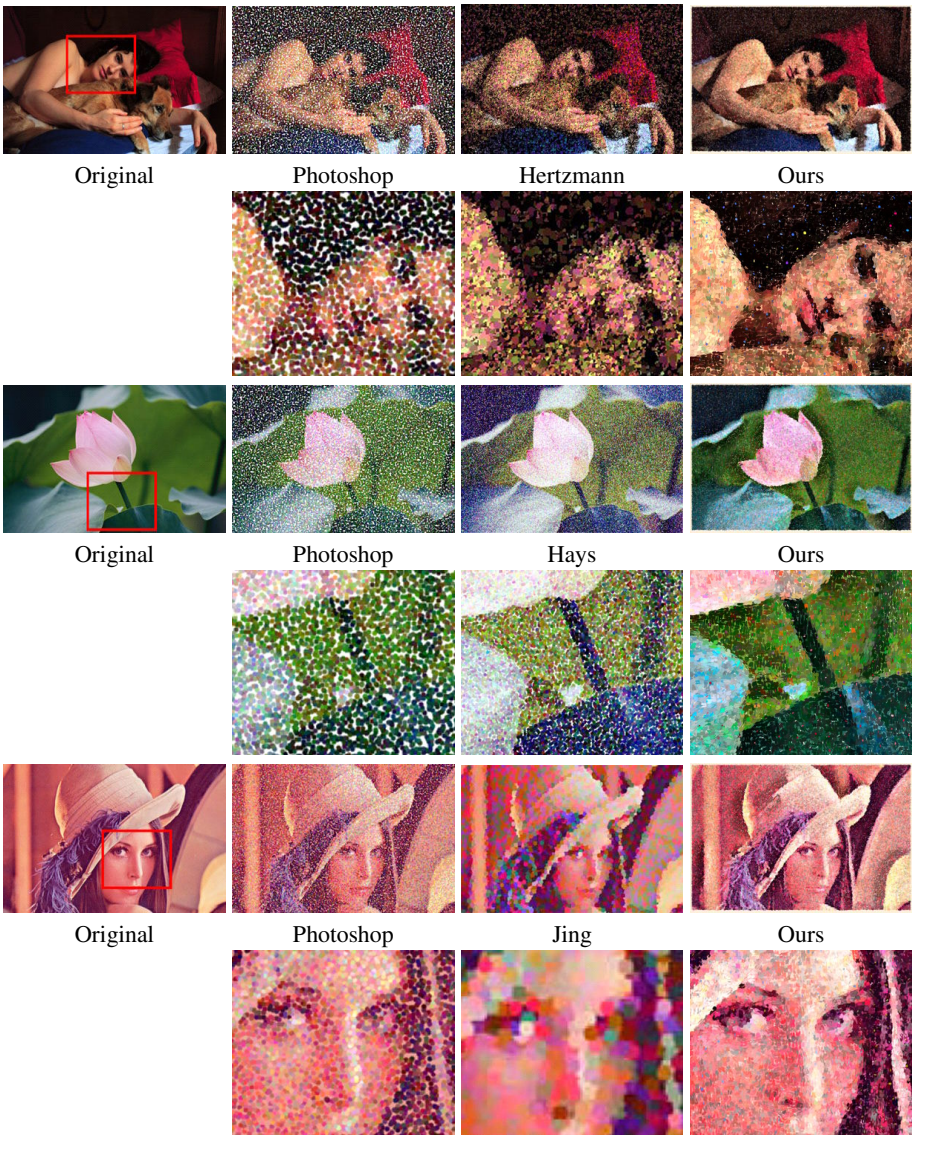


Fig. 6. Comparisons with Photoshop and three previous works. In each group, the original image is shown at the upper-left corner, and the rest of the images are divided into three columns, with each column demonstrating the results (and their corresponding enlarged portions) by using different approaches.

Figure 4 demonstrates our system through a "step-by-step" simulation of Seurat's painting styles, as described in the previous section.

Figure 5 compares the results of our system with one of Seurat's paintings, and as can be seen from this figure, our system does successfully simulate some important features generally presented in Seurat's work.

Finally, Figure 6 compare our results with the ones generated by Photoshop, Hertzmann et al. [2], Hays et al. [1], and Jing et al. [7]. Noticeable improvement on halo, complementary colors, as well as the overall appearance can be easily observed.

6 Conclusions and Future Work

We have implemented a system that could faithfully perform a non-photorealistic rendering of Seurat pointillism. Our success lies in simulating those features that we commonly observed from Seurat's paintings, such as halos and complementary colors, which have not yet been properly simulated by others. Results are demonstrated and compared with both Seurat's existing paintings and previous attempted simulations. In the future, we plan to automate our simulation process by reducing the involved parameter tuning as much as possible. We would also like to faithfully simulate the styles of other impressionism painters as well.

References

1. Hays, J., Essa, I.: Image and Video Based Painterly Animation. In: NPAR '2004. (2004) 113–120
2. Hertzmann, A.: Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In: SIGGRAPH '1998. (1998)
3. Luong, T., Seth, A., Klein, A., Lawrence, J.: Isoluminant Color Picking for Non-Photorealistic Rendering. In: Graphics Inteface '2005. (2005) 233–240
4. Litwinowicz, P.: Processing Images and Video for An Impressionist Effect. In: SIGGRAPH '97. (1997) 407–414
5. Winkenbach, G., Salesin, D.: Computer-Generated Pen-and-Ink Illustration. In: SIGGRAPH '1994. (1994) 91–100
6. Hertzmann, A.: Image Analogies. In: SIGGRAPH '2001. (2001) 327–340
7. Jing, L., Inoue, K., Urahama, K.: An npr technique for pointllistic and mosaic images with impressionist color arrangement. In: International Symposium on Visual Computing 2005. (2005) 1–8
8. Yellot, J., I, J.: Spectral consequences of photoreceptor sampling in the rhcsus retina. Science **221** (1983) 382–385
9. Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. The Annuals of Mathematical Statistics **29** (1958) 610–611
10. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698

Semantically Relevant Image Retrieval by Combining Image and Linguistic Analysis

Tony Lam and Rahul Singh

Department of Computer Science
San Francisco State University
San Francisco, CA94132
tonster@sfsu.edu, rsingh@cs.sfsu.edu

Abstract. In this paper, we introduce a novel approach to image-based information retrieval by combining image analysis with linguistic analysis of associated annotation information. While numerous Content Based Image Retrieval (CBIR) systems exist, most of them are constrained to use images as the only source of information. In contrast, recent research, especially in the area of web-search has also used techniques that rely purely on textual information associated with an image. The proposed research adopts a conceptually different philosophy. It utilizes the information at both the image and annotation level, if it detects a strong semantic coherence between them. Otherwise, depending on the quality of information available, either of the media is selected to execute the search. Semantic similarity is defined through the use of linguistic relationships in WordNet as well as through shape, texture, and color. Our investigations lead to results that are of significance in designing multimedia information retrieval systems. These include technical details on designing cross-media retrieval strategies as well as the conclusion that combining information modalities during retrieval not only leads to more semantically relevant performance but can also help capture highly complex issues such as the emergent semantics associated with images.

1 Introduction

With the rapid proliferation of image-based information, image retrieval systems are becoming increasingly significant and have many applications in domains such as multimedia information, personal information management, geographical information systems, and bio-medical imaging to name a few. Much research has been done in this area, especially ones geared towards designing better image analysis techniques for use in Content Based Image Retrieval (CBIR) systems. As newer and better image analyzing methods are developed, *the best image-based recognition algorithm have been found to provide only partial solution to the image retrieval problem.* This is because of the fact that the image retrieval problem is more complex than analyzing pixel-level information. In recent past, researches have attempted to take advantage of the non-visual information that may be associated with images. The most common of these are related to searching for images on the Web, where the search results are

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 770 – 779, 2006.
© Springer-Verlag Berlin Heidelberg 2006

typically based (at the state-of-the-art) on analysis of the textual information (such as the caption) associated with an image along with an analysis of the hyperlink structure of the page where the image is found.

Our goal is to develop retrieval strategies that go beyond examining the problem from the extremal viewpoints of image-only or text-only analysis. Beyond that, we want to understand what the query means and how meaningful the retrieved result is to the query. Since annotations are linguistic constructs and languages (in our case English) have a structure, we use WordNet [11], a psycholinguistic dictionary project developed at Princeton University. WordNet encodes a variety of semantic relationships inherent to the language. Thus, with its help, we can perform better text analysis and comparison that is based not only on keyword matching but also on the semantics of the image annotation.

In summary, we pose the retrieval problem not merely in terms of determining pixel-level similarity. This complementary formulation allows one modality to compensate for the deficiency of the other. Furthermore, the formulation also allows the retrieval strategies to be mutually supportive, especially in cases where one of them can provide results that are semantically more relevant than the other.

2 Previous Work

Numerous content-based image retrieval systems are available today; all perform excellent low level feature matching retrieval: color and texture [3][7][10][16]. Others use newer representation such as wavlets [18]. While many object recognition and region segmentation algorithms have developed from this problem space, these techniques are not perfect and will probably never be. Aside from the difficulty in developing perfect computer vision techniques, learning the semantic meaning of an image is an even harder problem.

Text-based image retrieval is more novel [14]. Many online search engines have expanded their WWW search to include subsystems made especially for image retrieval [8][9]. Most WWW images are associated with web pages and the textual content contained in them are very helpful in describing the images. Analyzing the text information as part of the image retrieval process can capture the semantic implication of the image from a linguistic level just as it can capture the same type of information from a web page. However, little research has been done to pursue capturing image semantics during the retrieval process [1][2][17].

Other text-based image retrieval systems, including meta-tagging the images, try to address the image semantic issue [19], but most are only able to address low level semantics of the image. The low level semantics of an image can be directly captured from low level feature analysis. But a more complex problem exists in capturing high level semantics, or the conceptual neighbors. Adding to the complexity is the assumption that all annotations are meaningful to the image. This brings us to the concept of emergent semantics. For example, one user may annotate a photograph of a tiger as a trip to a zoo, and another user may annotate the photograph as an animal in the jungle. Both annotations are valid given the proper user perspective and context. Combining

the content and textual based approaches to image retrieval will yield more meaningful and contextual results and allow us to search using the semantics of the image in addition to low level object searching.

One way to capture image semantics is to combine the two aforementioned approaches [4][5][12] and pursue image retrieval from a conceptual level instead of the traditional feature level that most abovementioned research has done. This way would guarantee to outperform either approach alone. The combined approach is similarly used in hierarchical clustering of WWW image search results [6], but our proposed technique is used in the actual retrieval phase with the intention to capture the semantics of the image.

3 Image Semantics and Philosophy of the Proposed Approach

In the context of information retrieval, one predominant difference between image and textual media is how the corresponding semantics may be discerned. While textual media have relatively straightforward interpretations where each word is a contributing part of the whole meaning, image semantics is fundamentally different. Image semantics, unlike the actual image, cannot be dissected into semantically relevant blocks. It does not always possess the compositional nature of image representation and therefore cannot be captured through analysis of the image features alone.

For images, we distinguish three different levels of semantics. *Low level semantics* is the information that can be retrieved through direct interpretation of the image annotation or through content object segmentation. For the purpose of many information goals, this level of interpretation is sufficient and no additional information about the image needs to be discovered. *High level semantics* expands upon the low level semantics of an image by examining the conceptual relationships that may exist. It thus goes beyond simple image interpretation. For instance, given the image of a tiger, based on high level semantics, the image can be related to those of felines or carnivores. While such interpretation is obvious, it is hard to achieve using image-based information alone. Finally, *emergent semantics* is perceptual and contextual [15]. This level of semantics is the most difficult semantic level to capture since it is highly subjective and differential. Since emergent semantics is nearly impossible to capture with any single mode of feature-based analysis, a better approach to capture the emergent semantics is to combine information from multiple semantically correlated media (whenever available). However, and we underline this, it is important to note that in such cases some or all of the media may be noisy, have errors, or be semantically non-correlated. For image-content based strategies, causes can be noise, poor image processing or low level (signal-level) features that do not necessarily correspond to semantics of the query. For textual strategies, poor performance may be due to the lack of relevant keywords in the annotation, poor keyword selection, or inadequate capture of the underlying linguistic relationships. Therefore, retrieval by combining information from multiple media is better than focusing on a specific media alone. In addition to dealing with the above issues, an important step in combining information from different media, such as in our approach, is to determine how the annotation and the image are semantically related.

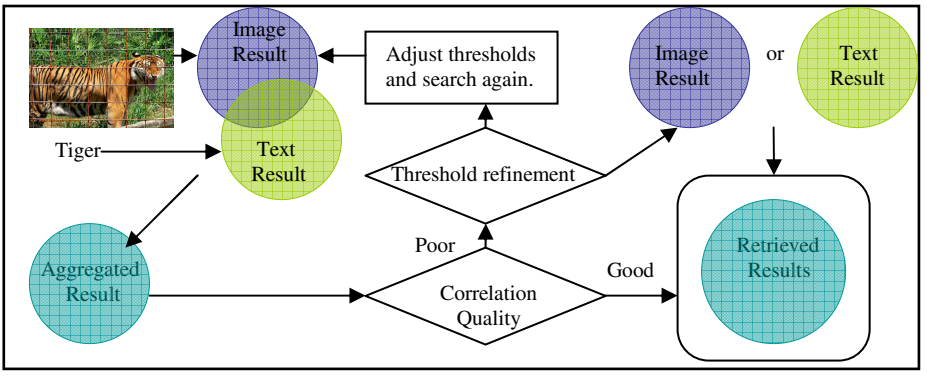


Fig. 1. Outline of the proposed approach: the correlation between the image and the annotation is analyzed. If the correlation is good, results are retrieved using information from both modalities. Otherwise the thresholds are iteratively adjusted. If the correlation continues to be poor, the more information rich modality is determined and used for retrieval.

The information at the image-level and the annotation-level may span one or more of the aforementioned semantic classes. Further, either of the modalities may suffer from any of the aforementioned sources of noise. To improve retrieval quality, we therefore need to understand the correlation between the annotation and the image and minimize the effects due to noise or imprecision in either of the modalities. Our solution philosophy is centered on the idea that if we find the annotation to image correlation to be good, the aggregated result from using both sources of information is returned. If the correlation is poor, the more semantically relevant information modality (annotation-based or image-based) is identified and used for retrieval.

4 Algorithm and System Development

Our approach of image retrieval combines text and image analysis to capture the semantics of the image collection. Image collections are added and annotated by the user. The annotation can range from a simple one word descriptor or a verbose paragraph. Retrieval is initialized by a text search, and a representative image is picked from result. system will search the database using both image features and text query and/or image annotation using the user-selected minimal similarity thresholds. Each analysis is discussed in greater detail in the following sub-sections.

4.1 Image Analysis

Semantically similar images seldom have high visual similarity due to different color and texture features. We adapted the image segmentation technique used in the Blobworld image retrieval system [3]. Blobworld segments an image using Expectation-Maximization (EM) algorithm based on color and texture.

In our system, an image is size-normalized and converted to the L*a*b* color space. Each pixel is then analyzed and clustered together to form regions with similar characteristics based on ten iterations of EM. Only regions with size of at least 2% of

the entire image are deemed significant and have their characteristics stored in the database. Each region's simplified color histogram, contrast, anisotropy, and size are stored as a feature vector in the database.

Upon choosing the image that will be used for image similarity comparison, the user selects the region(s) of interest. Each selected region is compared with regions of other images in the database. The feature vector for each region includes color, texture, and size. Each image comparison is indexed after the initial search for efficiency purposes. All images are preprocessed offline since processing time for each image is between three and five minutes.

We compute the image similarity score μ_i for the chosen image with selected regions r_i in the follow manner:

1. For each region r_j in the database image with feature vector v_j, Euclidean distance between v_i and v_j, and similarity between the two regions is calculated:

$$d_{ij}^2 = (v_i - v_j)^T \sum (v_i - v_j) \qquad \mu_{ij} = e^{-\frac{d_{ij}}{2}} \tag{1}$$

2. The score μ_i is:

$$\mu_i = \max_j \mu_{ij} \tag{2}$$

The matrix Σ is block diagonal with the block corresponding to the texture and size features being the identity matrix with diagonal values 20, 50 and 5. The block corresponding to the color feature is the matrix A for i and j corresponding to the color bin for the color histogram of the region:

$$A_{ij} = \{1.0 \ \textit{if} \ \ i = j, \ 0.5 \ \textit{if} \ \ d_{ij} < 1.8, \ 0.0 \ \textit{otherwise} \tag{3}$$

4.2 Text Analysis

Due to inherit subjectivity of information retrieval, low level keyword matching is not enough to completely capture the image semantics. We expand the text information associated with the image with psycholinguistic data available through WordNet and TFIDF to capture the semantics of the image annotation. TFIDF is popular text analysis technique that employs a term weighting scheme that adjusts the frequency weight of terms in a document by taking the product of the term frequency and the inverse document frequency (document frequency refers to the number of documents the term appears in).

We extended our prior research [20] for text analysis of the image annotations, including also synonym and holonym (x is a holonym of y if y is part of x) hierarchies in addition to hypernyms. This technique allows image semantics beyond the low level to emerge. WordNet stores each lexical unit as a synset (synonym set) that consists of terms that semantically mean the same thing. Each term may have several sense associations. For example, "tiger" can mean a big cat or a fierce person. For each term in the annotation, the follow is computed:

1. A TFIDF value is computed for each term where the IDF consists of a background set. We queried Google using random keywords and retrieved the top 50

documents for each query and built a term frequency matrix based on those results.

2. A hierarchy of lexical relationships including synonym, hypernym and holonyms. As the level of the hierarchy becomes more general, the weight assigned to the level decreases. The weight assigned to each term in the hierarchy is adjusted by the total length of the tree. For any term T let T_k denote the k_{th} term in the hierarchy. Also, let *depth(T)* denote the depth of the hierarchy for term T. The weight W_k for T_k is computed:

$$W_k = \frac{depth(T) - k}{depth(T)} \tag{4}$$

The similarity score between the query and the image annotation is computed from how closely the two hierarchies match. Comparisons are made with hierarchies of the same type (noun hypernym tree with noun hypernym) and the maximum score is assigned as the text score for the image. This score is then multiplied by the TFIDF value of term in the annotation. For queries with multiple keywords, the search is performed with each keyword having equal weight. The total score for each image is the sum of the text score for each keyword in the query. For a set of terms T, let T_i denote i_{th} term in T and T_{ij} denote j_{th} term in the lexical hierarchy for T_i. Let W denote the weight and n be the number of terms in the query. For query Q and each image annotation A in the database, the similarity score t_i between query Q_i and annotation A_k for matching Q_{ij} and A_{kl}, and the total text score t is:

$$t_i = \max(W_{Q_{ij}} \times W_{A_{kl}} \times tfidf_{A_k}) \qquad t = \frac{1}{n}\sum_{i=1}^{n} t_i \tag{5}$$

4.3 Determining Image-Annotation Correlation

After retrieving results based on text and image query, we proceed to analyze their correlation. We establish the correlation score by looking at the images in the intersection of the text search result and the image search result with total similarity score above user provided threshold which is preset at 0.7. Let T be the result set returned by text analysis and I be the result set returned by image analysis and S be the set of images in the intersection above the threshold. Each image result S_i has a text similarity score t_i and an image score μ_i . Let n be the cardinality of S, we compute the correlation score c by:

$$c = \frac{1}{n}\sum_{i=1}^{n} (t_i \times \mu_i) \tag{6}$$

If we determine that a good correlation ($c > 0.6$) exists between the text query and the image, we can simply return the intersection ranked by the total score, followed images with text or image scores above the corresponding text or image threshold. The minimum text and image thresholds are established by taking the min(t_i) and min(μ_i) from S. For poor correlation ($c \leq 0.6$), we need to determine which retrieval technique would yield more accurate overall results. We compute the text similarity

score $\bar{t}$ and image similarity score $\bar{\mu}$ by taking the mean of the respective similarity score of images in *S*.

We then adjust the search text and image search threshold with the text and image similarity scores by multiplying each threshold by the ratio of the average text and image similarity scores. Let *T(q,t)* denote the text results retrieved with the query *q* with minimum text threshold *t* and *I(p, μ)* denote the image results retrieved with image *p* with minimum image threshold μ . Also, let *S* denote the new aggregated result set after the threshold adjustments:

$$a = \frac{\bar{t}}{\bar{\mu}} \quad t^* = \begin{cases} \bar{t} & if\ a \geq 1 \\ \bar{t} \times a & otherwise \end{cases} \quad \mu^* = \begin{cases} \bar{\mu} & if\ a \leq 1 \\ \bar{\mu} \times a^{-1} & otherwise \end{cases} \tag{7}$$

$$S = T(q, t^*) \bigcap I(p, \mu^*) \tag{8}$$

If the overall correlation *c* does not improve beyond the preset minimum of 0.6 within three search iterations, the more relevant text or image only result is returned.

5 Experimental Evaluation

We conducted our experiments on a collection of 1564 images where each image is annotated with a brief description of what is in the image. Annotations varied in length, context and semantic level. The experiments investigate the efficacy of our solution philosophy, provide insights about the nature of the data, compare our approach with more-classical single media retrieval, and analyze parameters characterizing the retrieval performance.

To demonstrate how the system works we consider two queries with different levels of semantic complexity. The first of these is the query "tiger" with the similarity thresholds set at 0.50. Retrieval using text-only information and semantic relations from Wordnet is shown in Fig. 2 and can be observed to be highly relevant. However, an image of a woman is included in the top results. This stems from the fact that the image's annotation is "model" and the word "model," in the sense of a person, who is role model, is highly similar to "tiger," which also has the common root of "person" (fierce person). This example illustrates the fact that even with highly relevant annotation; text analysis alone cannot adequately capture the semantics of the query.

Our combined approach returned images that are much more coherent than either approach. The image of the model is demoted to a lower rank due to the lack of image similarity with the query image. Other images of tiger are promoted to the top to reflect the positive reinforcement of having the text and image scores combined. Those irrelevant images returned as top matches for the image result are filtered away because they have no textual similarity with the text. The calculated correlation in this search was 0.68. In this example, there is good correlation between the text query and the query image and our technique is able to capture the correct semantics of the query and retrieved images similar to the tiger semantics.

In the second example, we use the information goal "London vacation". The reader may note that semantics underlying this information goal are highly subjective and

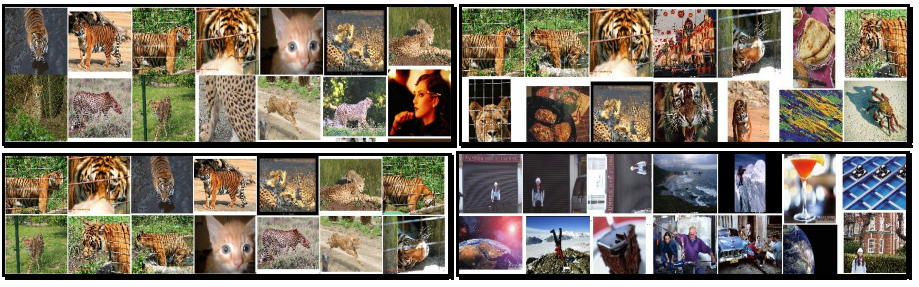


Fig. 2. *From top to bottom, left to right:* Top text result for the query "tiger" with text similarity threshold set at 0.60 (the image of the female "model" lower right), top image results for the image query with similarity threshold set at 0.75, top combined search result for the query "tiger" and the tiger image with overall threshold set at 0.70, and search results for the query "London vacation"

contextual. From the text result, an image of a person standing in from of a closed store is selected as representative because the person should be in most of the London vacation images being looked for. The result for this search is shown in Figure 2. The poor coherence in the result demonstrates the difficulty in capturing such semantics from the image. However, our approach has better recall than those obtained with text or image-based information alone. Additional experiments were conducted and data is summarized in Table 1 and Figure 3. Our combined approach has good precision performance with a high similarity threshold set at 0.90. As we relaxed the threshold requirement, there is a noticeable drop in precision for some queries such as "roses" and "vacation." This phenomenon is due to the highly contextual nature of those queries. Vacation is a concept that is difficult to capture with annotation and nearly impossible with image features; roses come in all different colors and they are not usually enough of a featured object in an image to have an explicit annotation. Since our approach analyzes result from text and image, our performance is dependant on the quality of those retrievals. In almost all cases, the quality of the combined result surpasses both techniques.

Table 1. Precision and recall for selected queries using text only, image only, and the proposed method at 0.70 and 0.90 similarity thresholds

	Query	Retrieval Technique			
		Text	Image	0.70	0.90
Precision	tiger	27%	31%	35%	50%
	wine	67%	10%	75%	75%
	cake	60%	8%	68%	100%
	vacation	100%	6%	22%	100%
	strawberry	65%	33%	83%	83%
	roses	17%	7%	15%	100%
Recall	tiger	30%	85%	85%	85%
	wine	23%	20%	10%	10%
	cake	88%	11%	88%	18%
	vacation	16%	16%	16%	5%
	strawberry	100%	40%	100%	100%
	roses	80%	20%	80%	80%

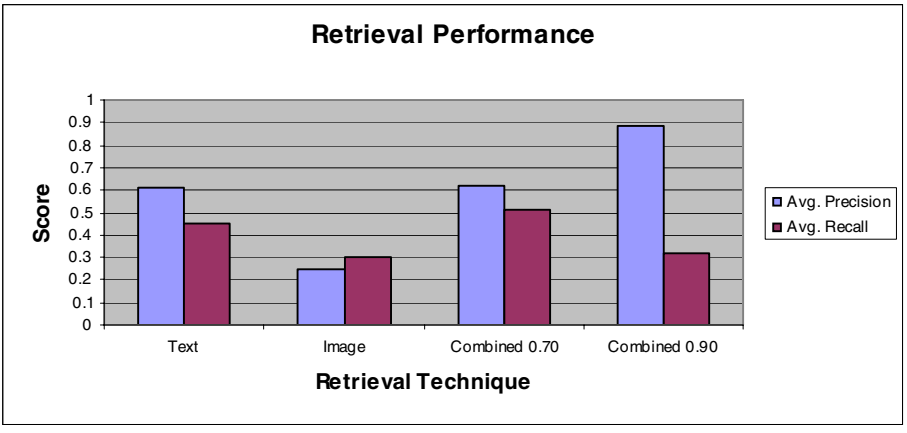


Fig. 3. Average retrieval performance for ten random queries using different retrieval strategies.Text and image similarity thresholds are set at 0.6 and 0.75 respectively. For our technique, the overall similarity thresholds are set at 0.7 and 0.9.

5 Conclusion

In this paper we address the problem of image retrieval by combining visual information along with the linguistic relationships that exists in the textual annotations associated with images. This formulation is distinct from pure image-based retrieval, where techniques have been stymied due to the signal-to-symbol gap. It is also distinct from pure text-based approaches, which are prone to textual ambiguities and by disregarding image-based information can not detect inconsistencies between the image-based and text-based descriptions, when they occur. The proposed approach combining these two perspectives and is able to retrieve information that is semantically more meaningful and robust, than what is possible using image-only or text-only strategies. Furthermore, the proposed approach is able to handle complex characteristics, such as emergent semantics, that are associated with images. While preliminary, our investigations provide intriguing evidence in favor of the proposed strategy for using multiple-media in problems of complex information retrieval.

References

1. Y. Aslandogan, C. Their, C. Yu, J. Zou and N. Rishe, “Using Semantic Contents and WordNet in Image Retrieval”, in Proceedings of *ACM SIGIR* Conference, Philadelphia, PA, July 1997.
2. K. Barnard and D. Forsyth, "Learning the Semantics of Words and Pictures", International Conference on Computer Vision, vol 2, pp. 408-415, 2001.
3. C. Carson, S. Belonge, H. Greenspan, and J. Malik, “Blobworld: Image segmentation using Expectation-Maximization and its application to image querying,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, SUB.

4. F. Chen, U. Gargi, L. Niles, and H. Schütze, "Multi-modal browsing of images in web documents," *Proc. SPIE Document Recognition and Retrieval*, 1999.
5. M. La Cascia, S. Sethi and S. Sclaroff, "Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web," *IEEE Workshop on Content-based Access of Image and Video Libraries*.
6. C. Deng, X. He, Z. Li, W. Ma and J. Wen, "Hierarchical Clustering of WWW Image Search Results Using Visual, Textual and Link Information", *In Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 952 – 959, 2004.
7. Y. Deng, B. Manjunath, C. Kenney, M. Moore and H. Shin, "An Efficient Color Representation for Image Retrieval", *IEEE Transactions on Image Processing*, Vol.10, No.1, pp.140-147, 2001.
8. Flickr, http://www.flickr.com/.
9. Google search engine, http://www.google.com/.
10. C. Jacobs, A. Finkelstein, D. Salesin, "Fast Multiresolution Image Querying", *in Proceedings of Computer Graphics, Annual Conference Series*, pp. 277-286, 1995.
11. G. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller, "Introduction to WordNet: An on-line lexical database", *International Journal of Lexicography*, Vol. 3, No. 4, pp. 235—312, 1990.
12. S. Paek, C. L. Sable, V. Hatzivassiloglou, A. Jaimes, B. H. Schiffman, S.-F. Chang and K. R. McKeown, "Integration of Visual and Text based Approaches for the Content Labeling and 21 Classification of Photographs," *ACM SIGIR'99 Workshop on Multimedia Indexing and Retrieval*, 1999.
13. K. Rodden, W. Basalaj, D. Sinclair, and K. R. Wood. "Does organisation by similarity assist image browsing?" *In Proceedings of Human Factors in Computing Systems*, 2001.
14. C. Sable and V. Hatzivassiloglou, "Text-based approaches for the categorization of images", *In Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries*, pp. 19-38, 1999.
15. S. Santini, A. Gupta and R. Jain, "Emergent Semantics Through Interaction in Image Databases", Knowledge and Data Engineering, Vol. 13, No. 3, pp. 337 351, 2001.
16. S. Sclaroff, L. Taycher and M. La Cascia, "ImageRover: A Content-Based Image Browser for the World Wide Web," *IEEE Workshop on Content-based Access of Image and Video Libraries*, TR97-005 06/97.
17. J. Wang, J. Li, G. Wiederhold, "Semantics-Sensitive Integrated Matching for Picture Libraries", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 9, pp. 947-963, 2001.
18. J. Wang, G. Wiederhold, O. Firschein, and S. Wei, "Content-based image indexing and searching using daubechies' wavelets," *International Journal of Digital Libraries*, Vol.1, No. 4, pp. 311-328, 1998.
19. K. Yee, K. Swearingen, K. Li and M. Heart, "Faceted Metadata for Image Search and Browsing", *In Proceedings of the Conference on Human Factors in Computing Systems*, pp. 401-408, 2003.
20. B. Zambrano, R. Singh and B. Bhattarai, "Using Linguistic Models for Image Retrieval", *Proc. International Symposium on Visual Computing, Lecture Notes in Computer Science, Springer Verlag*, 2005.

A New Algorithm for Solid Texture Synthesis

Jia-Wei Chiou and Chuan-Kai Yang

National Taiwan University of Science and Technology, Taipei, 106, Taiwan, ROC

Abstract. Despite the tremendous rendering power offered by modern GPUs, real-time and photo-realistic rendering is still often out of reach of traditional polygonal-based rendering. Thanks to the invention of texture mapping, a scene with a moderate number of triangles could be readily and vividly rendered by nowadays popular and inexpensive graphics cards. However, as a desired texture often comes with a very limited size, the technique of texture synthesis, i.e., synthesizing a larger texture from an originally smaller texture, has become a popular research topic in recent years. Numerous techniques have been proposed to successfully synthesizing 2D textures in terms of quality and performance. 3D or solid texture synthesis, on the other hand, remains relatively unexplored due to its higher complexity. There are several types of existing algorithms for solid texture synthesis, and among them, the outstanding work by Jagnow et al. [1] opens a new door for solid texture synthesis of discrete particles; however, their work did not address two important issues, thus leaving room for further improvement. First, without the help of stereology, users need to explicitly provide the 3D shapes of target particles for synthesis, and this is especially true when oftentimes only a 2D texture image is available for solid texture synthesis. Second, the locations and orientations of the 3D particles are resolved by a *simulated annealing* method, which is intrinsically a non-deterministic approach, and thus the optimality is not always guaranteed. To solve the shape problem, we propose a simple algorithm that applies the idea of *visual hulls* to approximate the shapes of 3D particles when only a 2D image is given; to solve the location and orientation problem, we design a deterministic algorithm that could place these desired 3D particles in space more properly. Most importantly, there is no need for user's intervention for both algorithms. We have successfully implemented the proposed algorithm and the experimental results are also presented for comparisons with previous results and also for the proof of our concepts.

1 Introduction

One frequent problem encountered in texture mapping is that the texture image source often comes with a very limited size. As a result, how to generate a larger texture from a given smaller texture, has become one of the most important problems in the field of *texture synthesis*.

Textures can be two-dimensional (2D), three-dimensional (3D), and four or even higher-dimensional. Despite the fact that there have been numerous researches successfully performing texture synthesis, in terms of quality and efficiency, most of the works concentrate on 2D texture synthesis, while 3D texture, or solid texture synthesis, receives relatively less attention. The scarcity of related papers is mainly attributed to the

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 780–789, 2006.
© Springer-Verlag Berlin Heidelberg 2006

much higher complexity involved in solid texture synthesis. Among the existing approaches for solid texture synthesis, the synthesis of discrete particles has attracted our attention, as it is an area that so far has been even less explored. The pioneering work in this specific direction, done by Jagnow et al. [1] especially caught our eyes for its great outcome, but it still leaves two important issues that have not been fully addressed. The first issue concerns the 3D shapes of target synthesized particles. According to the paper, these shapes could be derived by applying *stereology*, otherwise the provision of 3D particles is required. However, as often times only one 2D image is available for texture synthesis, the application of stereology may be difficult. Instead, we propose a simple algorithm that could approximately construct the shapes of desired 3D particles through the concept of *visual hull*, assuming the synthesized 3D particles are *iso-tropic*, i.e., bearing similar cross sections from every viewing direction. The second issue is regarding the placement of these 3D particles. In their paper, this issue is solved by a *simulated annealing* approach, where all the particles were initially put into the volume, then their locations or even orientations could be gradually adjusted to avoid collisions. Rather than using such a soft optimization technique, where the optimal solution can not always be guaranteed, we develop a simple algorithm that could deterministically and appropriately place the particles in the output texture volume.

The rest of the paper is organized as the following. Section 2 reviews some of the literature related to this study. Section 3 details how we synthesize solid texture of particles. Section 4 presents the experimental results produced by our system, while section 5 concludes our work and hints several potential future research directions.

2 Related Work

The concept of solid texture was first given by Gardner et al. [2], but the term *solid texture* was formally introduced by Peachy [3] and Perlin [4] in 1985. As pointed out in the introduction section, there have been numerous researches on texture synthesis. However, due to the involved complexity, only relatively few of them are on solid texture synthesis. In terms of completeness, the review done by Dischler et al. [5] does a good job, therefore we will only concentrate on the literature that is highly related to solid texture synthesis.

In terms of spectral analysis for solid texture synthesis, there exist several approaches, such as Ghazanfarpour et al. [6], [7], and Dischler et al. [8]. Our work, on the contrary, employs a spatial analysis approach. With regard to this direction, Chen et al. [9] introduced a 3D texture synthesis method that is based on texture turbulence and texture growing. Wei [10] proposed to synthesize solid texture from multiple 2D source. The basic idea follows exactly from his former paper [11], which in turn is an improved version of an earlier and similar paper by Heeger et. al [12]. Although both approaches operate in the spatial domain, they are not suitable for dealing with solid texture of particles, as the integrity of particles may get destroyed during the synthesis process.

To be able to synthesize solid texture of particles, Dischler et al. [13] proposed a method, operates in the spatial domain, to shape particles using the idea of *generalized cylinders*. Perhaps the most similar approach to ours is the one proposed by Jagnow

et al [1]. In the paper, they first showed how to perform an accurate estimation of 3D particle distribution from a 2D image, so that later particles of different sizes can be generated and arranged accordingly. Next they demonstrated how a *simulated annealing* approach is used to solve the location arrangement problem when all synthesized particles are to be put within the target texture volume. Finally they addressed the issue of adding variant colors and details on the synthesized particles. Like [1], our method also operates in the spatial domain and our target solid texture is for discrete particles, but unlike their approach to depend on *stereology* or on the provision of 3D particles for later synthesis, we try to build particles solely from the given 2D image(s). Moreover, our algorithm for particles' location arrangement is also quite different from theirs.

3 3D Volume Texture Synthesis

We present the core materials of this work in this section. To ease the discussion, we hereby distinguish two terms: *local volume*, and *global volume*. The first term refers to the crude volume from which a single particle is synthesized, while the second term the target volume of solid texture where all the generated particles to be placed into. Given one or more 2D images, our task is to synthesize the global volume of solid texture of discrete particles, where the cross sections of these discrete particles will look like what appear in the input 2D images. Let us start with an system overview which explains what the major steps are, and how these steps are linked together.

3.1 Overview

Figure 1 depicts an overview of our system. First, we start with the segmentation of the given one of more 2D images, so that the associated information of the target particles can be analyzed and collected for later synthesis use. Second, particles with too small sizes are filtered to get rid of possible noise presented in the input image(s). Third, according to the contours extracted from the previous stage, we could synthesize a single particle by applying the *visual hull* algorithm. Fourth, according to the information collected from the input image(s), we could scale the sizes of a particle non-uniformly along each dimension, assign the color of a particle, and arrange a spatial location for a particle, so that the statistics shown in 3D will match what we observed in the given 2D images. The execution of the third and the fourth steps forms a loop, that is, particles are generated one by one until the desired number of particles or the specified overall *crowdedness* is reached.

3.2 Segmentation of 2D Input Images

The segmentation process extracts a set of 2D shapes from a given one of more 2D images. Currently this process is done manually for the following reasons. First, some particles may possess textures which make segmentation more difficult. Second, for the sake of robustness in the *visual hull* process, particles whose sizes are too small should be eliminated. Third, some particles which intersect with the image boundary should also be removed, so that they will not create sharp boundary during the visual hull process.

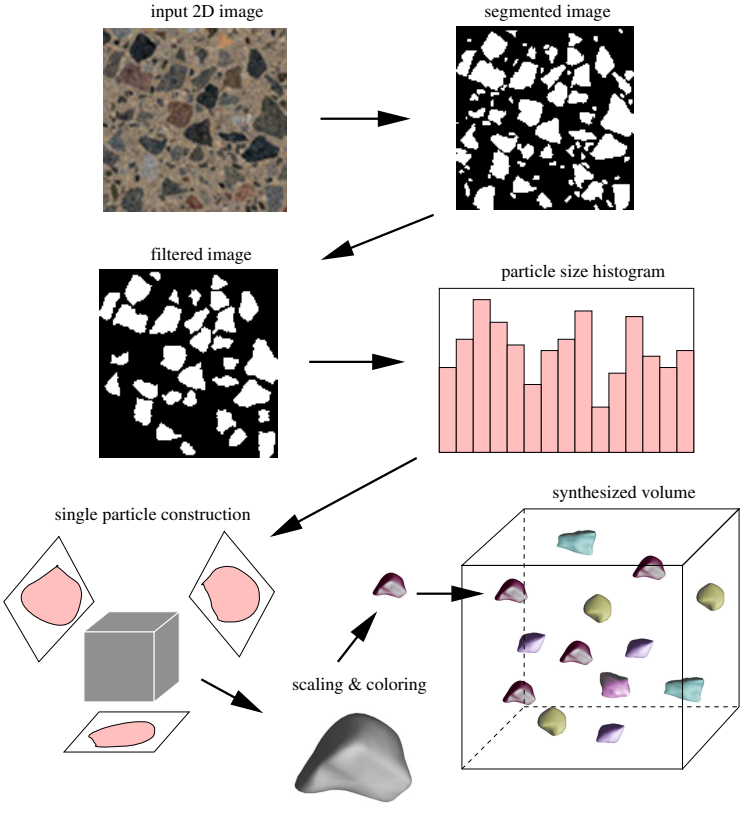


Fig. 1. An overview of the system used in this work

3.3 Histogram Analysis of Particle Attributes

The first information to be collected, after the previous segmentation stage, is the size of particles. We measure the height and width of the 2D bounding-box of a particle, and build the associated histogram. In addition, color information is collected in this stage as well. After the analysis of size and color information, the corresponding histograms, together with the 2D contours are stored so that the arrangement of locations and colors could be determined accordingly later on.

3.4 Synthesis of a Single Particle

The next step is to synthesize particles. We first show how to synthesize one particle, and we will explain how to adjust the sizes of a generated particle along different dimensions. To address this, Dischler et al. proposed in [13] to perform the synthesis of a particle through *generalized cylinders*, i.e., a base figure followed by a moving path. This technique, though proven to be effective in some cases, still shows orientation preference, and the modeling process itself is more complicated then our method to be

proposed. With only one image given, we choose to make a seemingly bold assumption that target particle shape is *isotropic*; that is, the shape does not differ much when viewed for different orientations. We later will show how this assumption may be modified when more than one 2D images are given and when anisotropic solid texture is desired. The assumption comes from the observation that if particles of different sizes and different orientations are to be placed randomly without collisions with others, then the 2D contours that we found from one slice simply represent cross sections of these particles along arbitrary directions. This seems to be reasonable as the shapes of particle's cross sections in general do not show too much variation. We therefore randomly *select a subset of the 2D contours collected in the previous segmentation stage and treat them as the cross sections of a single particle from different view angles*, then use the idea of *visual hulls* [14,15,16] to reconstruct the 3D shape of a particle.

Although simple, there is still one implementation issue of visual hulls that is worth mentioning. In order to produce a smooth particle, usually at least a dozen of cross sections are needed for projection, and the result is the intersection of all these projections. However, imagine there is one 2D contour used for projection that is significantly smaller than others, then the size of the resulting particle will be reduced rapidly due to intersection. This essentially means that it is more difficult to control the size of generated particle if the involved 2D contours present non-negligible variations in terms of width or height. We therefore choose to "decouple" this factor from the synthesis of a single particle. More specifically, after recording the original size or bounding-box information of all the 2D contours for future use, we identify the largest width and largest height from these 2D bounding-boxes. Assume the larger one of the two is L, we then stretch all the 2D bounding-boxes into *square images* with the side length L. We will refer to this stretching process as a *normalization* process.

After the determination of L, the construction of a single particle is as follows. We start with a cubic volume whose side length is L, and the voxels of this local volume are initialized to be the color that we desire, as will be described later, a dozen of randomly chosen *normalized* 2D contours are then used to "carve" the 3D shape of a target particle from an arbitrary direction.

Note that after the synthesis is done, the bounding-box of the synthesized particle is measured again for a more accurate estimation and for later processing, as the newly formed particle may have a quite different bounding-box dimension from that of the initial local volume, due to the effect of multiple cross-section projections.

3.5 Size Assignment

The aforementioned procedure can be used to generate a single particle. The next step is to scale it non-uniformly along each dimension so that the overall size distribution of all particles could match what we observe from the given 2D image(s). One way is to apply the idea described in [1], where the size and density of particles are analyzed in detail. Here we resort to a simpler approach which in practice performs satisfactorily well. Recall that we have previously built the histogram information from the width and height information of the bounding-boxes of the extracted 2D contours, so each bounding-box in effect "pairs up" its width and height values. To respect the original distribution, now the size picking process has become a *hierarchical picking process*. At the first level, we

randomly picked a $(width, height)$ pair from the histogram recorded previously, while each such pair is associated with a probability according to its frequency of appearance. The next step, happened at the second level, is to pick a $depth$ value from the chosen pair's associated distribution. Note that due to the bounding box pairing process, there may exist several sizes that have been paired with either the chosen width or height value, and from their occurring frequency the probability of each size being selected can then be determined, thus obtaining totally three values as a result. These three values are then used to scale the 3D particle constructed in the previous phase accordingly. Notice that there still exists the possibility that a newly generated particle with the chosen width, height and depth cannot be accommodated into the global volume due to the occupancy of particles produced earlier. To address this, we dynamically adjust the distribution so the probability of larger sizes will become smaller and smaller.

3.6 Color Assignment

The color assignment can be done similarly. Here we adopt the idea of Jagnow et al.'s idea in [1], where the average colors for 2D particles, as well as for the background pixels are first calculated. 2D particles and background pixels are then colored by their averaged colors. A residual image is formed by subtracting the averaged-color image from the original input image. During the color assignment process, pixel values on the residual image can then be used to perturb the previously assigned color of a background or non-background voxel.

3.7 Location Assignment

Once the size and color assignment of a particle are settled, the location assignment comes next. One possible solution for particle arrangement is to generate all particles at once, then place them within the global volume, just like Jagnow et al.'s approach [1]. There are, however, two drawbacks with their approach. First, sometimes it may not be easy to determine the number of particles to be placed into the global volume, given the fact that particles' sizes and orientations may vary significantly. Second and most importantly, as particles may have arbitrary shape, to find the best placement for all particles simultaneously is more difficult than the well known *bin packing* problem, which is already a *NP complete* problem. Therefore in [1], a *simulated annealing* approach is used instead to probe for a sub-optimal solution.

A potentially less optimized but more efficient approach is to generate and place a particle one by one, while at each time we always try to find the best spot for arranging a particle just produced. To expedite the process of searching for the best spot, we could associate each voxel with a value, which represents the maximal radius that a sphere centered at this voxel can have without touching any other non-background voxels. Essentially this value indicates the *spaciousness* around a given voxel. One way to speed up the calculation of spaciousness of each voxel is through the use of *Delauney tetrahedralization*, a 3D generalization of *Delauney triangulation*, which is closely related to *Voronoi diagram* [17,18,19].

After each voxel being assigned its spaciousness value, and according to this value, each voxel will be distributed to the corresponding queue of the same spaciousness. When a newly generated particle comes, we applied the policy of *best fit* borrowed from

operating systems, to locate the queue with the best spaciousness, and then randomly pick a voxel location from the queue, for arranging the given particle, i.e., the particle's center will be put at the selected spot. Once a particle is added, the Delauney tetrahedralization is incrementally computed and the spaciousness of those affected voxels will be updated, i.e., deleted from their original queues and inserted to different queues.

Due to the limitation of time, we did not get the chance to implement the non-trivial 3D Delauney tetrahedralization algorithm, but instead, we employ a more brute-force like approach as follows. Just like what we mentioned in the previous implementation, we could build a number of queues where each queue stores the voxels with the same spaciousness value. To save storage and speedup computation, we make use of the observation on the bounding-box size. From the size histogram collected during the segmentation stage, assume the largest possible side length of a 3D bounding-box is D, it is easy to show that the largest spaciousness value to consider is $\sqrt{3}D/2$. This basically means we only need to build queues of spaciousness up to $\sqrt{3}D/2$. Also due to this limited size, initially all the voxels, except the boundary voxels of the global volume, are assigned to the queue of $\sqrt{3}D/2$ spaciousness, while the boundary voxels to the queues corresponding to their nearest distances to the boundary of the global volume. By applying the same reasoning as for the Delauney tetrahedralization, we know that once a new particle is placed at the best spot with the aforementioned *best fit* algorithm, dealing with only its boundary voxels is sufficient. Therefore we perform the potential update only for those voxels that are lying within the distance of $\sqrt{3}D/2$ of the boundary voxels on the newly added particle. Although such implementation incurs a slightly higher overhead, given the fact that added number of particles may not be huge, and in practice the value of D, compared with final global volume side length, is relatively small, the overall performance is quite acceptable. Note that, although this approach does not measure the spaciousness as accurate as the Delauney tetrahedralization approach does, its exhaustive checking for updates along the boundary of the added particle has made its precision not far from the ground truth; therefore the results it generates is satisfactorily well in practice.

The final issue is the estimation of the crowdedness. One natural definition of the 3D global volume crowdedness is the number of non-background voxels divided by the number of all voxels of the global volume. To simulate the distribution on the given 2D image(s), the crowdedness of the 3D global volume should be equal to the 2D image crowdedness, which is similarly defined as the number of non-background pixels divided by the number of all pixels. To achieve the same level of crowdedness, the synthesis of the new particles should not stop until the desired crowdedness is reached.

4 Performance Results

In this section, we demonstrate the results using our proposed algorithm. All the tests are performed on a Pentium IV 3.0GHz machine with 1GBytes memory running on the Windows XP operating system. We have synthesized three solid texture volumes from three different squared input images, respectively. The three input images are shown side by side in Figure 2. In general, the overall execution time ranges from 30 minutes to 2 hours.

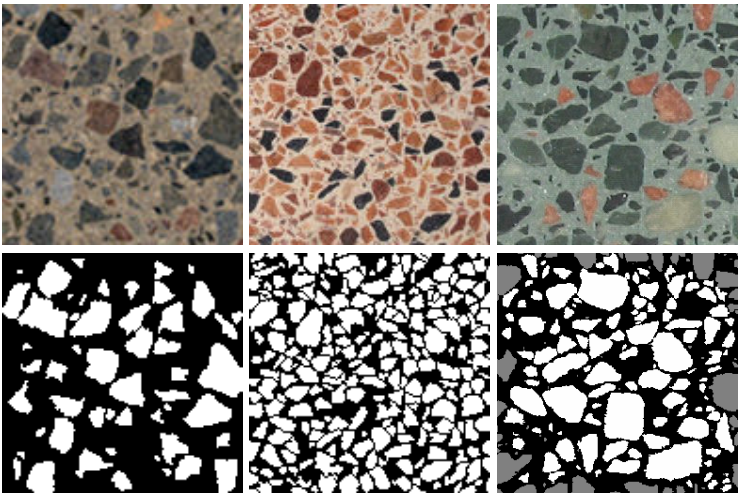


Fig. 2. Upper row: three input images used for the experiments, lower row: the corresponding 2D particle's masks extracted from the three input images, and used for later synthesis process. Note that the particles marked in gray in the third texture are not considered as they intersect with the image boundary.

Figure 3 shows the rendering results of two models with solid texture mapping, where the involved solid textures are synthesized from the first two input textures in Figure 2.

Figure 4 is the rendered image where compare multiple models using different solid textures generated using our system. Note that to facilitate rendering, we apply the transfer functions directly on the particles of synthesis, that is, the voxels of a particle get assigned both RGB colors and opacities immediately after the particle is generated. This is to ease the manipulation of transfer functions when different decisions have to be made on different particles. Alternatively, one could also design a more involved 3D interface to assign transfer functions to particles selectively.

5 Conclusion and Future Work

We propose a new algorithm for solid texture synthesis of particles. In this algorithm, each single particle could be constructed through the *visual hull* approach, and the locations of the generated particles are determined by employing a method that in spirit is similar to *Delauney tetrahedralization*. Performance results are shown to demonstrate the feasibility of our algorithm.

There are, however, still some limitations in our system, and how to go beyond these limitations deserves further study. First, our current assumption is that particle should be more or less *isotropic*, otherwise a *visual hull* approach for constructing the shape of a particle may fail. Remedies for this could be either requiring users to provide more 2D images corresponding to different viewing directions so that projections and involved images are carried out in pairs, or investigating more thoroughly on the given 2D



Fig. 3. Top two rows: a horse model rendered from the solid texture corresponding to the first input texture. Bottom two row: a bunny model rendered from the solid texture corresponding to the second input texture.



Fig. 4. Four solid-textured models using the texture volumes generated by our system

images for discovering the anisotropic patterns so that more intelligent projections could be performed. Second, it is evident that particles with concavity may not be modeled properly. One solution to this is to perform some deformation process, like what described in [13], and another possibility is to design a convenient interface so that users who are not satisfied with resulting shape synthesized by our system could easily perform their desired modification.

References

1. Jagnow, R., Dorsey, J., Rushmeier, H.: Stereological Techniques for Solid Textures. In: SIGGRAPH '2004. (2004) 329–335
2. Gardner, G.: Simulation of Natural Scene Using Textured Quadric Surfaces. In: SIGGRAPH '1984. (1984) 11–20
3. Peachey, D.: Solid Texturing on Complex Surfaces. In: SIGGRAPH '1985. (1985) 279–286
4. Perlin, K.: An Image Synthesizer. In: SIGGRAPH '1985. (1985) 287–296
5. Dischler, J., Ghazanfarpour, D.: A Survey of 3d Texturing. Computers & Graphics **25** (2001) 135–151
6. Ghazanfarpour, D., Dischler, J.: Spectral Analysis for Automatic 3-d Texture Generation. Computers & Graphics **19** (1995) 413–422
7. Ghazanfarpour, D., Dischler, J.: Generation of 3d Texture Using Multiple 2d Models Analysis. Computer Graphics Forum **15** (1996) 311–323
8. Dischler, J., Ghazanfarpour, D., Freydier, R.: Anisotropic Solid Texture Synthesis Using Orthogonal 2d Views. Computer Graphics Forum **17** (1998) 87–96
9. Chen, Y., Ip, H.H.: Texture Evolution: 3d Texture Synthesis from Single 2d Growable Texture Pattern. The Visual Computer **20** (2004) 650–664
10. Wei, L.: Texture Synthesis from Multiple Sources. In: SIGGRAPH 2003 Sketches & Applications. (2003)
11. Wei, L., Levoy, M.: Fast Texture Synthesis Using Tree-Structured Vector Quantization. In: SIGGRAPH '2000. (2000) 479–488
12. Heeger, D.J., Bergen, J.R.: Pyramid-Based Texture Analysis/Synthesis. In: SIGGRAPH '1995. (1995) 229–238
13. Dischler, J., Ghazanfarpour, D.: Interactive Image-Based Modeling of Macrostructured Textures. Computers & Graphics **19** (1999) 66–74
14. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 150–162
15. Petitjean, S.: A Computational Geometric Approach to Visual Hulls. International Journal of Computational Geometry & Applications **8** (1998) 407–436
16. Laurentini, A.: The Visual Hull of Curved Objects. In: ICCV '1999. (1999) 356–361
17. Voronoi, G.M.: Nouvelles Applications des Paramètres Continus à la Théorie des Formes Quadratiques. Premier Mémoire: Sur Quelques Propriétés des Formes Quadratiques Postives Parfaites. J. Reine Angew Math. **133** (1907) 97–178
18. Voronoi, G.M.: Nouvelles Applications des Paramètres Continus à la Théorie des Formes Quadratiques. Deuxième Mémoire: Recherches sur les Paralléllòèdres Primitifs. J. Reine Angew Math. **134** (1908) 198–287
19. Berg, M.D., Kreveld, M.V., Overmars, M., Schwarzkopf, O.: Computational Geometry, Algorithms and Applications. second edn. Springer (2000)

What Can We Learn from Biological Vision Studies for Human Motion Segmentation?

Cheng Chen and Guoliang Fan

School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK 74078

Abstract. We review recent biological vision studies that are related to *human motion segmentation*. Our goal is to develop a practically plausible computational framework that is guided by recent cognitive and psychological studies on the human visual system for the segmentation of human body in a video sequence. Specifically, we discuss the roles and interactions of bottom-up and top-down processes in visual perception processing as well as how to combine them synergistically in one computational model to guide human motion segmentation. We also examine recent research on biological movement perception, such as neural mechanisms and functionalities for biological movement recognition and two major psychological tracking theories. We attempt to develop a comprehensive computational model that involves both bottom-up and top-down processing and is deeply inspired by biological motion perception. According to this model, object segmentation, motion estimation, and action recognition are results of recurrent feedforward (bottom-up) and feedback (top-down) processes. Some open technical questions are also raised and discussed for future research.

1 Introduction

The goal of semantic video segmentation is to detect, segment, and identify semantically meaningful objects in a cluttered scene, such as a car or a pedestrian. Semantic video segmentation, especially *human motion segmentation*, has many important applications, such as visual surveillance, human activity analysis, video coding and communication, etc. Particularly, the focus of this research, human motion segmentation, involves three interacting elements, i.e., *object segmentation*, *motion estimation*, and *action recognition*. While humans can effortlessly recognize complex biological movements and segment the object of interest out of a cluttered scene, semantic video segmentation, particularly human motion segmentation, remains one of the most challenging problems as well as one of the most under-achieved research topics in the field of computer vision. Since reliable detection and recognition of complex biological movements and actions are a crucial ability for many species, perception of biological motion has been extensively studied by psychologists and neurophysiologists [1, 2, 3, 4]. We believe that hints and inspirations for reliable semantic video segmentation, especially human motion segmentation, lie in examining the processes used in many successful biological vision systems.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 790–801, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Biological vision studies are still at their infancy stage. Current research about biological vision usually concentrates on only certain aspect, e.g., attention, motion perception, visual memory, low-level perception organization. Although we know that the human visual system (HVS) is a well organized system, different perception rules need to cooperate together, very little effort has been made to develop a comprehensive computational model for HVS. This fact makes it difficult to apply perception principles systematically in practice. Still several well-understood perception principles have been adopted in the development of computer vision algorithms, e.g., semantic video segmentation. In this work, we will review two types of biological vision studies, i.e., general perception and biological movement perception, based on which we attempt to develop a comprehensive computational model for human motion segmentation. This work allows us to gain more insights to this challenging problem. Under the proposed framework, several open questions are raised and discussed for future research. Some of them have already started to attract researchers' attention recently.

2 General Perception Principles

Since the early 20^{th} century, psychologists have found a set of rules that govern the HVS. In this section, we will first introduce several important perception principles, which are closely related to the bottom-up process for video segmentation. We then develop a computational model and deduce some guidelines for feature selection and classifier design. We will also discuss the top-down process in the HVS that involves high-level knowledge or prior information for visual inferencing. State-of-the-art video segmentation algorithms have been inspired and motivated by these vision studies to some extent.

2.1 Joint Spatial-Temporal Grouping Theory

Cognitive scientistic research shows that spatial and temporal groupings are jointly involved in the HVS [5, 6, 7, 8]. In other words, human vision recognizes salient objects in space and time simultaneously. Specifically, spatial grouping is a process that merges spatial samples to form more complex visual entities, e.g., objects. Temporal grouping is a process where visual entities are linked over time. The successive visual entities that undergo a series of spatiotemporal groupings are called *matching units* or *correspondence tokens* by Ullman [9], which are 3D volumes in space and time.

2.2 Perception Organization Theory

The study about perception organization can be traced back to the Gestalt school of psychology in the early 20^{th} century, and expanded by Marr [10], Palmer and Rock [11], Palmer [12], Kubovy and Gepshtein [5, 7]. According to perception organization theory, visual perception is as a result of complex cascade part–whole hierarchy organization of visual information that involves both low-level and middle-level vision [13], as discussed in the following.

Low-Level Vision. Visual information is first sampled by small receptors called photoreceptor neurons, and then their outputs are grouped into small intermediate elements (closed regions of homogeneous properties, e.g., intensity, texture, color, motion), which are called as *Uniform Connectedness (UC) regions* by Palmer and Rock in [11]. This process is generally called "low-level vision". The first stage of low-level processing mainly depends on local visual properties [12]. It involves local computations on adjacent areas [14]. It is also highly parallel over space. The retina and primary visual cortex seem wired up to perform this kind of heavy computation.

Middle-Level Vision. According to Palmer and Rock's theory [11], once these intermediate elements (or UC regions) have been constructed, they are submitted to the so-called figure-ground process that is one of middle level vision processes. This figure-ground process that aims at foreground and background separation may be influenced by the feedbacks from a later process, especially by "common fate" grouping [15]. The elements identified as figures can be either parsed into subordinate units or further grouped into more complex visual entities, by the action of at least some of the classical Gestalt laws such as: common fate, proximity, closure, similarity. At the end, semantic "understanding" of a scene can be achieved by high-level vision that will be introduced later.

2.3 Motion Perception Theory

Substantial evidences in biological vision systems show that the presence of motion makes object detection, segmentation, and recognition easier, since motion cues can provide critical information for visual perception.

Short-Range and Long-Range Motion. There are several different motion perception theories [16]. A major theoretical framework of motion perception is the *short-range* and *long-range* dual process theory, which was first proposed by Braddick [17]. It was suggested that motion perception is mediated by a *short-range* process and a *long-range* process. The short-range process is low-level vision, occurring at the early stage of a visual system and combining motion information over a relatively small spatial and temporal range. It occurs within brief temporal intervals and small spatial neighborhoods [18]. The long-range process is a higher-level visual process, which operates over wide distances and long durations. Usually, short-range motion information can serve as the inputs to long-range motion processing.

Common Fate Theory. Another important finding about motion perception is the *common fate theory*: the elements that move together are grouped together. According to Gestalt psychology, common fate motion is a critical and robust source of visual information for dynamic object segmentation. The common fate theory can serve as the theoretical foundation for trajectory similarity-based merging to define moving objects in multiple object tracking [19].

2.4 Bottom-Up Processing in Visual Perception

State-of-the-art computer vision research has been deeply inspired by many perception principles of the HVS. We believe that semantic video segmentation would benefit from a comprehensive understanding about visual perception. Marr proposed the bottom-up sequential perception processing model in [10]. In Fig. 1, we illustrate a bottom-up visual perception model by summarizing Palmer and Rock's discussion on the relationship among several perception processes [11] and by combining Braddick's motion study [17] as well as Kubovy's joint spatiotemporal grouping theory [5].

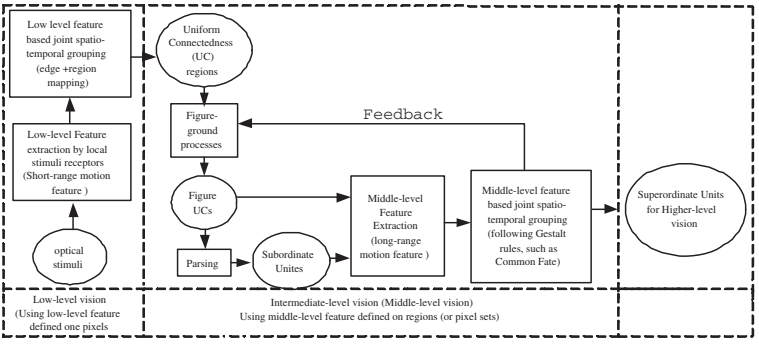


Fig. 1. The bottom-up visual perception model

As shown above, optical stimuli are first sampled by small receptors: photoreceptor neurons. Low-level visual cues (including short-range motion cues) are extracted by local stimulus receptors. Then, they are first grouped into small UC regions by joint spatiotemporal grouping, where both region and edge mapping could be involved. Once UC regions are constructed, they are submitted to the middle-level vision, i.e., the figure-ground process, where long-range motion cues should be involved. This process may be further affected by the feedback from later grouping processes, especially by "common fate" guided grouping [15]. These elements identified as figures can be either parsed into subordinate units or further grouped into more complex visual entities. These grouping processes may follow the rules of the classical Gestalt laws, e.g., common fate, proximity, closure [11].

2.5 Combined Bottom-Up and Top-Down Processing

The bottom-up sequential perception processing model has wide influence in the community of computer vision. But when this feed-forward model is used to process a noisy and cluttered scene, it usually fails to identify objects that can be easily recognized by human [20]. Bullier thought that the failure of bottom-up feed-forward model is due to the separation between the high-level prior

information and the local bottom-up segmentation [21]. More and more evidence from neuroscience and psychology show that the top-down modulation is essential and indispensable in visual perception.

In contrast to low-level vision, which is concerned with feature extraction and representation, the top-down modulation in high-level vision is primarily concerned with the interpretation and use of prior knowledge and information in a scene. High-level visual processes are performed on a selected portion of the image rather than uniformly across the entire scene, and they often depend upon the goal of the computation and prior knowledge related to specific objects [22]. Recently, psychologists modeled the top-down processing in high-level vision as statistical inference [23, 24, 25]. According to some recent neurophysiological evidences, and inspired by the most successful computer vision algorithms, Lee and Mumford suggested that the interactive bottom-up and top-down computations in visual neurons might be modelled by the mechanisms of particle filtering and Bayesian-belief propagation algorithms [26]. Combining the work of Lee and Mumford in [26] with these classic bottom-up sequential perception theories aforementioned, we present a computational model for visual perception in Fig. 2, where bottom-up and top-down processes are combined together.

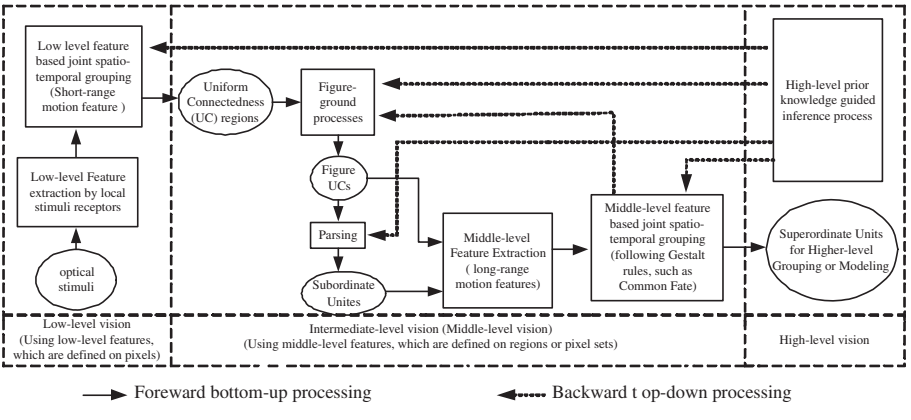


Fig. 2. Computational model of combined bottom-up and top-down processing

Within the framework of particle filtering and Bayesian-belief propagation algorithms, bottom-up processing generates data-driven hypotheses and top-down processing provide priors to reshape the probabilistic posterior distribution of various hypotheses. In this model, top-down processing could begin as early as the figure-ground process stage [27]. Specifically, for motion perception of human body, the prior knowledge and information about human body appearance pattern and motion pattern provide prediction and top-down prior, which play an important role in the interpretation and segmentation of human motion.

2.6 Application to Visual Segmentation

In the following, we will briefly review the recent research on visual segmentation, including image and video segmentation, which has been motivated and inspired by general perception principles.

Joint Spatiotemporal Approaches. Many video segmentation algorithms can be grouped into the catalog of joint spatiotemporal principle, such as the Normalize cuts graph partitioning method presented by Shi and Milk [28, 29], the mean shift method proposed by DeMenth [30], the Gaussian Mixture Model method proposed by Greenspan, et al [31]. A good survey of joint spatiotemporal grouping techniques for video segmentation can be found in [32]. According to the perception organization theory, low-level feature based segmentation is just at the beginning stage, i.e., "low-level vision". Therefore, it is not a surprise that the results of single-stage spatiotemporal segmentation algorithms are still far away from "semantic video segmentation" in a general scene.

Multi-layer Bottom-Up Approaches. The multi-layer framework is an effective approach for video segmentation. In [33,34,35,36], pixels are first grouped into small homogeneous regions in each frame. Then, segmented 2D regions are merged (or tracked) into 3D volumes in space and time. Recently, some multi-layer algorithms construct 3D space-time regions at the first stage. They match well with both the joint spatiotemporal principle and the hierarchical perception organization theory. In [37], a region growing method was proposed to construct the smallest homogeneous 3D blobs at the first stage, then, new features such as boundary, trajectory and motion of these blobs are extracted. Based on the extracted new features, these over-segmented small blobs can be further grouped into more advanced structures. The 3D watershed method was proposed in [38] to generate 3D blobs, and these blobs are then merged into more semantically meaningful objects. Multi-layer segmentation algorithms can use relevant visual features in different layers to achieve the final segmentation progressively. However, all multi-layer algorithms face some problems when there is a cluttered scene or objects have complicated visual properties and behaviors. Another challenge is how to decide when the cascaded merging processes should stop. Possible solutions to these problems may be provided by top-down processing.

Combined Bottom-Up and Top-Down Approaches. Although the segmentation scheme that combines both bottom-up and top-down processing is commonly advocated in the computer vision community [39], there has not yet been a widely accepted computational framework to achieve that goal. In [40], Borenstein et al. proposed an example of how to combine both bottom-up and top-down approaches into a single figure-ground image segmentation process. The unified segmentation, detection, and recognition framework proposed in [41] might be one of the most successful examples of applying this scheme in image segmentation. How to apply the similar idea to video segmentation remains to be a very challenging research topic.

3 Biological Movement Perception

Until now, little work has been done in combining bottom-up and top-down processes for semantic *video* segmentation, even though we know that it is one of the most promising directions. Since we are interested in human motion segmentation, we will review several neurophysiological and physiological literatures on biological movement perception, particularly human motion perception. Through this study, we try to find some inspiring hints and general guidelines to develop a practically plausible computational model to guide our future research.

3.1 Neural Mechanisms of Biological Movement Perception

Two-Pathways Vision Perception. Physiological studies found that there are two separate functional streams involved in vision perception: a ventral stream for the analysis of form (the "what" stream) and a dorsal stream for the analysis of position and motion (the "where" stream) [42, 43]. This discovery is called one of the major breakthroughs of the last few decades of vision research [44]. Inspired by this discovery, Giese and Poggio proposed a neural model for biological movement recognition [45]. This model has two separated parallel hierarchical pathways: *form pathway* and *motion pathway*, which are specialized for the analysis of form and motion information respectively. This model has a feed-forward architecture. The form and motion pathways consist of hierarchies of neural detectors that are connected unidirectionally in a bottom-up fashion.

Interaction and Convergence of Two Pathways. Further physiological and neuropsychological studies show that the two processing streams interact at several levels. Oram and Perrett found evidence for the convergence of two pathways in the anterior part of the superior temporal polysensory area (STPa) of the macaque monkey [46]. The integration of two separate aspects of information about a single object has been referred to as the binding problem. In [47], Sajda and Baek describe a probabilistic approach for the binding problem, which uses a generative network model to integrate both form and motion cues using the mechanism of belief propagation and Bayesian inference. The discovery of the convergence of two pathways in higher-level vision neural region is very important because the interaction of the two pathways may be realized by a feedback from the convergence place. A similar mechanism is used to explain the exchange of information between two distant neuron regions, where direct information exchange is inefficient or difficult [48].

3.2 Visual Tracking Theory

In computer vision, visual tracking is a common method for human motion and pose estimation that can provide prior information for the next frame. This strategy is well supported by a psychological theory called "object specific preview benefits" (OSPB) [49], which states *that the detection of a dynamic object's features are speeded when an earlier preview of those features occurs on*

the same object, ... Although the nature of visual tracking has not been systematically studied, there are two main theories for Multiple Object Tracking (MOT): Pylyshyn's visual index theory [50] and Kahneman's "object file" theory [49]. The visual index theory matches the construction process of 3D UC blobs very well. It is a low-level automated vision process, and no attention is needed. In contrast with the visual index theory, the "object file" theory suggests that the effortful attention is needed for a successful tracking process. An object file is a middle-level visual representation stored in the shot-term working memory, which collects spatiotemporal properties of the tracked objects, and the content of object files will be updated when the sensory situation changes.

3.3 A Comprehensive Computational Model

Based on previous studies and discussions, we hereby propose a comprehensive computational model for human motion segmentation, as shown in Figure 3. This computational model is deeply inspired by previously reviewed perception principles and biological movement perception. Related to the hierarchical neural model proposed in [45], the top flow is the bottom-up hierarchical form pathway; the bottom flow is the hierarchical motion pathway. This model is developed from the the general computational model as shown in Figure 2. Moreover, the bottom-up processes begin from the input visual stimuli (pixels). The output of form pathway is the appearance pattern (object segmentation), and that of motion pathway is the motion pattern (motion estimation). Action recognition is achieved by the integration of both pathway outputs.

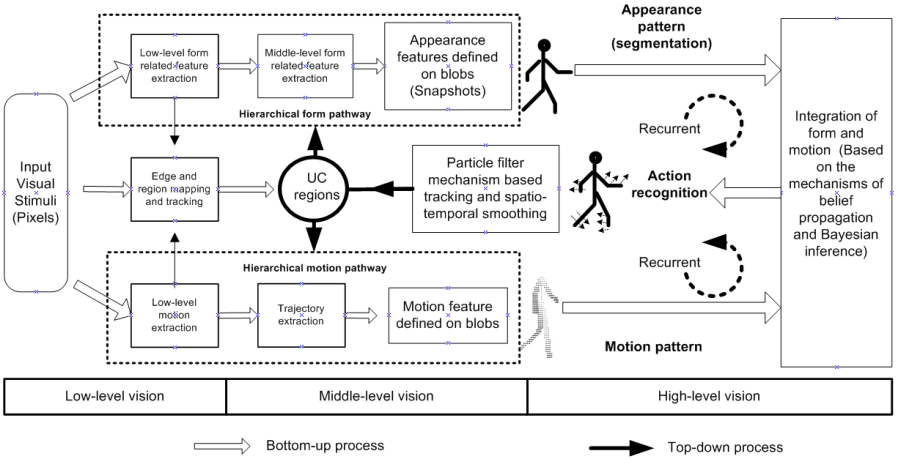


Fig. 3. A comprehensive computational model of human motion segmentation

For human motion perception, top-down processing begins from stored (or learned) prior knowledge about the human appearance and motion patterns, and then combines with the outputs of bottom-up processing to recognize human

action via inference [47]. The recognized human action (appearance+motion) is used as prior for tracking. UC region construction is the entry-level unit for the part-whole hierarchy, and we argue that the top-down inference should not go beyond the UC region. In other words, the UC region is the fundamental unit for top-down tracking. In the proposed model, Kahneman's "object file" theory in [49] can explain visual tracking after UC regions are generated. This is also supported by Kahneman's suggestion that "visual index" might be the initial phase of a sample object file. Therefore, in the proposed model, the two tracking theories can be integrated at different levels and in a serial flow.

4 Discussions and Conclusions

The model in Fig. 3 could help us understand some difficult problems in human motion segmentation. For example, in a bottom-up framework, motion estimation and object segmentation are often considered as a chicken-egg problem. Here object segmentation, motion estimation, and action recognition are results of recurrent and interwound feedforward/feedback processes. Also, model order estimation in bottom-up processing can be better understood under this framework. According to [22]: *the top-down process tells the bottom-up process where to stop.* As we mentioned earlier, particle filtering algorithms can help physiologists better understand human perception. In turn, studies of biological vision may help us to improve our computer vision algorithms. Guided by the model as shown in Fig. 3, we also raise the following open questions for future research.

- **How to combine both appearance and motion prior information about a moving human into statistical inference?** Most current particle filter algorithms use only a dynamic appearance model, few of them consider motion features or patterns. It has been already proved that combining both motion and appearance information can achieve very promising results for human motion detection and recognition [51, 52].
- **How to use bottom-up results as the inference unit in the top-down process?** In other words, how can we build a data driven particle filtering structure to combine both bottom-up and top-down processing? In [53], Tu and Zhu proposed a data driven method for image segmentation, where the fundamental units for inference are pixels. In human motion segmentation, the UC region should be the fundamental unit for object tracking processing.
- **May we use the idea of "object file" theory to attack the occlusion problem by registering tracked objects across frames?** It is well known that occlusion is a difficult problem for particle filtering-based human tracking. Decomposing a complex object into several independent moving parts, tracking them individually, and building an "object file" for each of them may be a good way to deal with the occlusion problem. A similar idea has been implemented in the tracking algorithm proposed in [54].
- **How to use the idea of matching between adjacent frames into the particle filter to enhance the tracking performance?** In the proposed

framework, tracking acts as a bridge between objection segmentation, motion estimation, and action recognition. Tracking processing plays a key role to sustain and stimulate recurrent interactions among them. However, matching between adjacent frames, as a nature of tracking in the HVS, start to received attention in the community of particle filtering research. An appearance-adaptive method proposed in [55] is an exemplary effort in this direction.

– **How to represent and learn prior knowledge for statistical inference?** In general, we assume that prior knowledge is known by certain off-line learning algorithm. When the off-line learned prior information does not fit current observations, tracking performance could suffer. Therefore, good balance between online and off-line learning may greatly improve the robustness and effectiveness of tracking. In [52], Lim et. al. proposed an example of online manifold learning, which has archived promising results.

Acknowledgments

This work is supported by the NSF under Grant IIS-0347613 (CAREER).

References

1. Johansson, G.: Visual perception of biological motion and a model for its analysis. Perception & Psychophysics **14** (1973) 201–211
2. Troje, N.: Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. Journal of Vision **2** (2002) 371–387
3. Thornton, I., Pinto, J., Shiffrar, M.: The visual perception of human locomotion. Cognitive Neuropsychology (1998) 535–552
4. Knoblich, G., Thornton, I.M., Grosjean, M., Shiffrar, M.: The human body. Perception from the inside out. Oxford University Press, New York, NY (2006)
5. Gepshtein, S., Kubovy, M.: The emergence of visual objects in space-time. Proceedings of the National Academy of Sciences of the United States of America **97** (2000) 8186–8191
6. Kubovy, M., Gepshtein, S.: Grouping in space and in space-time: An exercise in phenomenological psychophysics. In Behrmann, M., Kimchi, R., Olson, C., eds.: Perceptual Organization in Vision: Behavioral and Neural perspectives, Lawrence Erlbaum Association, Mahwah, N.J. (2003) 45–85
7. Kubovy, M., Gepshtein, S.: Gestalt: from phenomena to laws. In: Perceptual Organization for Artificial Vision Systems, Academic Publishers, Boston (2000) 41–71
8. C.N.Olivers, Humphreys, G.: Spatiotemporal segregation in visual search:evidence from parietal lesions. Journal of Experimental Psychology: Human Perception and Performance **30** (2004) 667–688
9. Ullman, S.: The Interpretation of Visual Motion. Cambridge, MA: The MIT Press (1979)
10. Marr, D.: Vision: a Computational Investigation into the Human Representation andProcessing of Visual Information. W.H. Freeman and Company, NY (1982)
11. Palmer, S.E., Rock, I.: Rethinking perceptual organization: The role of uniform connectedness. Psychonomic Bulletin and Review **1** (1994) 29–55

12. Palmer, S.: Vision Science: Photons to Phenomenology. Bradford Books MIT Press (1999)
13. McClelland, J.L.: On the time relations of mental processes : An examination of systems of processes in cascade. Psychological Review **86** (1979) 287–330
14. Stillings, N.A., Weisler, S.E., Chase, C.H., Feinstein, M.H., Garfield, J.L., Rissland, E.L.: Cognitive Science: An Introduction. Cambridge, MA: The MIT Press (1995)
15. Palmer, S.E., Brooks, J.L., Nelson, R.: When does grouping happen? Acta Psychologica **114** (2003) 311–330
16. C.W.Clifford, J.Freedman, LM.Vaina: First- and second-order motion perception in gabor micropattern stimuli: Psychophysical and computational modelling. Cogn Brain Res **6** (1998) 263–271
17. Braddick, O.: A short-range process in apparent motion. Vision Res. **14** (1974) 519–527
18. Thornton, I.M., Pinto, J., Shiffrar, M.: The visual perception of human locomotion. Cognitive Neuropsychology **15** (1998) 535–552
19. Franconeri, S.L., Halberda, J., Feigenson, L., Alvarez, G.A.: Common fate can define objects in multiple object tracking. Journal of Vision **4** (2004) 365a
20. Mumford, D. In: Neuronal architecture for pattern-theoretic problems. MIT Press, Cambridge, MA, USA (1993)
21. Bullier, J.: Integrated model of visual processing. Brain Reseach Reviews **36** (2001) 96–107
22. Ullman, S.: High-level vision : object recognition and visual cognition. Cambridge, MA: The MIT Press (1996)
23. Kersten, D., P.Mamassian, A.Yuille: Object perception as bayesian inference. Annual Review of Psychology **55** (2004) 271–304
24. D.C.Knill, W.Richards: Perception as Bayesian Inference. UK: Cambridge Univ. Press (1996)
25. R.P.N.Rao, Olshausen, B., Lewicki, M.: Probabilistic Models of the Brain: Perception and Neural Function. Cambridge, MA: MIT Press (2002)
26. Lee, T., Mumford, D.: Hierarchical bayesian inference in the visual cortex. Journal of Optical Society of America **20** (2003) 1434–1448
27. Vecera, S.P., O'Reilly, R.C.: Figure-ground organization and object recognition processes: An interactive account. Journal of Experimental Psychology: Human Perception and Performance **24** (1998) 441–462
28. Shi, J., Malik, J.: Motion segmentation and tracking using Normalized cuts. In: Proc. of Int. Conf. on Computer Vision. (1998) 1151–60
29. Fowlkes, C., Belongie, S., Malik, J.: Efficient spatiotemporal grouping using the Nystrom method. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (2001) 231–238
30. DeMenthon, D., Megret, R.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. Technical Report: LAMP-TR-090/CAR-TR-978/CS-TR-4388/UMIACS-TR-2002-68 (2002)
31. Greenspan, H., Goldberger, J., Mayer, A.: A probabilistic framework for spatio-temporal video representation and indexing. In: Proc. European Conf. on Computer Vision. Volume 4., Berlin, Germany (2002) 461–475
32. Megret, R., DeMenthon, D.: A survey of spatio-temporal grouping techniques. Technical report, University of Maryland, College Park (2002) http://www.umiacs.umd.edu/lamp/pubs/TechReports/.
33. Moscheni, F., Bhattacharjee, S., Kunt, M.: Spatiotemporal segmentation based on region merging. IEEE Trans. Pattern Anal. Mach. Intell **20** (1998) 897–915

34. V.Mezaris, I.Kompatsiaris, M.G.Strintzis: Video object segmentation using bayes-based temporal tracking and trajectory-based region merging. IEEE Trans. Circuits and Systems for Video Technology **14** (2004) 782–795
35. Gelgon, M., Bouthemy, P.: A region-level motion-based graph representation and labeling for tracking a spatial image partition. Pattern Recognition **33** (2000) 725–740
36. Wang, D.: Unsupervised video segmentation based on watersheds and temporal tracking. IEEE Trans. Circuits and Systems for Video Technology (1998) 539–546
37. Porikli, F., Wang, Y.: Automatic video object segmentation using volume growing and hierarchical clustering. Journal on Applied Signal Processin **3** (2004) 442–453
38. Tsai, Y., C.Lai, Y.Hung, Z.Shih: A bayesian approach to video object segmentation. IEEE Trans. Circuits syst. video Technology **15** (2005) 175–180
39. Hochstein, S., Ahissar, M.: View from the top: Herarchies and reverse hierarchies in the visual system. Neuron **36** (2002) 791–804
40. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2004)
41. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.C.: Image parsing: Unifying segmentation, detection, and recognition. Int'l Journal of Computer Vision **63** (2005) 113 – 140
42. Ungerleider, L.G., Mishkin, M. In: Two cortical visual systems. MIT Press, Cambridge, MA, USA (1982) 549 – 586
43. elleman, F.J., D.C., V.E.: Distributed hierarchical processing in primate cerebral cortex. Cerebral Cortex **1** (1991) 1–47
44. Burr, D., Ross, J.: Vision: The world through picket fences. Current Biology **14** (2004) 381–382
45. Giese, M.A., Poggio, T.: Neural mechanisms for the recognition of biological movement. Nature Neuroscience Review **4** (2003) 179–192
46. Oram, M.W., Perrett, D.I.: Integration of form and motion in the anterior part of the superior temporal polysensory area (STPa) of the macaque monkey. Journal of neurophysiology **76** (1996) 109–129
47. Sajda, P., Baek, K.: Integration of form and motion within a generative model of visual cortex. Neural Networks **17** (2004) 809–821
48. Bullier, J.: Integrated model of visual processing. Brain research review **36** (2001) 96–107
49. D.Kahneman, Terisman, A., Gibbs, B.J.: The reviewing of object files: object specific integration of information. Cognitive Psychology **24** (1992) 175–219
50. Pylyshyn, Z., Storm, R.: Tracking multiple independent target: Evidence for a parallel tracking mechanism. Spatial Vision **3** (1988) 1–19
51. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: Proc. IEEE Int'l Conference on Computer Vision. (2003)
52. Lim, H., Morariu, V., Camps, O.I., Sznaier, M.: Dynamic appearance modeling for human tracking. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2006)
53. Tu, Z., Zhu, S.C.: Image segmentation by data-driven markov chain monte carlo. IEEE Trans on Pattern Anal. Mach. Intell. **24** (2002) 657–673
54. Micilotta, A., Bowden, R.: View-based location and tracking of body parts for visual interaction. In: Proc. of British Machine Vision Conference. (2004) 849–858
55. Zhou, S.K., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Processing **13** (2004) 1491–1506

3D Geometry from Uncalibrated Images

George Kamberov[1], Gerda Kamberova[2], O. Chum[3], Š. Obdržálek[3], D. Martinec[3], J. Kostková[3], T. Pajdla[3], J. Matas[3], and R. Šára[3,⋆]

[1] Stevens Institute of Technology
Hoboken, NJ 07030, USA
gkambero@stevens.edu
[2] Hofstra University,
Hempstead, NY 11549, USA
cscglk@hofstra.edu
[3] Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University
Prague 6, Czech Republic
{sara, martid1, pajdla, chum, matas}@cmp.felk.cvut.cz

Abstract. We present an automatic pipeline for recovering the geometry of a 3D scene from a set of unordered, uncalibrated images. The contributions in the paper are the presentation of the system as a whole, from images to geometry, the estimation of the local scale for various scene components in the orientation-topology module, the procedure for orienting the cloud components, and the method for dealing with points of contact. The methods are aimed to process complex scenes and non-uniformly sampled, noisy data sets.

1 Introduction

In this paper we present a collection of computational methods and a pipeline system which extracts the geometric structure of a scene and makes quantitative measurements of the geometric properties of the objects in the scene from a set of images taken in real-life conditions at different unknown discrete instances over some period of time, without the luxury of performing calibration and parameter estimation during the acquisition. The pipeline does not require a human in the loop. This work was motivated by many applications where, due to communications, synchronization, equipment failures, and environmental factors, an intelligent system is forced to make decisions from an unorganized set of images offering partial views of the scene. The presented pipeline has three principal stages: (i) an image-to-3D-point-and-tangent-plane pipeline; (ii) an orientation-topology module which orients the cloud, assigns topology, and partitions it into

⋆ This research was supported by The Czech Academy of Sciences under project 1ET101210406 and by the EU projects eTRIMS FP6-IST-027113 and DIRAC FP6-IST-027787.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 802–813, 2006.
© Springer-Verlag Berlin Heidelberg 2006

connected manifold components; and (iii) a geometry pipeline which recovers the local surface geometry descriptors at each surface point.

The contributions in the paper are the presentation of the system as a whole, from images to geometry, the estimation of the local scale for various scene components in the orientation-topology module, the procedure for orienting the cloud components, and a method of dealing with points of contact.

The results on two different sets of unorganized images are illustrated through out the paper.

Previous Research. The data acquisition method implied by our scenario could lead to reconstructions of exceptional geometric complexity. The point clouds could be sparse and nonuniform; both positions and tangent plane estimates are expected to be noisy. Because of noise, but also because of the freedom to collect and combine different views, we might have objects that touch each other – such touching objects should be separated. Furthermore, many of the reconstructed surfaces will not be closed; they will have boundary points. There are numerous surface reconstruction methods [1,2,3,10,11,12] which work well for closed surfaces and clouds that satisfy some reasonable sampling conditions (for example, uniformly sampled clouds, evenly sampled clouds, or densely sampled clouds). Such methods require a human in the loop to deal with boundary points, non uniformly sampled clouds, and touching surfaces. Furthermore, recent studies indicate that the polygonal surfaces and other surface type reconstructions do not lead to gained precision in computing geometric properties [35]. What is needed is a method to find neighborhoods tight in terms of surface distance and a robust method for computing the differential properties at a point. The framework proposed in [18] shows how to do this at the 3D point level without ever spending time on reconstruction of polygonal, parametric, or implicit representations. Here we present novel methods for adaptive scale selection and for rapid orientation of a cloud endowed with tangent planes only (not oriented normals), see the relevant subsection in Section 2.2. The main difference with earlier approaches [22,28] dealing with scale selection and orientation of point clouds is that we do not impose limitations on the sampling density, do not assume known noise statistics, and do not use a set of implicit tuning parameters; compared to the orientation propagation in [14,12] which rely on minimum spanning tree, our approach is more efficient since it does global flips on sets of normals.

Our system differs from the system for 3D reconstruction from video proposed in [30], in two aspects: (i) the 3D reconstruction can be obtained from wide baseline images only [24] and (ii) the scene is automatically segmented into objects.

2 System Overview: From Images to 3D Geometry

The system is organized as a pipeline consisting of three modules: the image-data pipeline, the orientation-topology module, and the geometry pipeline. The input data are photographs of the scene to be reconstructed, taken with a hand-held compact digital camera which is free to move in the scene and to zoom in and out. The images are input into the data pipeline which outputs an unorganized

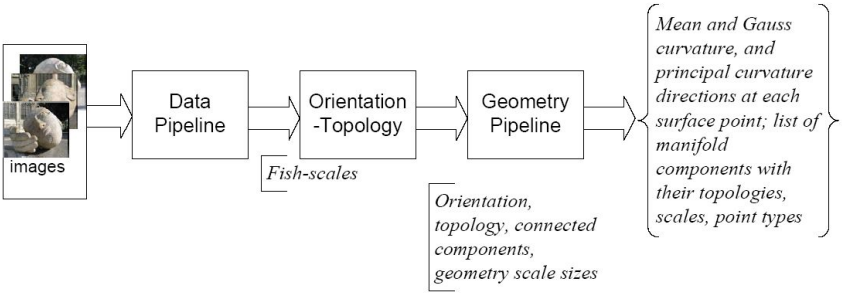


Fig. 1. The system: from images to cloud with 3D geometry descriptors

3D point cloud with fish-scales attached to the points.Each fish-scale encodes a 3D point and an estimate of the plane tangent to the cloud at this point. The fish-scales (3D points with tangent planes) are input into the orientation-topology module which recovers local scale, orientation and topology for the cloud and separates it into connected manifold components. The latter are processed by the geometry pipeline which recovers the geometric descriptors at the surface points.

2.1 The Data Pipeline

The data processing pipeline consists of several steps: (1) wide-baseline matching, (2) structure from motion, (3) dense matching, (4) 3D model reconstruction. The 3D model produced by the data pipeline consist of a collection of fish-scales.

We used the method in [24]. A brief sketch follows. First, sparse correspondences are found across all image pairs. Pairwise image matching is done with Local Affine Frames [26] constructed on intensity and saturation MSER regions, LaplaceAffine and HessianAffine [27] interest points. An epipolar geometry unaffected by a dominant plane is found using [7]. The inliers are used as the pool for drawing samples in calibrated RANSACs. This scheme is applied to the 6-point algorithm [34] as well as to the 5-point algorithm [29]. First the 6-point algorithm is run on all pairs (with some minimal support) and the focal length is estimated as the mean of the estimates from individual pairs. Then, the 5-point algorithm [29] is run using fully known camera internals.

A multi-view reconstruction is estimated given pair-wise Euclidean reconstructions by [29] up to rotations, translations and scales. The partial reconstructions are glued by the following three step procedure: (i) camera rotations consistent with all reconstructions are estimated linearly; (ii) all the pair-wise reconstructions are modified according to the new rotations and refined by bundle adjustment while keeping the corresponding rotations same; (iii) the refined rotations are used to estimate camera translations and 3D points using Second Order Cone Programming by minimizing the L_∞-norm [15].

The method from [24] can be used in extreme cases of missing data, i.e., when each point is visible in two images only in a (sub)set of images. It is capable of

dealing with degenerate situations like dominant planes, pure camera rotation (panoramas) and zooming. The Head2 scene used in this paper is an example in which all these inconveniences appear. No projective-to-metric upgrade is needed as in [23]. Compared to incremental structure-from-motion methods, gluing all pair-wise geometries at the same time has the advantage that the global minimum of an approximation to the reprojection error is achieved [23]. As a consequence, no drift removal [9] is needed. The current limitation of [24] is that the translation estimation using [15] can be harmed by mismatches. This limitation can be removed by using the method in [33] instead. See [25] for a demo with 3D vrml models of difficult data sets.

The next step is pairwise image rectification, which improves matching efficiency. We use Hartley's method [4]. Pairs for dense matching are selected based on the mutual location of the cameras, as described in [8]. After that, dense matching is performed as disparity search along epipolar lines using Stratified Dense Matching [20]. The algorithm has a very low mismatch rate [21], it is fast, robust, and accurate, and does not need any difficult-to-learn parameters. The output from the matching algorithm is one disparity map per image pair admitted for dense matching. By least squares estimation using an affine distortion model the disparity maps are upgraded to sub-pixel resolution [31].



Fig. 2. The Data Pipeline: From an unorganized set of images to fish-scales. A fish-scale is a 3D point with a local covariance ellipsoid centered at it. It encodes the position, a measure of the spatial density and the tangent plane to the cloud at the 3D point.

The disparity maps are used to reconstruct the corresponding 3D points. The union of the points from all disparity maps forms a dense point cloud. An efficient way of representing distributions of points is to use fish-scales [32]. Fish-scales are local covariance ellipsoids that are fit to the points by the K-means algorithm. They can be visualized as small round discs. A collection of fish-scales approximate the spatial density function of the measurement in 3D space.

2.2 Orientation-Topology Module

To deal with the complexity of the data and noise (intermediate point sets recovered in the data pipeline use millions, redundant, noisy points) the points in the final output cloud of the data pipeline are the centers of the fish-scales. This cloud, unorganized set of fish-scale centers, together with the corresponding set of the covariance ellipsoids is the input to the orientation-topology module.



Fig. 3. The Data Pipeline: A series of images and the 3D cloud reconstructed from them. The area around the door is reconstructed at much higher density than the rest of the cloud.

Each covariance ellipsoid defines a tangent plane at the corresponding cloud point. The output of the orientation-topology module is a consistently oriented, topologized point cloud, each point of which is equipped with a normal and a neighborhood of points closest to it in terms of surface distance; the points are classified as isolated, curve, and surface points, and the later points are divided into interior and boundary points; the connected components of the topologized cloud are identified, thus the scene is segmented into objects.

A key parameter in the orientation-topology module is the determination of geometrically meaningful local scales. In our scenario different parts of the scene may be reconstructed at different scales since the unorganized set of input images could consist of images collected at different zoom levels and possibly obtained with different cameras. The orientation-topology module is based on the ideas in [18] combined with novel adaptive scale selection and a method for rapid orientation of the cloud endowed with tangent planes only (not oriented normals), see subsection "Scale, orientation and topology of 3D fish-scales cloud" in Section 2.2. A key ingredient is the method for topologizing oriented point clouds introduced in [17] and outlined below.

Topology of an Oriented Cloud (After [17,18]). An oriented point cloud M is a set of 3D points p with normals $\mathbf{N}$, $M = \left\{(p, \mathbf{N}) \in \boldsymbol{R}^3 \times S^2\right\}$. The neighbors of an oriented point P are chosen based on a proximity measure $L_P : M_\rho(P) \to [0, 1]$, where $M_\rho(P)$ is a ball/voxel centered at p of radius ρ. For every candidate neighbor $Q = (q, \mathbf{N}_q) \in M_\rho(P)$, $L_P(Q)$ expresses the likelihood that Q is nearby P on the sampled surface. The likelihood incorporates three estimators of surface distance: a linear estimator based on Euclidean distance, a quadratic estimator based on the cosine between normals, and a new third order estimator δ_p defined below (the last estimator is crucial in distinguishing points which are far on the surface but close in Euclidean distance),

$$L_P(Q) = \left(1 - \frac{|p-q|}{D_{\text{Eucl}}(P)}\right) t(P,Q), \text{ where } D_{\text{Eucl}}(P) = \max_{Q' \in M_\rho(P)} (|p - q'|),$$

$$t(P,Q) = \frac{1}{2}\left(1 + \langle \mathbf{N}_p \,|\mathbf{N}_q \rangle\right) \left(1 - \frac{\delta_p(Q)}{D_{\text{Surf}}(P)}\right)$$

$$\delta_p(Q) \;=\; |\, \langle \mathbf{N}_q - \mathbf{N}_p \,|\overrightarrow{pq}\rangle + 2\,\langle \mathbf{N}_p \,|\overrightarrow{pq}\rangle\,|, \text{ and } D_{\text{Surf}}(P) = \max_{Q' \in M_\rho(P)} (\delta_p(Q')).$$

Note that $< \cdot|\cdot >$ in the equations above denotes the the Euclidean dot product in $\boldsymbol{R}^3$. Once a scale $\rho > 0$ is chosen, then one can construct a neighborhood $U(P)$ of an oriented point $P = (p, \mathbf{N})$, $U(P) = \{Q_i = (q_i, \mathbf{N}_i) : |p - q_i| < \rho\}$, such that the projection of q_i in the plane through p perpendicular to N is not inside a triangle fan centered at p and whose remaining vertices are projections of the remaining base points in the neighborhood. The last condition implies that each neighbor Q_i carries unique information about the distribution of the surface normals around P. Such a neighborhood is called Δ neighborhood of P and the minimum of $L_P(Q_i)$ is called the likelihood of the neighborhood.

The Δ neighborhoods provide a tool to classify the points in an oriented cloud. The *isolated points* do not have Δ neighborhoods with positive likelihood; the rest of the points are either *curve points*, *boundary surface* points, or *interior surface* points. The point P is a surface point if one can use a Δ neighborhood to estimate the orientation along at least two orthogonal directions emanating from the base point. The point P is an interior surface point if one can use a positive likelihood Δ neighborhood to compute the orientation in a full circle of directions centered at p and perpendicular to $\mathbf{N}$. A point is a curve point if it is not a surface or an isolated point. The surface points are samples of 2D manifolds and we can compute differential properties at each such point.

To build Δ neighborhoods we pre-compute and store in a look-up table the proximity likelihoods for points within an Euclidean distance ρ. After that the construction of the neighborhoods for different points can be done in parallel. There is no additional burden involved in dealing with boundary points – in contrast with the extra steps needed if one uses a Dealunay-Voronoi based approach.

Scale, Orientation and Topology for 3D Fish-Scales Cloud. We use an iterative procedure, `OrientationAndTopology` which simultaneously recovers the *voxel scales* appropriate for the different parts of the cloud, the *orientation and topology*, and partitions the cloud into *connected components*. See Figures 7 and 8.

The iterations correspond to the terms in an increasing sequence of voxel sizes. Each re-entry in the loop processes a current fish-scale cloud equipped with an *entry orientation*.

At the initial entry in the loop: the voxel size is set to equal the largest fish-scale diameter in the fish-scale cloud; and the cloud is assigned an initial orientation by choosing an orientation for each fish-scale plane so that all normals point in the same fixed closed half-space (for example with respect to the $x - y$ plane.

During each iteration a procedure, `FixedScaleOrientationAndTopology` is used to compute orientation and topology adapted to this voxel scale, and the connected manifold components. All sufficiently large 2D manifold components are identified. As a heuristic at the moment we consider components to be large if they contain at least 10% of the fish-scales in the current cloud. (Presently we are developing a density based approach to determining the scale.) The current voxel size is assigned as the corresponding scale for these components, then they are removed from the fish-scale cloud, and are handed to the Geometry pipeline. The entry orientation is updated to equal the normals assigned by `FixedScaleOrientatinAndTopology` and the voxel size is increased linearly.

The loop is reentered until the fish-scales are exhausted.

`FixedScaleOrientationAndTopology` *Procedure:* The input is a voxel scale and a cloud of 3D points with normals. It produces a topology (an adjacency graph) by defining a Δ neighborhood for each point in the cloud, the connected components of the cloud, and a new orientation of the cloud adapted to the Δ neighborhoods. The whole process is organized as an iterative improvement (enlarging connected manifold components from largest to smallest) of some initial orientation and topology for fixed scale. The iteration stops when the components stabilize.

During each iteration we use the current orientation and voxel scale to find a topology by computing Δ neighborhoods. Then the cloud is segmented into connected components by finding the strongly connected components of the adjacency graph defined by the topology. Because the initial normals were chosen up to sign, typically the same geometric object will be split into multiple components (adjacent components with discontinuity in the Gauss map along adjacent component boundaries). Thus we have to synchronize the normals of such components.

The procedure explores the adjacency graph of the connected components (not the whole cloud), always starting at the current largest component that has not been touched previously. Two components are adjacent, if they have a pair of boundary points within a scale unit from each other. The normals are averaged over each of the two boundary points' neighborhoods. If the angle between the average normals is close to π, a decision whether to synchronize the normals of the smaller component with the larger one is made based on the proximity likelihoods of the Δ neighborhoods of the interior points which are neighbors of the boundary points in question. For interior surface points along boundary regions with "opposing" normals, but otherwise similar geometry, the likelihoods will be highly correlated and similar. We sample the proximity likelihoods of points in the Δ neighborhoods of the interior points adjacent to the boundary pair, thus we construct two sets of samples, one sample set for each component. We do a statistical test for the equality of the population means of the two samples, and if the population means are equal, we reverse all normals of the smaller component and mark the smaller component as "flipped". After all connected

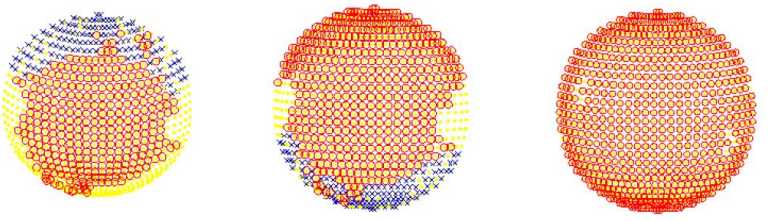


Fig. 4. Growing the topology and orientation. The final stage has a single 2D component which captures 98% of the cloud points. This is a synthetic example using randomly sampled points on a sphere.



Fig. 5. Two tangent surfaces: (Left) The original point cloud. (Right) The two largest connected components: part of a sphere and part of a plane. The two components contain roughly 94% of the cloud.

components with boundaries adjacent to the largest component are examined (and possibly their normals reversed), the topology procedure is run again using the updated normals. The reversal of the normals of a connected component can be implemented very efficiently in almost any programming environment. This orientation propagation amounts to a traversal of a tree with depth not bigger than the number of connected components. This approach allows us to resolve tangency cases as in Figure 5. Then the main loop of the procedure is re-entered until the size of the largest connected component stops to increase. See Figure 4.

2.3 The Geometry Pipeline

The geometry pipeline takes 3D oriented, topologized components and computes the geometric descriptors at surface points (mean curvature, Gauss curvature, and principal curvature directions) following the methods in [14, 13]. These methods use discrete versions of (1)-(3): (1) is a basic identity, [19], used for computing the mean curvature H, $\mathbf{f}$ is a surface parametrization with Gauss map $\mathbf{N}$, and for every tangent vector $\mathbf{v}$, $J(\mathbf{v})$ is the unique tangent vector satisfying $d\mathbf{f}\,(J(\mathbf{v})) = \mathbf{N} \times d\mathbf{f}\,(\mathbf{v})$; the Hopf form, $\omega\,(\mathbf{v})$, defined by (2), is used in computing the symmetric, trace-free coefficient matrix $A = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}$ of the form $< \omega(\cdot)|d\mathbf{f}(\cdot) >$ with respect to an arbitrary positively oriented local basis $(\mathbf{u}, \mathbf{v})$ (note that $< \cdot|\cdot >$ is the Euclidean dot product in $\boldsymbol{R}^3$); and in (3), K is the Gauss curvature, and a and b are the entries of A as defined earlier.



Fig. 6. Geometry Pipeline Output: The two curvature lines foliations extracted on a 2D component in the cloud from Figure 7

$$-Hd\mathbf{f}(\mathbf{v}) = \frac{1}{2}\left(d\mathbf{N}(\mathbf{v}) - \mathbf{N} \times d\mathbf{N}(J(\mathbf{v}))\right) \tag{1}$$

$$\omega(\mathbf{v}) = \frac{1}{2}\left(d\mathbf{N}(\mathbf{v}) + \mathbf{N} \times d\mathbf{N}(J(\mathbf{v}))\right) \tag{2}$$

$$K = H^2 - (a^2 + b^2) \tag{3}$$

The principal curvature vectors of are expressed explicitly in closed form involving the entries of the matrix A. Given an oriented point P and its neighborhood $\mathcal{U}(P) = \{P, P_1, \ldots, P_{k_P}\}$, a directional derivative can be computed along each edge $p_i - p$. Thus, there is a 1-1 correspondence between the neighbors of P and a set of *adapted frames* ϕ_i, defined as $\phi_i = (p, \mathbf{u}_i, \mathbf{v}_i, \mathbf{N})$, where $\mathbf{u}_i$ is a unit vector collinear with the projection of $p_i - p$ in the plane orthogonal to $\mathbf{N}$, and $\mathbf{v}_i = \mathbf{N} \times \mathbf{u}_i$. Redundant computations of the geometry descriptors are done based on each adapted frame ϕ_i, next the extreme values are trimmed and from the rest of the descriptors, and by averaging, estimates of the final geometric descriptors at the point are obtained. Multiple geometry pipelines can be run in parallel for the different components. An extensive study and comparison of the

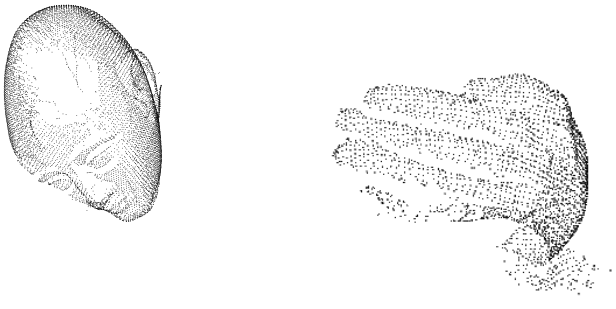


Fig. 7. Orientation-Topology Output: The two largest 2D components of the cloud from Figure 2. Together they contain 12182 points out of the total 12692 points in the scene reconstructed by the Data Pipeline.

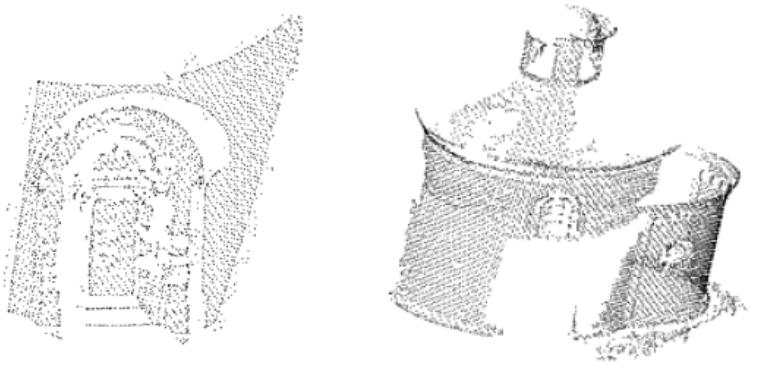


Fig. 8. Orientation-Topology Output: A high-resolution oriented 2D component (left) and a lower resolution component (right) extracted from the cloud in Figure 3. The high resolution component contains 3398 points and the low resolution contains 7012. The total size of the cloud produced by the Data Pipeline is 12499 points.

methods for computing curvatures has been reported [16,35], and here for the purpose of illustration we show two families of principal curvature directions. See Figure 6.

3 Summary

The paper presents a methodology for recovering the geometry of a 3D scene from a set of unordered, uncalibrated images. The images are fed into an automatic data processing pipeline which produces a collection of fish-scales; the fish-scales the are presented as input in an automatic pipeline which assigns orientation and topology, segments the cloud into connected manifold components, and recovers the local surface geometry descriptors at each surface point. The data pipeline, the recovery of the topology of an oriented cloud and the method for computing the geometric surface descriptors have been introduced previously and their performance had been analyzed [8,16,18,24]. The results presented here are for illustrative purposes. The theoretical contributions presented are limited to the iterative procedure that simultaneously recovers orientation, topology and segments the cloud into manifold components. We are conducting a performance evaluation and comparison study focusing on the orientation and scale selection. The results are the subject of a forthcoming paper.

References

1. M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, "Computing and Rendering Point Set Surfaces", *Trans. Vis. Comp. Graph.*, Volume 9(1): 3-15, 2003.
2. Nina Amenta and Yong Joo Kill, "Defining Point-Set Surfaces", ACM Trans. on Graphics, vol. 23(3), 264-270, Special Issue: Proceedings of SIGGRAPH 2004.

3. E. Boyer, Petitjean, S., "Regular and Non-Regular Point Sets: Properties and Reconstruction", Comp. Geometry, vol. 19, 101-131, 2001.
4. R. Hartley, "Theory and Practice of Projective Rectification," Int J Computer Vision, 35(2), 115-127, 1999.
5. R. Hartley, R. and Zisserman, A., "Multiple View GeomeLtry in Computer Vision", Cambridge University Press, 2000.
6. O. Chum, Matas, Jiří, and Kittler, J., "Locally optimized RANSAC," Proc DAGM, 236-243, 2003
7. O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR*, vol. 1, pp. 772–779, 2005.
8. H. Cornelius, Śara, R., Martinec, D., Pajdla, T., Chum, O., Matas, J., "Towards Complete Free-Form Reconstruction of Complex 3D Scenes from an Unordered Set of Uncalibrated Images," SMVP/ECCV 2004, vol. LNCS 3247, (2004) pp. 1-12.
9. K. Cornelis, F. Verbiest, and L.J. Van Gool. Drift detection and removal for sequential structure from motion algorithms. *PAMI*, 26(10):1249–1259, October 2004.
10. T. K. Dey, Sun, J., "Extremal Surface Based Projections Converge and Reconstruct with Isotopy", Tech. Rep. OSU-CISRC-05-TR25, Apr. 2005.
11. H. Edelsbrunner, Mucke, E.P., "Three-dimensional alpha shapes," ACM Trans. Graph., Volume 13(10, 43-72, 1994.
12. M. Gopi, Krishnan, S., Silva, C. T., "Surface reconstruction based on lower dimensional localized Delaunay triangulation", EUROGRAPHICS 2000, Computer Graphics Forum, vol 19(3), 2000.
13. J.-Y. Guillemaut, Drbohlav, O., Sara, R., Illingworth, J., "Helmholtz Stereopsis on rough and strongly textured surfaces ", 2nd Intl. Symp. 3DPVT, Greece, 2004.
14. H. Hoppe, DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W. "Surface reconstruction from unorganized points", *Comp. Graph. (SIGGRAPH '92 Proceedings)*, 26, 71–78, (1992)
15. F. Kahl. Multiple view geometry and the l_∞-norm. In *ICCV05*, pp. II: 1002–1009, 2005.
16. G. Kamberov, G. Kamberova, "Conformal Method for Quantitative Shape Extraction: Performance Evaluation", ICPR 2004, Cambridge, UK, August, 2004, IEEE Proceedings Series, 2004
17. G. Kamberov, G. Kamberova,"Topology and Geometry of Unorganized Point Clouds", 2nd Intl. Symp. 3D Data processing, Visualization and Transmission, Thessaloniki, Greece, IEEE Proc. Series, in cooperation with Eurographics and ACM SIGGRAPH, 2004.
18. G. Kamberov, Kamberova, G., Jain, G., "3D shape from unorganized 3D point clouds", Intl. Symposium on Visual Computing, Lake Tahoe, NV, Springer Lecture Notes in Computer Science, 2005.
19. G. Kamberov, P. Norman, F. Pedit, U. Pinkall, "Quaternions, Spinors, and Surfaces", American Mathematical Society , 2002, ISBN 0821819283.
20. J. Kostková and Šára, R. "Stratified Dense Matching for Stereopsis in Complex Scenes," Proc BMVC, Vol. 1, 2003.
21. J. Kostková and Šára, R. "Dense Stereomatching Algorithm Performance for View Prediction and Structure Reconstruction," Proc SCIA, 101-107, 2003.
22. J. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert "Fifth Scale Selection for Classification of Point-sampled 3-D Surfaces," Proc. International Conference on 3-D Digital Imaging and Modeling (3DIM 2005), 2005.
23. D. Martinec, and Pajdla, T. "3D Reconstruction by Fitting Low-rank Matrices with Missing Data," Proc CVPR 2005, Vol. I, pp. 198-205.

24. D. Martinec and T. Pajdla. 3D reconstruction by gluing pair-wise euclidean reconstructions, or "how to achieve a good reconstruction from bad images". In *3DPVT*, p. 8, University of North Carolina, Chapel Hill, USA, June 2006.
25. D. Martinec, demo at http://cmp.felk.cvut.cz/~martid1/demo3DPVT06.
26. J. Matas, Š. Obdržálek, and O. Chum. Local affine frames for wide-baseline stereo. In *ICPR(4)*, pp. 363–366, 2002.
27. K. Mikolajczyk et al. A Comparison of Affine Region Detectors. *IJCV*, 2005.
28. N. Mitra, N. Nguyen, and L. Guibas "Estimating surface normals in noisy point cloud data," International Journal of Computational Geometry & Applications, Vol. 14, Nos. 4&5, 2004, pp. 261–276.
29. D. Nistér. An efficient solution to the five-point relative pose. *PAMI*, 26(6):756–770, June 2004.
30. M. Pollefeys et al. Image-based 3d recording for archaeological field work. *CG&A*, 23(3):20–27, May/June 2003.
31. R. Šára, "Accurate Natural Surface Reconstruction from Polynocular Stereo," Proc NATO Adv Res Workshop Confluence of Computer Vision and Computer Graphics, NATO Science Series No. 84, pp. 69-86, Kluwer, 2000.
32. R. Šára and Bajcsy, R., "Fish-Scales: Representing Fuzzy Manifolds", *Proc. Int. Conf. on Computer Vision*, Bombay, India, Narosa Publishing House, 1998.
33. K. Sim and R. Hartley. Recovering camera motion using l_∞ minimization. In *CVPR*, vol. 1, pp. 1230–1237, New York , USA, June 2006.
34. H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *CVPR*, vol. 2, pp. 789–794, 2005.
35. T. Surazhsky, Magid, E., Soldea, O., Elber, G., and Rivlin, E., "A comparison of Gaussian and mean curvatures estimation methods on triangular meshes", 2003 IEEE Interntl. Conf. on Robotics & Automation (ICRA2003)

Hierarchical Image Database Navigation on a Hue Sphere

Gerald Schaefer[1,2] and Simon Ruszala[2]

[1] School of Engineering and Applied Science, Aston University, U.K.
[2] School of Computing and Informatics, Nottingham Trent University, U.K.

Abstract. Due to the ever growing size of image collections efficient and effective ways of browsing and navigation through them as well as locating and searching desired images are in high demand. Most current approaches project image thumnails onto a 2-dimensional visualisation space so that images that are visually similar are placed close to each other on the display. Unfortunately, this approach is only of limited use for larger image sets as images are occluding each other while at the same time certain parts of the projection space are not utilised. In this paper we provide an image database navigation approach that does not suffer from these disadvantages. Images are projected on a spherical visualisation space which users can intutively rotate to focus on images with a different main hue or brightness. Thumbnails are placed on a regular grid on the sphere's surface to avoid any overlapping effects. Large image datasets are handled through a clustering technique which allows browsing in a hierarchical manner.

Keywords: image database navigation, image database visualisation, content-based image retrieval.

1 Introduction

Efficient and effective ways for managing and browsing large image databases are increasingly thought after. This is due to sheer explosion of availability of digital images during the last few years. Nowadays, the sizes of home user's image collections are already typically in the 1,000s while professional image providers overlook databases in excess of 1,000,000 images. Common tools display images in a 1-dimensional linear format where only a limited number of thumbnail images are visible on screen at any one time, thus requiring the user to search back and forth through thumbnail pages to view all images. Obviously, this constitutes a time consuming, impractical and exhaustive way of searching images, especially in larger catalogues. Furthermore, the order in which the pictures are displayed is based on attributes like file names and does not reflect the actual image contents and hence cannot be used to speed up the search.

Recently, several approaches have been introduced which provide a more intuitive interface to browsing and navigating through image collections [10]. The basic idea behind most of these is to place images which are visually similar,

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 814–823, 2006.
© Springer-Verlag Berlin Heidelberg 2006

as established through the calculation of image similarity metrics based on features derived from image content, also close to each other on the visualisation screen, a principle that has been shown to decrease the time it takes to localise images [8]. One of the first approaches was the application of multidimensional scaling (MDS) [3] used to project images being represented by high dimensional feature vectors to a 2-dimensional visualisation plane [9]. In the PicSOM system [4] tree-structured self organising maps are employed to provide both image browsing and retrieval capabilities. In [2] a hierarchical tree is employed to cluster images of similar concepts while the application of virtual reality ideas and equipment to provide the user with an interactive browsing experience was introduced in [7].

While the application of techniques such as MDS provides an intuitive and powerful tool for browsing image collections it is only of limited use for medium-sized and large image collections. For such databases it provides a relatively poor representation as many images are occluded, either fully or partially, by other images with similar feature vectors. In addition, empty spaces are common in areas where no images fall, creating an unbalanced representation on screen. In this paper we introduce an image database navigation method that does not suffer from these drawbacks. We employ a spherical visualisation space where images are placed on the surface of a globe based on their average hue and brightness [12]. Users hence have the possiblity to browse an image collection through rotation of the sphere which changes the average colour and through tilting which will focus on either brighter or darker images. Projection of image thumbnails onto the globe's surface is done in a hierarchical manner which can cope also with large image datasets and also has the advantage that all levels of the hierarchy can be pre-computed, thus allowing real-time browsing of the image database. In addition, images are laid out on a regular grid structure which avoids any unwanted overlapping effect between images. Furthermore, the visualisation space is better utilised by branching out images into otherwise unoccupied parts of the screen. The proposed method hence provides an effective, intuitive and efficient interface for image database navigation as is demonstrated on a medium sized image collection.

2 Hierarchical Image Database Navigation on a Hue Sphere

Current approachs to browsing image collections are often based on projecting image thumbnails on a 2-dimensional visualisation plane so that images which are visually similar (based on features such as colour, texture or shape descriptors) are also placed close to each other in the navigation interface. One common way of achieving this is through the application of multidimensional scaling (MDS) [3] as suggested in [9].

In Figure 1 we show an MDS plot of about 1400 images. As can be seen, this type of navigation techniques is not very well suited for large or even medium-sized image collections as images are either totally or partially occluded by other



Fig. 1. MDS display of about 1400 images

images with similar feature vectors; obviously the more images in the dataset the higher the probability of occlusion/overlapping. On the other hand, areas of the visualisation space in which no images fall remain empty and hence create and unbalanced representation.

In this paper we propose an image database visualisation and navigation approach that does not suffer from the disadvantages of occlusion and unused space. Based on our previous work on a spherical visualisation space [12] we propose a system that provides an intuitive and fast interface which is also suitable for large image collections.

2.1 Hue Sphere

All image database visualisation techniques are based on some underlying image features. We base our approach on the colour content of images and describe each image by its median colour. However rather than employing the standard RGB colour space we use the HSV space [11] as it more closely reflects the human visual system. We take the hue and value attributes only as the saturation descriptor is deemed less important for describing image content. Value, which describes the brightness of a colour is defined in HSV as [11]

$$V = \frac{R + G + B}{3} \tag{1}$$

where R, G, and B are red, green and blue pixel values. V ranges between 0 and 1 where 0 corresponds to pure black and 1 to pure white. Hue "*is the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colours, red, yellow, green and blue, or a combination of two of them*" [1] and is the attribute that is usually associated as 'colour'. Hue in HSV is defined as [11]

$$H = \cos^{-1} \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \quad (2)$$

It is apparent that hue constitutes an angular attribute; H goes from red to yellow to green to blue back to red and is also often referred to as hue circle. Considering this we choose a sphere as visualisation space onto which image thumbnails are projected [12]. Hue and value are converted to longitude and latitute values through a simple scaling transformation to arrive at the co-ordinates at which an image thumbnail is placed. As all colours with high V values are similar i.e. close to white and the same holds true for those colours with low V which become similarly close to black, and as black and white by definition don't have a hue quality, the two points $V = 0$ and $V = 1$ describe the poles of the sphere.

The use of this hue sphere not only comes naturally with the choice of features, it also has other clear advantages. The concept of a globe will almost certainly be familiar to the average user as it is a direct analogy of the earth globe. It therefore provides a very intuitive interface to the user who will have experience on how to navigate and find something on its surface. Since the co-ordinates are extracted directly from the images our approach is very fast and hence much more efficient than methods such as MDS which rely on a comparably slow and computationally expensive iterative convergence procedure. In addition, the features that we employ are intrinsically suitable for query-based image retrieval [13]. Furthermore, the axes of the co-ordinate system are well defined and meaningful which is in contrast to those obtained by MDS or other similar techniques (such as principal components analysis) where axes are not associated with attributes.

2.2 Grid Layout

Follwing the above procedure to project an image collection onto a hue sphere it is clear that out of necessity images will partially overlap with each other. Yet, as has been observed previously, this overlapping will have a negative impact on the browsing experience [8]. While approaches to minimise these effects have been proposed in [5], where images are slightly moved as a result of a local optimisation problem, these constitute only a partial solution to the problem. In [8] a user study is reported which compared image visualisation models where images overlap with each other as is the case in a typical MDS layout with those where images are placed on a regular lattice without any overlapping. The results demonstrated that users largely prefer the regular layout as for the unconstrained one overlap adds to confusion between images. Therefore it was concluded that visualisations that avoid overlapping will lead to faster retrieval times and more intuitive browsing interfaces. In our approach we adopt these findings and constrain images to be placed on a regular grid structure where images do not overlap each other. That is, we carry out the initial hue/brightness calculation as detailed in Section 2.1 but then map each image to its closest grid cell. The size of the grid structure is typically set to 24x30 or 27x35 depending

on monitor size and resolution. Clearly, and in particular for larger image sets, this will mean that more than one image can be mapped to a particular cell; in Section 2.4 we will describe how we are handling this case in an efficient and intuitive way through the employment of a hierarchical structure.

2.3 Filling Empty Cells

While constraining images to a grid lattice prevents any overlapping effects, in essence it provides a "quantised" form of a hue sphere display. Thus, it still suffers from the relatively unbalanced view that is usually generated where certain areas of the visualisation space are not filled, which is in particular often the case for smaller image collections. To address this problem and to provide a more balanced browsing screen, local search strategies are employed which move images across grid boundaries to previously unoccupied cells. First the positions of all empty cells are retrieved. For each of these cells the 4-neighbourhood is then inspected. If 3 or 4 of the neighbours are occupied a relative percentage of those images closest to the borders is moved across the border to fill the previously empty cell as is illustrated in Figure 2.

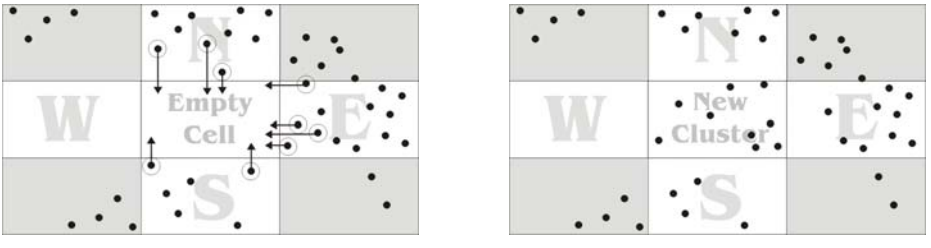


Fig. 2. Images from neighbouring cells being allocated to previously empty cell

Performing this operation will usually fill some but not all of the empty cells. However, repeating the process based on the newly generated layout will in turn fill more cells. Hence the process is repeated a few (usually 3 or 4) times. Not all empty cells will have been assigned after that but then this is not desired as it would mean images being positioned too far from their original co-ordinates which in turn would distort the overall premise that images that are visually close should remain close on the browsing display.

2.4 Hierarchical Browsing

As mentioned above classical visualisation approaches will provide only limited usability when being applied to large but also to medium-sized databases of a few thousand images. The reason for this is that due to the limited space on the visualisation plane images not only overlap each other partially but many images do not appear at all due to occlusion and hence only a partial view of the database is provided. This can be clearly observed in the MDS plot of

Figure 1. Zooming in provides only a partial solution, in particular if there are many images with similar image features.

In our approach we employ a hierarchical tree structure to address both the navigation through large image collections and to eliminate the need for further computations. Hierarchical browsing environments such as the one described in [2] have been shown to provide an effective and efficient way of moving through large image datasets. In our approach we make direct use of the grid mapping introduced above to build a hierarchical tree based on clustering images. The resolution of the grid layout (e.g. 24x30 cells) directly determines the maximal number of clusters present at a given level (which will only be met if all cells are filled). The grid cells (after applying the filling strategy explained in Section 2.3) also determine which images fall into which clusters. What remains to be performed is the selection of a representative image to be displayed on the visualisation grid. To do this we simply select the centroid image I_c, that is the image for which the cumulative distance to all other images in the cluster

$$D_i = \sum_{j=1}^{N} d(I_i, I_j) \tag{3}$$

where I_i is the i-th of N images in the cluster and $d(.,.)$ denotes the distance between two images, is minimal, i.e.

$$D_c < D_i, \forall i \neq c \tag{4}$$

Distances $d(.,.)$ are calculated directly from the longitude and latitude values of the respective images using standard spherical trigonometry (note that we always keep both the original MDS and the derived co-ordinates for each image).

This procedure is adopted at the each level of the tree hierarchy, i.e. first at the root node (the initial global display) and then for each non-empty cell again in a recursive manner, where the images of each child node are again mapped to a grid structure, until the complete tree is derived.

The resulting interface provides an intuitive way of browsing to the user who can, starting from the initial display, successively select a representative image to refine the search. That image cluster is then expanded in the next level of the display where the user can again select an image group to navigate further into the image collection. Based on a grid of 24x30 cells and a fairly conservative estimate of 40% of cells being filled on average this approach requires - on average - only 3 levels of the hierarchy to provide access to each of more than 23 million $(((24 * 30) * 0.4)^3)$ images (i.e.) which will suffice for even the largest images databases these days.

The grid-tree structure also provides another advantage. As the structure is fixed it can be pre-computed in completeness off-line, together will all possible grid views configuration the user can encounter[1] which in turn provides the user with the possibility of real-time browsing large image collection.

[1] These structures only contain pointers to the images and can hence be maintained in memory. Image thumbnails are then loaded upon request.

2.5 Image Spreading in Tree Cells

In the tree nodes of the cells it will commonly occur that only a few images occur most of which will be visually fairly similar. To avoid them from being mapped to the same cell and hence to trigger another tree level, a spreading algorithm is applied which displays them on the same screen once only a certain percentage of cells are filled for a cluster (we currently set this threshold to 25%).

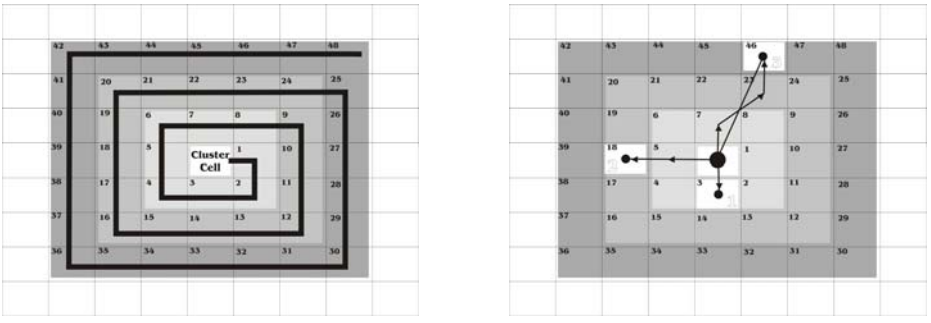


Fig. 3. Spread strategy for filled cells at tree nodes

The algorithm is based on the "place", "bump" and "double-bump" principle and is similar to the one employed in [8]. When a cluster is encountered a spiral scan is initiated that searches for and fills empty cells close by until all images are distributed. If an empty cell is encountered on the first ring around the cell, the next image of the cluster is assigned to that cell ("place"). When an empty cell in the second ring is found it is first established which of the cells of the first ring is closest to the direct path from to the identified empty cell. The image from the thus identified cell is then moved to the empty cell whereas the next image from the cluster is placed in the cell from the first ring ("bump"). The same principle is applied to empty cells identified in the third ring with images from the first and second ring being moved ("double bump"). Both the spiral scan and the three placement strategies are illustrated in Figure 3.

3 Experimental Results

We tested our novel approach to image database navigation on the MPEG-7 common colour dataset [6], a medium-sized image collection of about 4500 images. In the following figures we give examples of the capabilities of our approach.

A global view of the image database based on our hierarchical hue sphere approach is given in Figure 4. As can be seen, the globe as a visualisation body provides an intuitive interface for navigation where it is clear to the user in which part to look for certain image classes. We further notice that in contrast to a standard MDS layout where many images overlap each other (see Figure 1), here the grid structure greatly contributes to the clarity of the visualisation.



Fig. 4. Global hue sphere view of the MPEG-7 dataset



Fig. 5. View after rotation and tilt operations

In Figures 5 and 6 we show the results of some user interaction where the user first rotated the sphere to focus on images with a different hue followed by a tilt operation to bring up darker images resulting in the view given in Figure 5. Figure 6 shows the result of a zoom operation where the user chose one of the images to display those pictures that are contained in that selected part of the tree. To aid navigation the previous hierarchy level are also displayed and the current position within those grid marked with the red dot. At the shown level the image spreading algorithm described in Section 2.5 has been automatically applied as originally fewer of the cells were filled.



Fig. 6. View after zoom operation

4 Conclusions

We presented an intuitive, effective and efficient technique to image database navigation. Each image is characterised by its median hue and brightness which then provide the co-ordinates of the location of the image thumbnail on a hue sphere. A hierarchical tree structure is generated which can be pre-computed completely and is hence able to provide image browsing functionality in real time. Overlapping and occlusion effects are avoided through the adoption of a regular grid layout paired with a hierarchical browsing functionality.

References

1. CIE. *International Lighting Vocabulary.* CIE Publications 17.4, Commission International de L'Eclairage, 4th edition, 1989.
2. S. Krishnamachari and M. Abdel-Mottaleb. Image browsing using hierarchical clustering. In *4th IEEE Symposium on Computers and Communications*, 1999.
3. J.B. Kruskal and M. Wish. *Multidimensional scaling.* Sage Publications, 1978.
4. J. Laaksonen, M. Koskela, P. Laakkso, and E. Oja. PicSOM - content-based image retrieval with self organising maps. *Pattern Recognition Letters*, 21:1197–1207, 2000.
5. B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T.S. Huang. Visualization and user-modeling for browsing personal photo libraries. *Int. Journal of Computer Vision*, 56(1-2):109–130, January 2004.
6. Moving Picture Experts Group. Description of core experiments for MPEG-7 color/texture descriptors. Technical Report ISO/IEC JTC1/SC29/WG11/ N2929, 1999.
7. M Nakazato and T.S. Huang. 3D MARS: Immersive virtual reality for content-based image retrieval. In *IEEE Int. Conference on Multimedia and Expo*, 2001.

8. K. Rodden, D. Basalaj, W. ans Sinclair, and K. Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *IEEE Symposium on Information Visualization*, pages 36–43, 1999.
9. Y. Rubner, L. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Image Understanding Workshop*, pages 661–668, 1997.
10. S.D. Ruszala and G. Schaefer. Visualisation models for image databases: A comparison of six approaches. In *Irish Machine Vision and Image Processing Conference*, pages 186–191, 2004.
11. J. Sangwine and R.E.N. Horne. *The Colour Image Processing Handbook*. Chapman & Hall, 1998.
12. G. Schaefer and S. Ruszala. Image database navigation: A globe-al approach. In *Int. Symposium on Visual Computing*, volume 3804 of *Springer Lecture Notes on Computer Science*, pages 279–286, 2005.
13. M. Stricker and M. Orengo. Similarity of color images. In *Conf. on Storage and Retrieval for Image and Video Databases III*, volume 2420 of *Proceedings of SPIE*, pages 381–392, 1995.

Image-Based Informatics for Preclinical Biomedical Research

Kenneth W. Tobin[1,*], Deniz Aykac[1], V. Priya Govindasamy[1], Shaun S. Gleason[2], Jens Gregor[3], Thomas P. Karnowski[1], Jeffery R. Price[1], and Jonathan Wall[4]

[1] Image Science and Machine Vision Group, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831
{Tobinkwjr, Aykacdb, Muthussamygov, Karnowskitp, Pricejr}@ornl.gov
http://www.ornl.gov/sci/ismv
[2] Siemens Preclinical Solutions, Knoxville, Tennessee 37932
Shaun.Gleason@Siemens.com
[3] Department of Computer Science, University of Tennessee, Knoxville, Tennessee 37996
Jgregor@cs.utk.edu
[4] University of Tennessee Graduate School of Medicine, Knoxville, Tennessee 37920
Jwall@mc.utmck.edu

Abstract. In 2006, the *New England Journal of Medicine* selected medical imaging as one of the eleven most important innovations of the past 1,000 years, primarily due to its ability to allow physicians and researchers to visualize the very nature of disease. As a result of the broad-based adoption of micro imaging technologies, preclinical researchers today are generating terabytes of image data from both anatomic and functional imaging modes. In this paper we describe our early research to apply content-based image retrieval to index and manage large image libraries generated in the study of amyloid disease in mice. Amyloidosis is associated with diseases such as Alzheimer's, type 2 diabetes, chronic inflammation and myeloma. In particular, we will focus on results to date in the area of small animal organ segmentation and description for CT, SPECT, and PET modes and present a small set of preliminary retrieval results for a specific disease state in kidney CT cross-sections.

1 Introduction

Imaging performs an extremely important role in the understanding of human disease through its preclinical application to small animal research. High-resolution, high-throughput, multi-modality imaging provides the capability to carry out non-destructive, longitudinal studies on large populations of animals that previously required animal sacrifice and painstaking dissection to accomplish. A result of this advancing capability is the generation of copious amounts of digital image and

* Prepared by Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 824 – 834, 2006.
© Springer-Verlag Berlin Heidelberg 2006

ancillary data. Most preclinical researchers today maintain this data in an ad-hoc manner, distributed across a number of computers, on different media types, and in different physical locations. For those who have dedicated picture archiving and communications systems (PACS) in their facilities, the largest data component in the system - the imagery - is only marginally indexed for retrieval using simple associated text fields. Over time this image data loses its informational power simply because it becomes irretrievable, limiting a researcher's ability to pursue research questions that leverage the historical image repository.

The Oak Ridge National Laboratory (ORNL), the University of Tennessee Graduate School of Medicine (UTGSM), and the University of Tennessee Department of Computer Science (UTCS) are working together to develop an informatics system for small animal imaging that will support UTGSM's research in the study of amyloid disease in mice. Amyloidosis is a protein aggregation disorder associated with a growing number of fatal and debilitating diseases, such as Alzheimer's disease, type 2 diabetes, chronic inflammatory disorders, and myeloma.

Our main focus area in this regard is to apply content-based image retrieval (CBIR) methods to describe and index the multiple modes of anatomic and functional imagery that are generated through these studies and to make that image data retrievable in future studies. CBIR refers to techniques used to index, retrieve, and manage images from large image repositories based on visual content. Visual content is derived from the structures, morphology, and textures intrinsic to the 2D and 3D multi-modal imagery used for small animal imaging today such as micro CT, PET, SPECT, and MRI.[1] There are many researchers today that are applying CBIR to the general problem of image retrieval [1] and to the biological or biomedical fields [2], but there is not yet a functional PACS that takes advantage of both the extrinsic and intrinsic characteristics of imagery - particularly anatomic and functional imagery - to facilitate "what-if" search scenarios through terabytes of image data to locate images related by morphology, visual phenotype expression, and disease pathologies in the preclinical research environment.

We have developed and fielded CBIR technology and data management systems for the semiconductor industry that address similar problems created by the growing proliferation of automated microscopy inspection in semiconductor manufacturing applications, i.e., the management and reuse of the large amounts of image data collected during semiconductor wafer inspection and review [3, 4]. We have adapted this technology to other fields including geographical information science [5] and retinal diagnostics.[2] In this paper we will describe our preliminary results to date in the area of small animal organ segmentation and description for CT, SPECT, and PET modalities and present a small set of preliminary retrieval results for a specific disease state in kidney cross-sections. Our goal is to present the utility of applying CBIR methods and technology to the informatics of preclinical, small animal research. In Section 2 we will review our previous research and motivation for the use of mouse models to research disease, in particular amyloidosis. In Section 3 we will review our

[1] Computed Tomography (CT), Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), and Magnetic Resonance Imaging (MRI).

[2] R01 EY017065-01, Edward Chaum (PI); Automated Screening for Diabetic Retinopathy by Content.

progress to date on 2D and 3D segmentation of soft tissue organs in micro CT imaging along with some preliminary results for fusing anatomic CT and functional PET data. In Section 4 we will review our preliminary results related to the indexing of a small population of kidney cross-sections from CT and discuss architectural concepts for structuring the representation of image features in a small animal informatics environment.

2 Small Animal Imaging in the Study of Amyloidosis

The amyloidoses represent an ever growing number of insidious diseases characterized by the aggregation of normally innocuous, soluble protein or peptides into highly ordered fibrils that accumulate in tissues and vital organs leading to organ dysfunction and ultimately death [6, 7]. The history of basic and clinical amyloid research has been strongly dependent on imagery and visual peculiarity.

Recently, advances in medical imaging and tracer development have made possible the visualization of amyloid lesions in patients with systemic and cerebral disease and micro-imaging technologies have fueled preclinical research into the pathobiology of amyloid disease in mouse models and the development and evaluation of novel diagnostic and therapeutic agents. For example, UTGSM has developed a multi-disciplinary research program[3] that focuses on non-invasive microimaging of amyloid deposits in mice to better understand the pathogenesis of these fatal diseases and to provide tools (e.g., including both animal models and imaging methods) to examine the efficacy of novel therapeutic, anti-amyloid agents *in vivo* [8, 9].

More specifically, UTGSM, UTCS, and ORNL, working with Siemens Preclinical Solutions, have fabricated a hybrid SPECT/CT microimaging system (see Fig. 1, top) and used it, in addition to a dedicated microPET instrument (Fig. 1, bottom), to provide quantitative images of amyloid deposits in transgenic mice using, as a tracer, a highly specific radioiodinated amyloid-binding protein [8, 10].



Fig. 1. UTGSM/ORNL in vivo laboratory animal systems used for imaging systemic amyloidosis in mice. Siemens microCAT™ II + SPECT (top); microPET P4 system (bottom).

[3] RO1 EB000789, Jonathan Wall (PI); "SPECT/CT Imaging of Systemic AA-Amyloidosis in Mice."

We have developed extensive experience with high-fidelity reconstruction of both CT and SPECT images. We have thus developed a parallel-processing version of the Feldkamp algorithm for cone beam imaging [11, 12] in a distributed computing framework that facilitates the use of iterative reconstruction algorithms, e.g. [13, 14]. With respect to SPECT we have implemented an OSEM algorithm using a conic view based system model as described in [15]. These codes have all been integrated with the RVA™ software that runs on the Siemens microCAT™ family of machines from which the SPECT/CT image data shown in this paper have been obtained. MicroPET/CT co-registered images (e.g. Fig. 2) were generated using data from a P4 microPET imaging system (Siemens Preclinical Solutions) and contrast-enhanced CT data from the microCAT II. Imaging protocols have now been developed that allow us to generate high-resolution SPECT and PET images of amyloid deposits in the viscera of mice, that are readily co-registered with anatomic CT, to provide highly detailed easily interpreted visualizations, as shown in Fig. 2 [16].

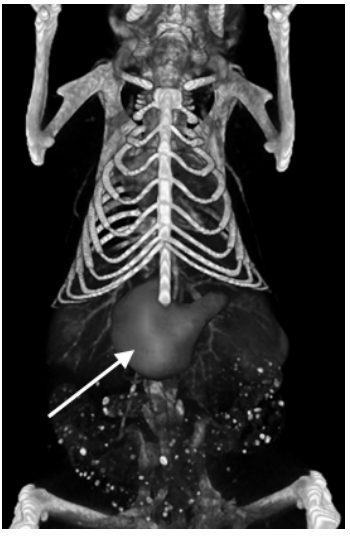


Fig. 2. MicroPET/CT visualization of AA amyloid deposits (arrow) in a mouse using ^{124}I-SAP. Visualization generated by UTGSM.

3 Image Analysis Segmentation and Registration

Quantitative analysis of small animal imagery requires the segmentation of anatomic structures of interest. The results of such segmentation – the shape and volume of a specific organ, for example – can serve as important features for query and retrieval from a CBIR system. In our earlier work in 2D biomedical image segmentation, we applied probabilistic shape and analysis models (PSAM) to the segmentation of soft-boundary anatomic structures [17]. The PSAM method uses an *a priori* model based on a set of landmark points selected by the user during a training step. The PSAM method has been shown effective for anatomic structures

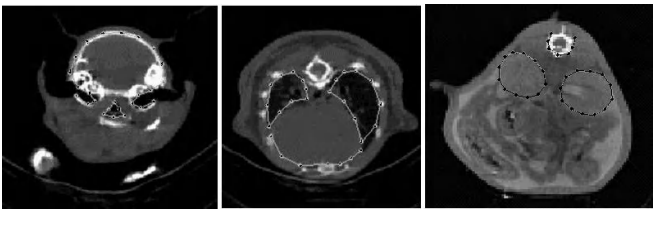


Fig. 3. Examples of segmentations achieved on various anatomic structures using the PSAM method. Cranium (left), heart and lungs (middle), and kidneys (right).

that are relatively uniform across populations such as the kidney or skeletal components (see Fig. 3), but for other organs with greater morphological variability, we have taken an alternative approach.

We have recently developed relevant 3D image segmentation methods [18] in support of quantifying amyloid deposits in a transgenic mouse model of AA-amyloidosis [19] via anatomic and functional imaging. To date, we have focused on identifying the spleen surface using contrast-enhanced CT data with the aim of applying that surface to co-registered SPECT or PET data to quantify the amount of amyloid tracer inside the spleen. Content defined by 3D visual information is a growing topic of research today as a result of the increasing availability of volumetric imaging modalities.

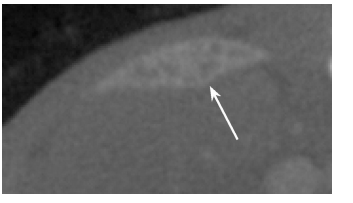


Fig. 4. Example CT slice showing mouse spleen (arrow) after injection of a venous contrast agent. Bright interior regions represent the blood pool and the darker regions represent follicles.

Segmentation of the mouse spleen (and other organs) in CT imagery can be difficult because of poor contrast, even with the use of contrast-enhancing agents. An example CT slice showing a spleen region can be seen in Fig. 4; this data was acquired using 350μl of Fenestra vascular contrast (VC) agent (Advanced Research Technologies/ Alerion Biomedical Inc., Quebec, Canada). The blood pool of the spleen is relatively bright, while the darker interior regions represent the lymphoid follicles. We previously [18] adapted a 2D level sets segmentation method [20] for semi-automatic 3D spleen segmentation. We extended the 2D algorithm to 3D via slice-by-slice processing and also improved performance by introducing statistical and proximity (relative to the previous slice) weighting terms. An example result from this approach can be seen in Fig. 5, where the spleen labeled "Auto" was segmented with our approach and the spleen labeled "Manual" was segmented via manual slice-by-slice thresholding, which is often the method of choice in the pre-clinical setting.

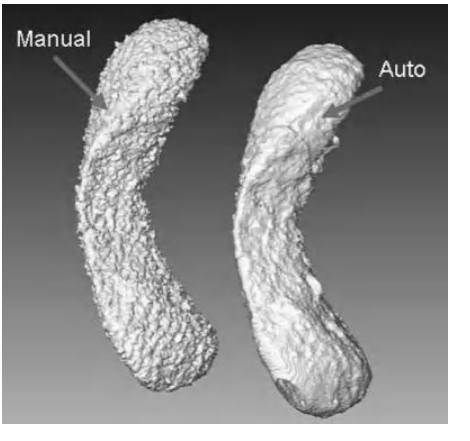


Fig. 5. Example spleen segmentation result ("Auto") using our 3D slice-by-slice approach. The "Manual" spleen was acquired using slice-by-slice, manually-adjusted thresholding.

We have very recently developed a fully 3D (i.e., not slice-by-slice) level set method for segmentation of both the spleen and its interior follicular architecture [21]. In addition to adapting our statistical and proximity weighting terms to the 3D case, we also implemented a modification to the level set energy functional that significantly improved follicle segmentation. An example

result from this algorithm is shown in Fig. 6, where the spleen is rendered transparently and the follicles are opaque. Segmentation of the follicles is important for two reasons. First, we are interested in quantifying the volume occupied by the "blood pool" red pulp as this is decreased as amyloid deposits accumulate in the spleens of mice with systemic AA amyloidosis. Secondly, variations in the follicular architecture may indicate changes in the extent of the amyloid disease and that of other lymphoproliferative disorders that involve the spleen, such as lymphoma. Quantifying the follicle structure and storing in the CBIR system will allow us to detect changes – statistically across populations and/or in single-subject longitudinal studies – that indicate important biological changes. Thus, indexing the volumetric data by its image content becomes extremely relevant to the study of amyloid disease.

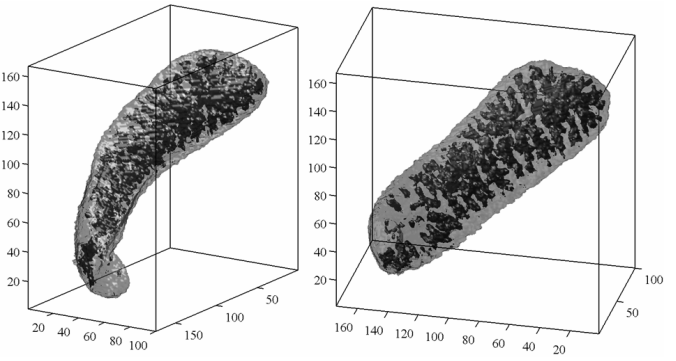


Fig. 6. Result from our recently developed fully 3D spleen (transparent) and follicle (opaque) segmentation technique

We have also used the results from anatomic (CT) image analysis to quantify functional (PET) data. For example, we have recently applied the spleen and follicle boundaries from segmentation to co-registered PET data to quantify the amount of amyloid-specific tracer in the blood pool and follicular regions of an effected spleen. We aligned the CT and PET data sets using fiducial markers visible in both CT and PET modalities. Rigid transformation parameters were calculated via constrained gradient descent, though more sophisticated approaches certainly exist if needed [22, 23]. Two viewpoints of volume-rendered, co-registered PET and CT data can be seen in Fig. 7, where the bright PET region (arrow) corresponds to amyloid burden in the spleen.

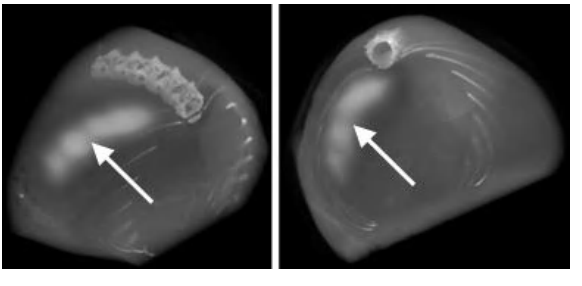


Fig. 7. Volume renderings (two viewpoints) of co-registered CT and PET data. The bright PET region (arrow) corresponds to induced amyloid deposits in the spleen. The spleen boundary is applied to the PET data to quantify the amyloid-bound radiotracer in the blood pool and follicular regions.

4 Preliminary Results and Architecture Concepts for CBIR

Both the context and the content of image data represent an inherent source of information and learning that is only marginally accessed today in the biomedical clinical and preclinical environments [24, 25]. While researchers continue to discuss how this may be achieved, practical progress has not kept pace with the production of imagery [26-28]. *Image context* is represented by extrinsic properties such as the time and date of data acquisition, subject data regarding animal strain and parentage, study data such as the type of SPECT tracer or CT contrast agent used along with the route of administration, and perhaps textual information regarding other unique conditions under which the data were collected. For our purposes, we will refer to this extrinsic image data as metadata.

Image content is represented by the intrinsic properties of the image itself such as texture and regional structure [29], i.e., attributes that the human eye keys on to comprehend an image. Image content is extensive yet most data management systems today rely solely on extrinsic properties of the data to catalogue it for future use. Commercial relational database products and PACS reduce the ad-hoc nature of the cataloguing procedure by leveraging extrinsic image properties, but to effectively access the entire historical repository requires an ability to simultaneously engage both the extrinsic and intrinsic properties in a manner that is reasonably transparent to the user.

Fig. 8 shows examples of the wide variety of descriptive imagery that is generated at UTGSM and ORNL in support of our small animal research. Extracting, indexing, and managing the informational content of this data is of paramount importance.

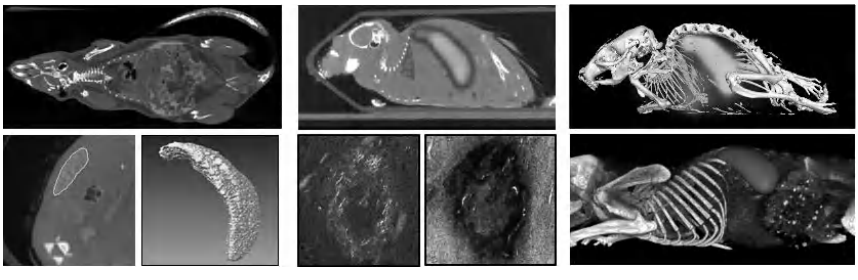


Fig. 8. Examples of the wide variety of descriptive imagery that is collected in support of small animal studies. This data was generated by UTGSM and ORNL to support amyloid research and includes (left) contrast-enhanced CT and both planar and volume segmentation of the spleen, (center) registered CT and SPECT data plus autoradiographs of splenic amyloid deposits imaged with ^{125}I-SAP, and (right) visualizations of amyloid burden generated from co-registered SPECT/CT (upper) and PET/CT data (lower).

Our CBIR technology is a library of C++ objects that create, manipulate, and access database tables as well as measure unique numerical descriptions of image content. Two main procedures of indexing and retrieval are carried out. The indexing process consists of measuring features from images, then using the features to build

tree structures that describe the image population. The indexing procedure begins by reading an image and extracting the image features from multiple regions of interest. For this research, these regions are generated using the 2D and 3D segmentation methods described earlier. As images are added to the table, indexing trees are generated. We use a modified approximate nearest-neighbor (ANN) algorithm to build and maintain the indexing trees [30]. In practice, multiple trees are generated, each representing a group of attributes associated with the image population, e.g., shape attributes and texture attributes, as shown in Fig. 9.

In retrieving, features are extracted from a query image and each indexing tree is searched to find the closest examples based on an L-norm distance metric. The final retrieval is a function of the intersection of these sets. A user can easily enable or disable these various attribute sets while performing queries, therefore making the system flexible and useful to a large population of end-users, each with differing search goals.

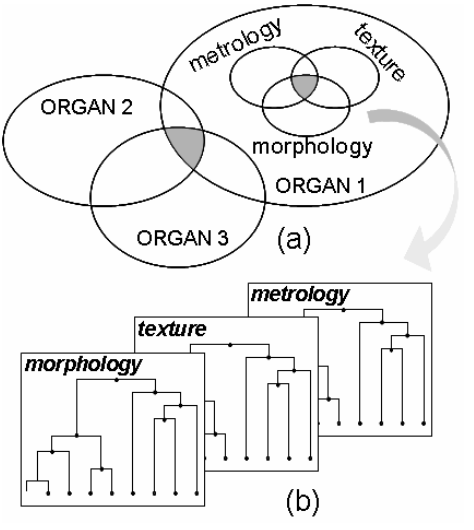


Fig. 9. Example feature indexing architecture. In (a) users select the descriptors of interest for each queried organ. The retrievals for each organ type (if multiple are selected) are returned in a list based on the Boolean intersection of the sets. In (b) the ANN indexing trees are represented that are accessed for each selected sub-group within an organ.

To demonstrate the initial application of our CBIR technology to this research, we have built a small CT cross-section databases for testing with existing feature constructs. The database contains 239 images from the microCT system of normal kidneys and polycystic kidney disease (PKD) cross sections. Each kidney image in the database is indexed individually such that one mouse is represented twice, once for each kidney. The kidney mask (i.e., segmentation) was automatically generated using the 2D PSAM algorithm of Gleason in [17]. The left image in Fig. 10 shows a random population of images from the database. The right image shows the result of querying with the left kidney of Mouse #32, which has PKD. Note that the first query result in the list contains Mouse #32's right kidney. The second and third results show Mouse # 33's right and left kidney respectively. Mouse #33 also has PKD. There were no other PKD mice in the database.

For this example the image texture attributes in the segmented kidney regions plays an important role in differentiating PKD from normal mice, although a total of 111 features representing intensity, texture, and shape where applied. Even though this was a simple example on a relatively small data set, it represents the inherent

investigative power that could be achieved by a fully developed informatics system that leverages visual content to perform queries on historical data populations. The incorporation of other study data (i.e., metadata) into the query process will extend this capability further and facilitate rapid investigations through hundreds of thousands of images while providing for the effective reuse of large historical repositories.

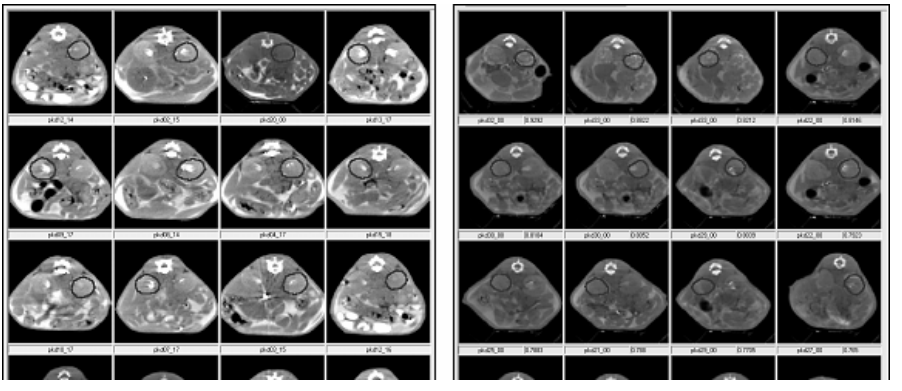


Fig. 10. Example database of normal and PKD kidneys generated using the micro CT imaging at ORNL. The left image shows a random retrieval of kidney image cross-sections. The right image shows a query with Mouse #32's left PKD kidney.

5 Conclusions

Micro imaging has become a predominant means for generating high-resolution, high-throughput, multi-modality small animal data. A result of this advancing capability is the generation of copious amounts of digital image and ancillary data that must be indexed and managed to retain it's usefulness in the preclinical study of genetics and disease. We have presented results of our preliminary research to segment and describe anatomic and functional structures, primarily soft organ tissue, collected from CT, SPECT, and PET imaging modalities. We have also shown an example of image indexing and retrieval on a small CT database of mice exhibiting polycystic kidney disease, demonstrating an ability to locate similar diseased mice in a population using kidney morphology and texture. The goal of this research is to develop a small animal information system that will improve the ability of preclinical researchers to make important biomedical discoveries by drastically increasing the achievable scope and size of their studies, therefore accelerating the translation of the preclinical investigation of disease to the clinical environment. More importantly, through the effective analysis and indexing of image content, the data system will provide researchers with unprecedented access to information in the imagery, which comprises the largest data component of the system.

References

1. Santini, S., *Exploratory Image Databases, Content-based Retrieval.* Academic Press, San Fransisco, CA, 2001.
2. Shyu, C., et al., *ASSERT, A physician-in-the-loop content-based image retrieval system for HRCT image databases.* Computer Vision and Image Understanding, July/August 1999. **75**(1/2): p. 111-132.
3. Tobin, K.W., et al., *Content-based Image Retrieval for Semiconductor Process Characterization.* Journal on Applied Signal Processing, 2002. **2002**(7).
4. Tobin, K.W., Karnowski, T.P., Arrowood, L.F., and Lakhani, F. *Field Test Results of an Automated Image Retrieval System*. in *12th Annual IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*. 2001. Munich, Germany.
5. Tobin, K.W., Bhaduri, B.L., Bright, E.A., Cheriyadat, A., Karnowski, T.P., Palathingal, P.J., Potok, T.E., Price, J.R., *Automated Feature Generation in Large-Scale Geospatial Libraries for Content-Based Indexing.* Journal of Photogrammetric Engineering and Remote Sensing, 2006. **72**(5).
6. Merlini, G. and V. Bellotti, *Molecular mechanisms of amyloidosis.* N Engl J Med, 2003. **349**(6): p. 583-96.
7. Bellotti, V., P. Mangione, and G. Merlini, *Review: immunoglobulin light chain amyloidosis--the archetype of structural and pathogenic variability.* J Struct Biol, 2000. **130**(2-3): p. 280-9.
8. Wall, J.S., et al. *Radioimaging of Primary (AL) Amyloidosis with an Amyloid-Reactive Monoclonal Antibody*. in *Amyloid and Amyloidosis: Proceedings of the Xth International Symposium on Amyloidosis*. 2005. Tours, France: CRC Press.
9. Schell, M., et al. *Prevention of AA-amyloidosis by active immunotherapy*. in *Amyloid and Amyloidosis: Proceedings of the IXth International Symposium on Amyloidosis*. 2001. Budapest, Hungary: David Apathy.
10. Wall, J.S., et al., *Quantitative high-resolution microradiographic imaging of amyloid deposits in a novel murine model of AA amyloidosis.* Amyloid, 2005. **12**(3): p. 149-56.
11. Gregor, J., Benson, T., Gleason, S., Paulus, P. *Support algorithms for x-ray micro-CT conebeam imaging*. in *Int. Conf. Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*. 2003. Saint Malo, France.
12. Gregor, J., Gleason, S., Paulus, M., Cates, J., *Fast Feldkamp reconstruction based on focus of attention and distributed computing.* Int. J. Imaging Systems and Technology, 2002. **12**: p. 229-234.
13. Benson, T., and Gregor, J., *Distributed iterative image reconstruction for micro-CT with ordered-subsets and focus of attention problem reduction.* J. X-Ray Systems and Technology, 2004. **12**: p. 231-240.
14. Benson, T., Gregor, J. *Modified simultaneous iterative reconstruction technique for faster parallel computation*. in *IEEE Medical Imaging Conf.* 2005. Puerto Rico.
15. Gregor, J., Gleason, S., Kennel, S., Paulus, M., Benson, T., Wall, J. *Approximate volumetric system models for microSPECT*. in *IEEE Medical Imaging Conf.* 2004. Rome, Italy.
16. Yap, J.T., et al., *Combined Clinical PET/CT and micro PET Small Animal Imaging.* IEEE NUCLEAR SYMPOSIUM 1082-3654, 2004. **5**: p. 2995-2998.
17. Gleason S, S.-S.H., Abidi M, Karakashian F, Morandi F, *A New Deformable Model for Analysis of X-ray CT Images in Preclinical Studies of Mice for Polycystic Kidney Disease.* IEEE Trans. on Medical Imaging, 2002. **21**.

18. Aykac, D., Price, J.R., Wall, J. *3D Segmentation of the Mouse Spleen in microCT via Active Contours*. in *Proceedings of the IEEE, Nuclear Science Symposium and Medical Imaging Conference*. 2005. Puerto Rico.
19. Wall, J.S., Kennel, S.J., Paulus, M.J. , Gleason S.S. , Gregor, J. , Baba, J. , Schell, M. , Richey, T. , O'Nuallain, B. , Donnell, R. , Hawkins, P.N. , Weiss, D.T. , and Solomon, A. , *Quantitative high-resolution microradiographic imaging of amyloid deposits in a novel murine model of AA amyloidosis.* Amyloid, 2005. **12**(3): p. 149-156.
20. Chan, T.a.V., L., *Active contours without edges.* IEEE Transactions on Image Processing, 2001. **10**(2): p. 266-277.
21. Price, J., Aykac, D., Wall, J. *A 3D level sets method for segmenting the mouse spleen and follicles in volumetric microCT images*. in *IEEE Engineering in Medicine and Biology Conference (EMBC)*. 2006.
22. Besl, P.a.M., N., *A method for registration of 3-D shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(2): p. 239-256.
23. Micheals, R.a.B., T. *Increasing robustness in self-localization and pose estimation*. in *Proceedings of the 1999 Mobile Robots XIV*. 1999: SPIE.
24. Muller, H., et al., *A Review of Content-Based Image Retrieval Systems in Medical Applications - Clinical Benets and Future Directions.* International Journal of Medical Informatics, 2004. **73**(1): p. 1-23.
25. Chu, W.W., Cardenas, A.F., Taira, R.K., *Knowledge-based image retrieval with spatial and temporal constructs.* IEEE Trans. on Knowledge and Data Engineering, 1998. **10**(6): p. 872-888.
26. Bueno, J.M., et al. *How to Add Content-based Image Retrieval Capability in a PACS*. in *The 15 th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002)*. 2002.
27. Schultz, C.P., et al., *Molecular Imaging Portal: New Development IT Platform for Imaging, Nonimaging and Genomics.* Molecular Imaging, 2005. **4**(4): p. 71-77.
28. Le Bozec, C., Zapletal, E., Jaulent, M., Heudes, D., Degoulet, P. *Towards content-based image retrieval in a HIS-integrated PACS*. in *Proceedings AIMA Symposium*. 2000.
29. Smeulders, A.W.M., et al., *Content-Based Image Retrieval at the End of the Early Years.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. **22**(12): p. 1349-1380.
30. Arya, S., et al., *An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions.* Proc. of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 1994: p. 573-582.

Segmentation-Based Registration of Organs in Intraoperative Video Sequences

James Goddard[1,*], Timothy Gee[1], Hengliang Wang[2], and Alexander M. Gorbach[3]

[1] Image Science and Machine Vision Group, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831
goddardjsjr@ornl.gov, geetf@ornl.gov
[2] Naval Medical Research Center, Silver Spring, Maryland 20910
wangheng@mail.nih.gov
[3] National Institutes of Health, Bethesda, Maryland 20892
gorbach@helix.nih.gov

Abstract. Intraoperative optical imaging of exposed organs in visible, near-infrared, and infrared (IR) wavelengths in the body has the potential to be useful for real-time assessment of organ viability and image guidance during surgical intervention. However, the motion of the internal organs presents significant challenges for fast analysis of recorded 2D video sequences. The movement observed during surgery, due to respiration, cardiac motion, blood flow, and mechanical shift accompanying the surgical intervention, causes organ reflection in the image sequence, making optical measurements for further analysis challenging. Correcting alignment is difficult in that the motion is not uniform over the image. This paper describes a Canny edge-based method for segmentation of the specific organ or region under study, along with a moment-based registration method for the segmented region. Experimental results are provided for a set of intraoperative IR image sequences.

1 Introduction

Intraoperative optical images of exposed organs in visible, near-infrared, and infrared (IR) wavelengths in the body have the potential to be useful for real-time, non-invasive assessment of organ viability and image guidance during surgical intervention. Relatively low cost, non-invasiveness and good specificity for physiological changes make optical imaging desirable for a variety of surgical procedures [1-4]. However, the motion of the internal organs presents significant challenge for real-time data analysis of recorded two-dimensional video sequences.

The movement of the kidney, liver, urethra, etc. observed during surgery due to respiration, cardiac motion, and blood flow, can be between 15 and 20 mm. In addition to mechanical shift of the tissue from the surgical intervention, all of these cause organ reflection in the image sequence, making optical measurements for further analysis challenging. Correcting alignment is difficult in that the motion is not

* Prepared by Oak Ridge National Laboratory managed by UT-Battelle, LLC, for the U.S.Department of Energy under Contract No. DE-AC05-00OR22725.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 835 – 842, 2006.
© Springer-Verlag Berlin Heidelberg 2006

uniform over the image. Also, other artifacts, such as glare caused by illumination from surgical lights reflecting off of wet tissue surfaces (in the case of visible wavelength imaging) and evaporative cooling (in IR imaging), add random noise to the imagery, do not allow clear visualization of internal fiducial markers, and, therefore, make the task more difficult.

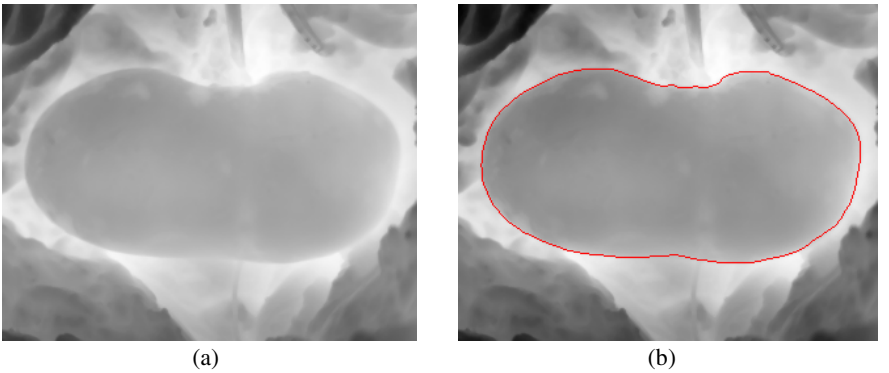
This paper describes a Canny edge-based method for segmentation of the specific organ or region under study, along with a shape moment-based registration method for the segmented region. Experimental results are provided for a set of intraoperative IR image sequences of the kidney. All organs generally exhibit 3D motions, in addition to having 3D volumes. Moreover, these organs and tissues are surrounded by deformable surfaces. While medical image registration has been well-researched [5-12], the registration to two-dimensional IR surgical image sequences to non-rigid 3D structures has not been extensively studied.

2 Image Registration Method

Image registration for an IR video is accomplished in two stages. The organ or region of interest is first automatically segmented throughout the video sequence. This process is accomplished by a combination of edge extraction and morphological operations that form a closed boundary of the organ. Then a shape-based registration algorithm is applied to the segmented organ to estimate the alignment error relative to a base image in the image sequence. These steps are detailed below.

2.1 Image Sequence Collection

IR imaging sequences of porcine kidney were collected to evaluate renal ischemia, particularly focal perfusion deficits. Laparotomy with renal mobilization, hilar dissection, and identification of segmental branches were performed in six porcine models. An IR camera (3-5 μ wavelengths) was directed at two exposed kidneys from 30 cm distance. Images (320x256x14 bits) were collected at the rate of one per second with 300-600 images per sequence.



(a) (b)

Fig. 1. IR image of pig kidney from video sequence shows varying contrast at kidney boundary. (a) shows original image while (b) shows an overlay of the kidney contour.

These image sequences have been used for development and testing of the registration method. Figure 1(a) shows a baseline IR image of a pig kidney collected during this evaluation. An outline of the kidney edge generated by the segmentation method is also shown in Figure 1(b). At this boundary, the contrast with the surrounding tissue varies from good to just visible.

As a preprocessing step, 2D median filtering is applied to remove fixed pattern noise present after calibration of the IR images. The improved result is shown below in Figure 2.

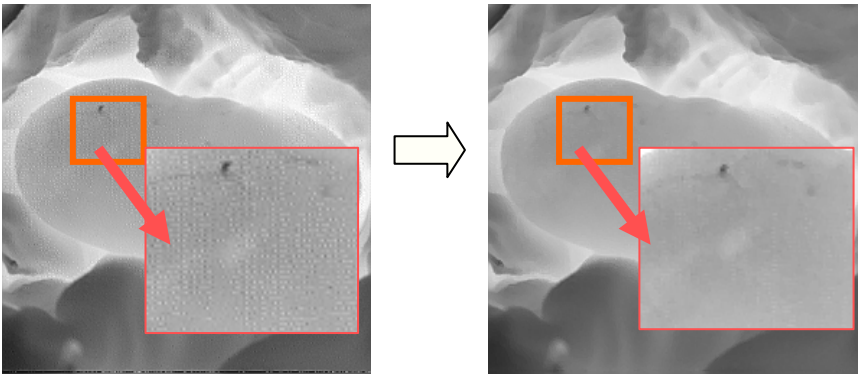


Fig. 2. 2D median filtering with a 3 by 3 window is applied to each input image to reduce fixed pattern noise as shown in the image insets

2.2 Organ Segmentation

In the acquired video, the kidney is moving in the foreground with the pig's other body organs in the background. The image content at the boundary of the kidney is changing slightly since the kidney is not fixed to the background at its edges. Therefore to achieve good video registration it is necessary to segment the organ from the background and consider it independently.

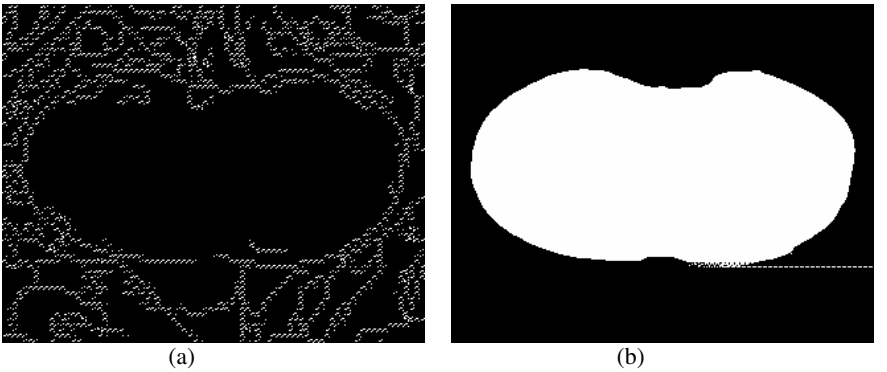
In much of the imagery, the overall contrast between the kidney and the overall background is useful. With this in mind, we can achieve segmentation using a Canny [13] edge finder followed by some morphological operations. Figure 3(a) shows the result following the Canny edge detection.

From the Canny edge detection, a binary image is obtained, in which edge pixels have the value 1, and other pixels are 0. This image is the input of morphological operations performed to acquire the kidney as an object. The first operation is *bridging* which joins pixels of value 1 that are separated by only one zero pixel. This fills gaps in the kidney contour. In some cases, it was necessary to dilate the edge image before bridging to allow bigger gaps to be filled. In these cases the edge was thinned afterward [14].

After bridging, an image fill operation is performed that starts at the center of the kidney and inserts values of 1 until stopped by a continuous boundary of 1s. If the kidney boundary was extracted without gaps, then this operation creates a binary object describing the kidney. If there are gaps in the boundary, then the fill operation

will fill a much greater area than the size of the kidney and will likely fill the entire image. This case is easily detected, leading us to use the kidney object obtained from the previous frame. However, if good contrast is available, the above procedure obtains the boundary with a high success rate.

After the image fill, binary connectivity analysis is performed so that the kidney can be identified as an object. An operation is then performed to fill any holes in this object, so that the kidney is completely covered. Additional morphological operations such as *opening* (erosion followed by dilation) are necessary to remove small appendages of the kidney that occur from background edges that are adjacent to the kidney. The result is a single object that only describes the kidney shape and location, as shown in Figure 3(b).



(a) (b)

Fig. 3. Canny edge (a) and segmented kidney (b) image

2.3 Registration

In the video that we have observed, the kidney is well-approximated as a rigid body. During the organ's movements, the background changes, and the kidney's location and rotation in the background changes, but the kidney's shape does not change significantly. Therefore, it's possible to perform registration using the object location and orientation that was extracted as described above. We have experimented with the use of tracking intensity textures or landmarks within the kidney region, but we have found the most accurate and robust results from tracking the kidney's contour.

As illustrated in Figure 4, by using first and second-order moments [14] of the obtained binary object, we can describe its location and orientation. The first-order moments, or centroid, define the x and y coordinates of the object's center of mass. The second-order central moments can be used to define the object's axis of maximum elongation. Here, the shape is modeled as an ellipse, and we find the angle of the major axis of an ellipse with the same second-order central moments as the binarized kidney.

The segmented kidney object in the first frame of the image sequence is used as a reference. The registration of each following frame is then defined by the translation needed to shift the centroid and align the major axis to match the reference. This defines a homogenous transform to map intensity values from image points in the

current frame to the corresponding locations of the reference image. Since subpixel locations are needed from the current frame, interpolation is required. We have found bi-cubic interpolation to be sufficient.

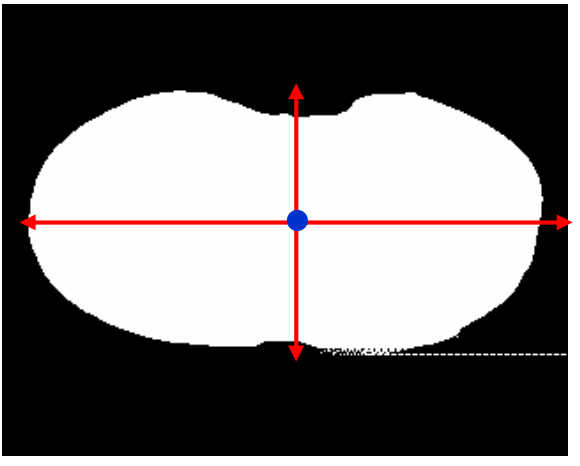


Fig. 4. Illustration of first and second-order moments in the segmented kidney object

3 Experimental Results

The registration method described in section 2 has been applied to five sequences of intraoperative IR pig kidney images. The resulting registered image sets have been evaluated using both a cross-correlation and a frequency-based measure. Typical results from each of these measures are given below.

3.1 Cross-Correlation Registration Performance Measure

The normalized cross-correlation value S is provided in Eq. (1), which allows computing pixel-wise cross-correlation coefficients between the pixel intensities of the two images, normalized by the square root of their autocorrelations.

$$S(d_x,d_y)=\frac{\sum_{(x,y)\in C}[I_1(x+d_x,y+d_y)-\overline{I_1(x+d_x,y+d_y)}]*[I_2(x,y)-\overline{I_2(x,y)}]}{\sqrt{\sum_{(x,y)\in C}[I_1(x+d_x,y+d_y)-\overline{I_1(x+d_x,y+d_y)}]^2 * \sum_{(x,y)\in C}[I_2(x,y)-\overline{I_2(x,y)}]^2}} \quad (1)$$

where I_1, I_2 are intensities of the two images respectively, C is the mask of the image, d_x, d_y are number of shifting pixels in x and y directions. The mask is such that the cross-correlation is performed only on interior pixels of the kidney.

When S reaches the maximum value, $d_x = d_{xmax}$, $d_y = d_{ymax}$. Then $D = |d_{xmax}| + |d_{ymax}|$, is defined as the number of shifting pixels. Ideally, for the two perfect registered images, $D = 0$. The smaller D value is, the less the motion affects the image sequence. The D values between each adjacent renal IR images are calculated to test the performance of image registration.

The numbers of shifting pixels (*D* value) for each image in the sequence before and after registration were calculated (see Figure 5). The mean *D* values for whole sequence are 1.8 for non-registered image sequence vs. 0.7 for registered image sequence. While the actual registration is at a subpixel level, the reduction in integer correlation shifts of the kidney interior proves the stabilization in kidney position for this sequence.

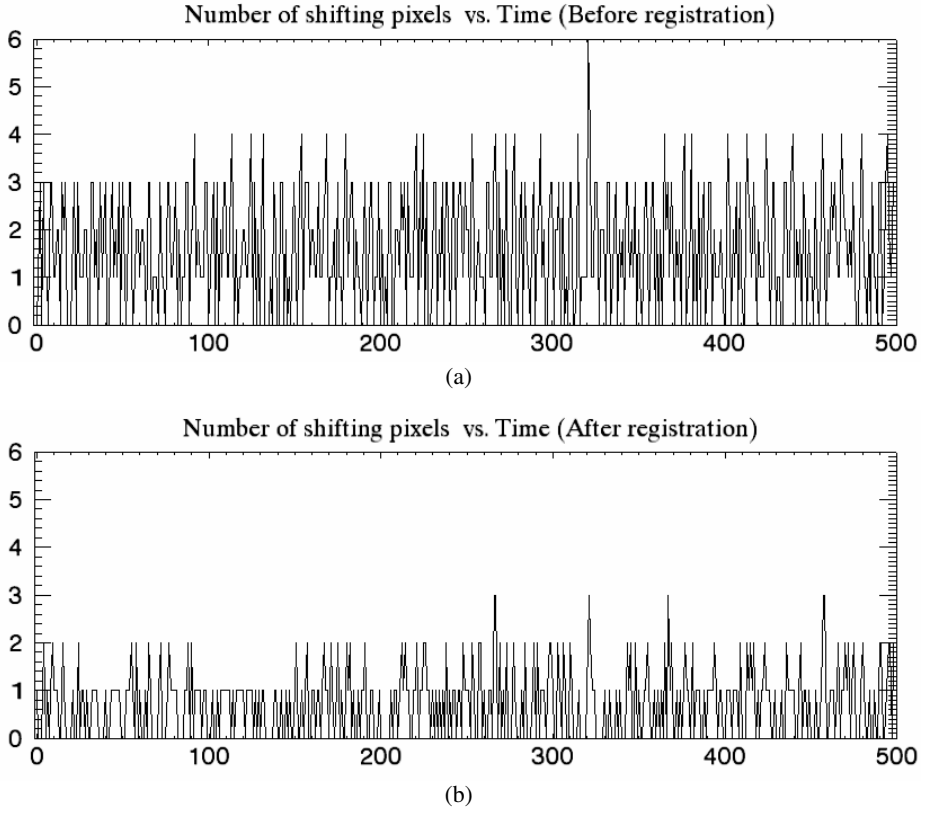
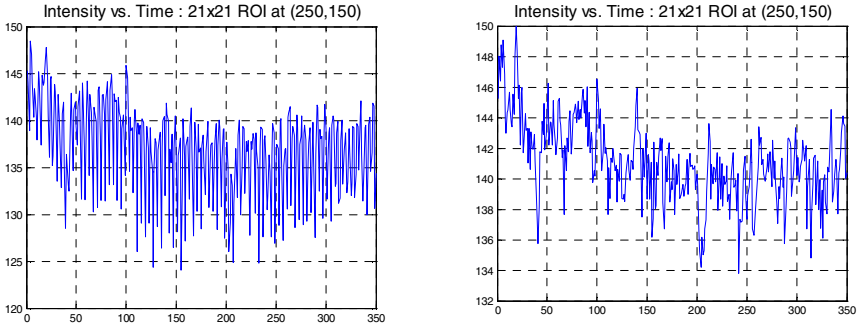


Fig. 5. Comparison of number of shifting pixels (D values) between two sequences of 500 images each: before (a) and after (b) registration

3.2 Frequency Domain Registration Performance Measure

A frequency domain analysis to measure alignment was performed on an interior image region of the kidney. For this test, a 21 by 21 pixel region was used. The mean value for this region was calculated for each image in the sequence. Figure 6 shows the before and after registration for one such typical region over the sequence. The time sequence before registration is seen to have higher noise than after registration.



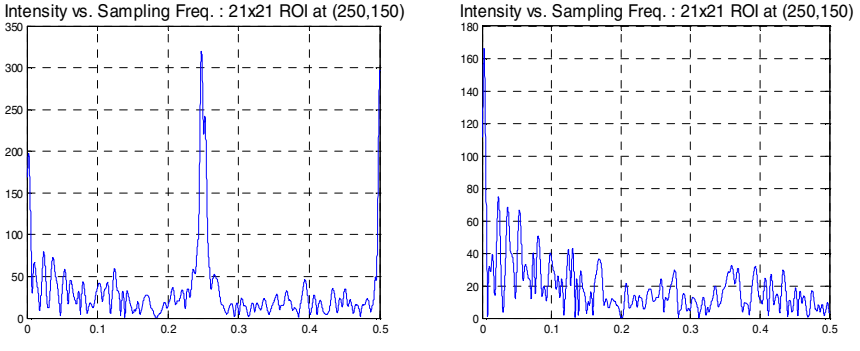
(a) (b)

Fig. 6. Time domain plot of 21 by 21 pixel region mean value for all images in sequence before registration (a) and after registration (b). A significant reduction in noise level can be seen after registration.

The discrete Fourier transform (DFT), f_k for $k = 0 \ldots N-1$, is calculated for the time sequence t_n for $n = 0 \ldots N-1$,

$$f_k = \sum_{n=0}^{N-1} t_n e^{-\frac{2\pi i n k}{N}} . \quad (2)$$

The frequency domain plot of the DFT gives further insight into the registration effectiveness. Figure 7(a) shows a large frequency component located approximately 0.25 of Nyquist. This component has a period of 4 seconds where the sampling frequency is 1/sec. and corresponds to the respiration rate. Figure 7(b) shows that this component is reduced to background noise levels after registration. The registration method removed the motion corresponding to the respiration.



(a) (b)

Fig. 7. Discrete Fourier transform of 21 by 21 pixel region mean value as shown in Fig. 6 before registration (a) and after registration (b). DFT shows elimination of significant frequency band (due to respiration-induced movement) after registration.

4 Conclusions

An image sequence registration method has been proposed that provides localized registration for specific regions and organs in intraoperative 2D medical image sequences. The IR image sets presented challenges due to thermal noise and low contrast. Successful segmentation of the kidney in these images was achieved using this method and good overall registration was applied to these regions. Performance results were presented that demonstrated quantitatively the improvement in the image alignment. This method can be applied not only to IR imagery but to visible optical images as well. Additional work is needed to characterize the thermal noise and to reduce the effects on the registration and measured thermalsignals. This method is capable of being implemented in real-time using available hardware and software.

References

1. Gorbach A, Simonton D, Hale DA, Swanson SJ, Kirk AD. Objective, real-time, intraoperative assessment of renal perfusion using infrared imaging. Am J Transplant. 2003 Aug;3(8):988-93.
2. Gorbach AM, Heiss JD, Kopylev L, Oldfield EH. Intraoperative infrared imaging of brain tumors. J Neurosurg. 2004 Dec;101(6):960-9.
3. Reuthebuch O, Haussler A, Genoni M, Tavakoli R, Odavic D, Kadner A, Turina M. Novadaq SPY: intraoperative quality assessment in off-pump coronary artery bypass grafting. Chest. 2004 Feb;125(2):418-24.
4. Mitsuta H, Ohdan H, Fudaba Y, Irei T, Tashiro H, Itamoto T, Asahara T. Near-infrared spectroscopic analysis of hemodynamics and mitochondrial redox in right lobe grafts in living-donor liver transplantation. Am J Transplant. 2006 Apr;6(4):797-805.
5. Bardera, M. Feixas, I. Boada, “Normalized similarity measures for medical image registration,” SPIE Medical Imaging, Proceedings of SPIE, Vol. 5370, 2004.
6. J. Pluim, J. Maintz, M. Viergever, “Mutual information based registration of medical images: a survey,” IEEE Transactions on Medical Imaging, Vol.22, No. 8, 2003.
7. J. Ruiz-Alzola, E. Suarez, C. Alberola-Lopez, S.K. Warfield, C.-F Westin, “Geostatistical Medical Image Registration,” MICCAI 2003, LNCS 2879, Springer-Verlag, 2003.
8. J. Maintz, P. van den Elsen, M. Viergever, “3D multimodality image registration using morphological tools,” Image and Vision Computing, Vol. 19, Elsevier, 2001.
9. C. Sorzano, P. Thevenaz, and M. Unser, “Elastic Registration of Biological Images UsingVector-Spline Regularization,” IEEE Transactions on Biomedical Engineering, Vol. 52, No. 4, pp 652-663, April, 2005.
10. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in DARPA Image Understanding Workshop. DARPA, 1981, pp. 121–130.
11. Maes, F., “Medical image registration using mutual information,” *Proceedings of the IEEE*, v 91, n 10, October, 2003, pp 1699-1721.
12. C. Davatzikos, J. L. Prince, and R. N. Bryan, “Image Registration Based on Boundary Mapping”, IEEE Transactions on Medical Imaging, Vol. 15, No. 1, Feb. 1996.
13. Canny, John, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI-8, No. 6, 1986, pp. 679-698.
14. Haralick, Robert M., and Linda G. Shapiro, *Computer and Robot Vision*, Volume I, Addison-Wesley, 1992.

On Asymmetric Classifier Training for Detector Cascades*

Timothy F. Gee

Oak Ridge National Laboratory

Abstract. This paper examines the Asymmetric AdaBoost algorithm introduced by Viola and Jones for cascaded face detection. The Viola and Jones face detector uses cascaded classifiers to successively filter, or reject, non-faces. In this approach most non-faces are easily rejected by the earlier classifiers in the cascade, thus reducing the overall number of computations. This requires earlier cascade classifiers to very seldomly reject true instances of faces. To reflect this training goal, Viola and Jones introduce a weighting parameter for AdaBoost iterations and show it enforces a desirable bound. During their implementation, a modification to the proposed weighting was introduced, while enforcing the same bound. The goal of this paper is to examine their asymmetric weighting by putting AdaBoost in the form of Additive Regression as was done by Friedman, Hastie, and Tibshirani. The author believes this helps to explain the approach and adds another connection between AdaBoost and Additive Regression.

1 Introduction

The Viola and Jones face detector [1] was a very significant change in approach to face detection. Based on this method, several researchers have developed systems to detect faces of different pose and other objects [2,3,4]. There is a considerable computational load on face detectors that rely on only the cues obtained from still, gray images. For such an algorithm to reliably detect faces of different scales, it must perform many scans through each image, parsing the image by small regions, using a small spatial increment. Viola and Jones report their algorithm having to classify 50,000 regions for a typical image. Considering that the number of regions that contain a face is likely to be on the order of 1-10, then it becomes obvious that it is important to perform rejections on most image regions as quickly as possible, while taking great care to avoid carelessly rejecting face regions. The cascaded face detector is very fast because of its use of Haar-wavelet-like classifiers that are efficiently calculated from integral images and the novel method of placing classifiers in a cascade, such that non-faces are usually rejected quickly.

In the original work, Viola and Jones leave the face and non-face samples equally weighted, and the number of training samples are on the same order of

* Document prepared by Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 843–850, 2006.
© Springer-Verlag Berlin Heidelberg 2006

magnitude. They feed the system with 4,916 positive face samples, the greatest majority of which should pass completely through the cascade. At each layer they incorporate 10,000 non-face samples, so the number of positive samples is only a factor of two less than the number of negative samples. At run-time, the detector will scale the size of the features to find faces of various sizes, but the base size used for training in such systems is typically about 24x24.

In the design of the cascade, the designer has in mind a particular goal for the overall system's detection rate, D, and false acceptance rate, A. These overall goals can be used to set goals for the detection and false accept rates of individual stages. This is because the overall performance rates are obtained from the product of the stage performance rates. That is

$$D = \prod_{c=1}^{C} d_c \tag{1}$$

and

$$A = \prod_{c=1}^{C} a_c \ , \tag{2}$$

where d_c is the detection rate for classifier c, and a_c is the false accept rate for classifier c in the cascade. With this is mind, Viola and Jones state that there is maximum acceptable a_c and minimum acceptable d_c for each cascade stage c, and they continue to add features to be evaluated in a given stage until an acceptable false accept rate a_c is achieved. While this is being performed, at each iteration, the discriminant threshold θ_c for the stage is adjusted to ensure that the detection rate d_c is high enough.

Viola and Jones use the steps described in Algorithm 1 to develop each stage classifier in the detector cascade. This is the Discrete AdaBoost algorithm with the introduced θ threshold added to the discriminate function. In their case, each weak classifier $f_m(x)$ is a test applied to a single Haar-wavelet-like feature that has been evaluated on the given training region.

2 Asymmetric Bias Weighting

As discussed above, the original form of the Viola-Jones face detector moved to a different point on the ROC curve by modifying the threshold, θ_c. However, this is clearly non-optimal, since the threshold is determined after weak classifiers have been chosen. In the later paper [5] by Viola and Jones, they intend to handle this my multiplying the weight of face and non-face training samples by a bias term. Initially they advocate multiplying the weight of each sample n by

$$\exp\left(y_n \ln \sqrt{k}\right) \ , \tag{3}$$

where y_n is the desired or true classification as used in 1. For this application, y_n is 1 for faces and -1 for non-faces. k is a weighting that is used to increase

Algorithm 1. Steps for Discrete AdaBoost with threshold θ

Assume training samples $\{(x_n, y_n)\}_{n=1}^{N}$
with $x \in \mathcal{R}^k$, and $y \in \{-1, 1\}$

Perform the following steps
1) Initialize observation weights
   $w_n = 1/N$ for $n = 1, 2, \ldots N$.
2) For $m = 1$ to M:
   a) Fit the classifier $f_m(x) \in \{-1, 1\}$
      to the training data weighted according to w_n.
   b) Compute the error as $\epsilon_m = \frac{\sum_{n=1}^{N} w_n I(y_n \neq f_m(x_n))}{\sum_{n=1}^{N} w_n}$.
   c) Compute $\alpha_m = \ln\left(\frac{(1-\epsilon_m)}{\epsilon_m}\right)$.
      (Note: must have $\epsilon_m < 0.5$)
   d) Set $w_i \leftarrow w_i \exp[\alpha_m I(y_n \neq f_m(x_n))]$
      for $n = 1, 2, \ldots N$.
3) Final classifier is $g(x) = \text{sgn}\left[\sum_{m=1}^{M} \alpha_m f_m(x) > \theta\right]$.

the detection rate for the resulting strong classifier. The motivation is to make the false rejections cost k times more than false detections. That is

$$cost(n) = \begin{cases} \sqrt{k} & \text{if } y_n = 1 \text{ and } g(x_n) = -1 \\ \sqrt{\frac{1}{k}} & \text{if } y_n = -1 \text{ and } g(x_n) = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where again, y_n is the true classification, and $g(x_n)$ is the estimated classification.

In the Viola and Jones paper, it is shown that the solution's training error bound achieves the desired cost ratio by weighting the original importance of the training samples using k. However, through empirical testing, it became evident that the weighting only affects the weak classifier in the first AdaBoost iteration. Then the authors decide to factor the term into parts and apply the Mth root of the weighting k during each of M rounds of AdaBoost training. This enforces the same bound, the asymmetric weighting was found to take effect throughout the training of the ensemble, and improved results were obtained in an illustrative two-dimensional classification example. Also, beyond this two-dimensional case, Viola and Jones train their face detector cascade using asymmetric AdaBoost and compare it to that obtained using symmetric AdaBoost and found that the new ROC curve points indicated higher detection rates for given false acceptance

Algorithm 2. Forward Stage-wise Additive Modeling [8]

Assume training samples $\left\{(x_i, y_i)\right\}_{i=1}^{N}$ with $x \in \mathcal{R}^k$, and $y \in \{-1, 1\}$

Perform the following steps
1) Initialize $f_0(x) = 0$.
2) For $m = 1$ to M:
   a) Compute $(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L\big(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)\big)$,
      where $b()$ is a parameterized basis function,
      and $L()$ is a norm.
   b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.
3) Final classifier is g(x) = sgn $[f_M(x)]$.

rates. It is worth noting that the Asymmetric AdaBoost approach has been repeated by others for face detection and other applications [6].

3 Comparison to Additive Regression

Based on the work of Friedman, Hastie, and Tibshirani [7] AdaBoost iterations can be shown to be equivalent to Additive Regression when a specific loss function is used. The steps for achieving a classifier from Additive Regression are shown in Algorithm 2.

It is shown in [7] that by using the loss function

$$L(y, f(x)) = \exp(-yf(x)) \,, \tag{5}$$

with Additive Regression, it is equivalent to Discrete AdaBoost. Here it is shown that the Asymmetric AdaBoost presented by Viola and Jones is equivalent to creating a classifier from additive regression using a loss function of:

$$L_m(y, f(x)) = \exp(-yf(x)) \exp\left(y \sum_{j=1}^{m} \frac{1}{M} \ln\sqrt{k}\right) \,, \tag{6}$$

where m is the number of the current additive iteration. The proof is as follows. At each iteration m, the parameters for additive regression are chosen by

$$(\beta_m, b_m) = \arg\min_{\beta,b} \sum_{i=1}^{N} \exp\left[-y_i \left(f_{m-1}(x_i) + \beta b(x_i)\right)\right] \exp\left(y_i \sum_{j=1}^{m} \frac{1}{M} \ln\sqrt{k}\right) . \tag{7}$$

This is equivalent to

$$(\beta_m, b_m) = \arg\min_{\beta,b} \sum_{i=1}^{N} \omega_i^{(m)} \exp\left[-y_i \beta b(x_i)\right], \tag{8}$$

where

$$\omega_i^{(m)} = \exp\left[-y_i f_{m-1}(x_i)\right] \exp\left(y_i \sum_{j=1}^{m} \frac{1}{M} \ln\sqrt{k}\right). \tag{9}$$

The ω_i^m correspond to the sample weights used in the AdaBoost algorithm. This derivation is identical to that shown in [7] with the insertion of the Asymmetric weighting term. As in [7] it can be shown that β and b are chosen identically to that of the AdaBoost algorithm. We can determine the update rule for sample weights by examining

$$\omega_i^{(m+1)} = \exp\left[-y_i f_m(x)\right] \exp\left(y_i \sum_{j=1}^{m+1} \frac{1}{M} \ln\sqrt{k}\right) \tag{10}$$

$$= \exp\left[-y_i \left(f_{m-1}(x_i) + \beta G(x_i)\right)\right] \exp\left(y_i \sum_{j=1}^{m} \frac{1}{M} \ln\sqrt{k}\right) \exp\left(\frac{1}{M} y_i s \ln\sqrt{k}\right) \tag{11}$$

$$= \omega_i^{(m)} \exp\left[-y_i \beta G(x_i)\right] \exp\left(\frac{1}{M} y_i \ln\sqrt{k}\right). \tag{12}$$

Thus, the sample weights are updated identically to what was proposed in [5].

4 Objective Function

Again, the work of [7] is followed closely in order to determine why it is desirable to use these iterations to find a strong classifier. The components of the final strong classifier are chosen in an attempt to minimize the expected value of the final loss function. Although, the loss function $L_m()$ changes with each iteration, at the final iteration, M, the loss function is

$$L_M(y, f(x)) = \exp\left[-y f(x)\right] \exp\left(y \sum_{j=1}^{M} \frac{1}{M} \ln\sqrt{k}\right) \tag{13}$$

$$= \exp\left[-y f(x)\right] \exp\left(y \ln\sqrt{k}\right). \tag{14}$$

The objective function to be minimized is then

$$J(F) = E\left[\exp\left[-y F(x)\right] \exp\left(y \ln\sqrt{k}\right)\right] \tag{15}$$

$$= P(y=1|x) \exp\left[-F(x)\right] \exp\left(\ln\sqrt{k}\right) + P(y=-1|x) \exp\left[F(x)\right] \exp\left(-\ln\sqrt{k}\right). \tag{16}$$

By taking the derivative of $J(F)$ with respect to F and setting it to 0, we obtain

$$F(x) = \frac{1}{2} \ln \left[\frac{kP(y=1|x)}{P(y=-1|x)} \right] . \tag{17}$$

AdaBoost uses the sign of this function to determine classification. It can be seen that the probability of choosing class $y = 1$ is multiplied by k.

5 Experimental Results

Although this paper is mainly a mathematical reinforcement of the paper by Viola and Jones, experimental results are given to provide some further quantification of the method. In a contrived example, 1000 samples are randomly obtained from each of two class distributions. Each class is distributed by a two-dimensional Gaussian uncorrelated in x and y and with standard deviations $\sigma_x = 2$ and $\sigma_y = 0.5$. The mean of class 1 is $\mu_1 = (0, -1)$, and the mean of class 2 is $\mu_2 = (0, 1)$. The training samples are shown in Figure 1.

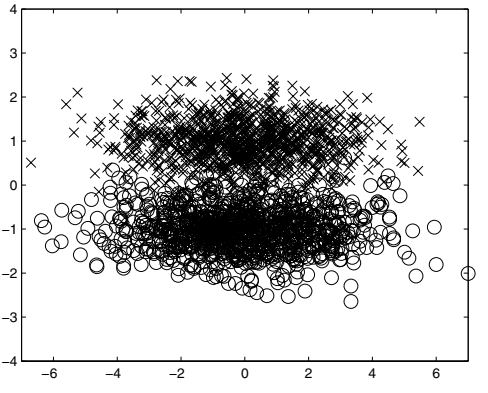


Fig. 1. Training data for two classes in example 1

In this hypothetical case, which is indicative of performance in near-ideal situations, $M = 12$ iterations were performed, with the asymmetric weighting $k = 10$. The weak classifiers are obtained by simply choosing a feature and threshold that minimizes the current weighted error. The obtained results are shown in Table 1.

In the second example, the same class distributions and class separation were used, but the Guassians were rotated 45 degrees with respect to the feature axes. The mean of class 1 is $\mu_1 = (-1/\sqrt{2}, -1\sqrt{2})$, and the mean of class 2 is $\mu_2 = (1/\sqrt{2}, 1/\sqrt{2})$. The training samples are shown in Figure 2. This example is indicative of lesser performance using the same method. $M = 12$ iterations were performed, with the asymmetric weighting $k = 10$. The obtained results are shown in Table 2.

Table 1. Example 1 performance in percent for strong classifier of 12 iterations, $k = 10$

	True Detections	False Detections	True Rejections	False Rejections
Training	99.6	7.4	92.6	0.4
Testing	99.5	6.0	94.0	0.5

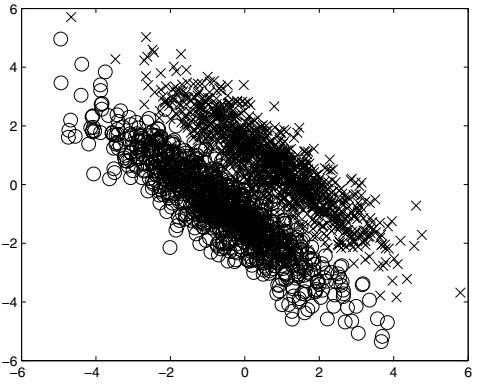


Fig. 2. Training data for two classes in example 2

Table 2. Example 2 performance in percent for strong classifier of 12 iterations, $k = 10$

	True Detections	False Detections	True Rejections	False Rejections
Training Data	98.6	34.0	66.0	1.4
Testing Data	97.7	37.0	63.0	2.3

6 Conclusions

This paper has shown how Asymmetric AdaBoost can be represented as a form of Additive Regression, given the correct loss function. The provides clarification for how this approach works, as well as increasing the general understanding of AdaBoost. Examples with simulated classes were used to show how the approach influences the training goal. The ability of the weak classifiers to precisely partition the sample space can distort the effects of the bias weighting. However, it can be easily seen from the examples that a significant decision bias is achieved.

References

1. Viola, P., Jones, M.: Robust real-time object detection. In: ICCV Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling, IEEE (2001)
2. Li, S.Z., Zhu, L., Zhang, Z., Zhang, H.: Learning to detect multi-view faces in real-time. In: Proceedings of the IEEE International Conference on Development and Learning. (2002)
3. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proceedings of the IEEE International Conference onImage Processing. Volume 1. (2002) 900–903
4. Wu, J., Mullin, M.D., Rehg, J.M.: Linear asymmetric classifier for cascade detectors. In: ICML '05: Proceedings of the 22nd international conference on Machine learning, New York, NY, USA, ACM Press (2005) 988–995
5. Viola, P., Jones, M.: Fast and robust classification using asymmetric adaboost and a detector cascade. NIPS 14 (2002)
6. Healy, M., Ravindran, S., Anderson, D.: Effects of varying parameters in asymmetric adaboost on the accuracy of a cascade audio classifier. In: SoutheastCon, 2004. Proceedings. IEEE. (2004) 169–172
7. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting (revised with discussions). The Annals of Statistics **28** (2000) 337–407
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer-Verlag (2001)

Active Stabilization of Images Acquired on a Walking Robotic Platform

Xander Twombly, Richard Boyle, and Silvano Colombano

NASA Ames Research Center, Moffett Field, CA 94035, USA
xtwombly@mail.arc.nasa.gov, Richard.Boyle@nasa.gov,
Silvano.P.Colombano@nasa.gov

Abstract. To increase the quality of scientific data collected from autonomous mobile agents such as rovers and walking robotic devices, biological methods can be mimicked for better navigation and balance control of both the agent itself and the manipulation of scientific instruments. Drawing on the design of the neuro-vestibular control system, the EarBot controller is designed to stabilize a multi-axis camera system mounted atop a moving agent. An eight-legged robot called the SCORPION, designed to navigate and explore rough terrain considered inhospitable to wheeled rovers, is used as the testbed to analyze the EarBot's functionality and behavior. Eventually, the EarBot will be used to control the balance the robot itself through expanded modelling of the vestibulo-motor control loops used in postural control. This paper presents the theoretical concepts and initial controller implementations for stabilizing the camera during walking motion of the SCORPION.

1 Introduction

The initial goal for the EarBot controller is to maintain a stabilized image from a camera system mounted on an autonomous, freely moving agent, primarily through mimicking the vestibulo-ocular response (VOR). During head rotations, the VOR maintains a steady image to the retina via a counter-movement of the eyes triggered from stimulation of the vestibular end organs located in the inner ear. The vestibular system is composed of 3 nearly orthogonal semicircular canals and 2 otolith organs (utricle and saccule) that detect angular and gravitational acceleration, respectively. Such a system is capable of characterizing 6-degree-of-freedom motion of the head, and is used to calculate the inertial motion and head location in space. The control circuit used to generate the VOR is based on a 3 neuron layer reflex arc, and uses the integrated acceleration signals to generate a controller driven primarily with velocity and rotation position information to maintain a steady fixation of the eye during head movements. The VOR can maintain a steady fixation using primarily tilt and yaw movements, but has only moderate ability to generate roll movements within the ocular system. Subsequently the VOR can maintain fixation on a point in space, but can only correct for small variations in changes due to roll of the head.

A second biological system of interest is the vestibulo-colic (VCR) reflex whose function is to move the neck to maintain head orientation in space; similar to the way

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 851 – 860, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the VOR is used to maintain eye location. A primary purpose of the VCR is to maintain the alignment of the head with the gravitational vector, optimizing the accuracy and response of the vestibular end organs to movements of the head in space. The VCR has a slower response time than the VOR due to the need to move the entire head, but has the advantage of being capable of re-aligning along tilt, yaw and roll. Combining the VOR and VCR provides the ability to maintain both the fixation and orientation of the eye on a target in space.

The camera system to be controlled by the EarBot is designed to use 4 of the 5 axes controlled by the VOR and VCR reflexes, and emulate the operation of these systems. The camera itself will be mounted on a 2-axis stage capable of performing tilt and yaw rotations, each capable of high accelerations due to a low amount of mass comprising the camera. This stage will be mounted on a second 2-axis stage capable of tilt and roll movements, which will be restricted to lower accelerations due to the mass of the camera-stage apparatus mounted atop. Each system will have a distinct plant response that may change during the lifetime of the apparatus, requiring an adaptive model of the plant to be encompassed within the control to accurately modify the EarBot characteristics over time. An illustration of the SCORPION and camera system mounted on the front of the SCORPION is shown in figure 1 & 2.

The EarBot will use a series of artificial neural networks (ANN) to emulate the circuitry used by the vestibular system to control the VOR and VCR. The three distinct subsystems that must be modelled are the extraction of the inertial motion and "neck" stage position based on accelerometer signals derived from sensors mounted in the "neck" stage, the characterization of the motor plant for each of the 4 motor axes controlling the camera position, and the controller system itself that will use the inertial motion and position information to both drive the cameras to allow target fixation. Finally, using both the plant and controller models an estimate of how far off the camera is from its intended position at any given time will be used in conjunction with the image capture system to allow for possible optical corrections to the image to align it to previous images.

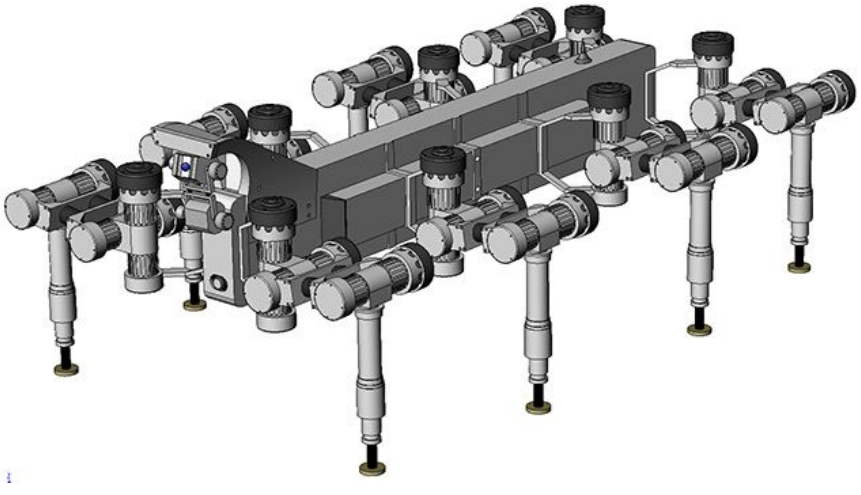


Fig. 1. Illustration of the walking SCORPION robot with camera system mounted in the front (Courtesy Craig Slyfield, Jr.)

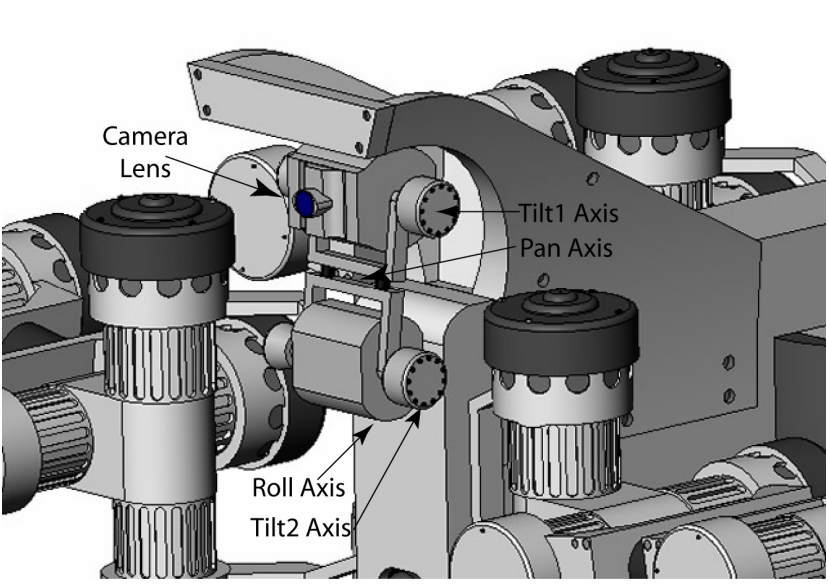


Fig. 2. Close-up illustration of camera control system and movement axes mounted on the front of the SCORPION robot (Courtesy Craig Slyfield, Jr.)

2 Background

Determining the re-orientation of a camera in space to maintain a fixation and upright orientation requires knowledge of both translational and rotational movements of the camera. Gyroscopic and linear accelerometers can provide such information, but the force exerted by gravity confounds the response of the linear accelerometers. The Einstein equivalency principle states that the response of linear accelerometers could be generated by either a translation or a re-orientation of the head relative to the gravitational vector, leading to an inherent ambiguity in interpreting the linear accelerometer signals. This effect has been studied extensively to understand how the vestibular system resolves this ambiguity, and the general consensus is that a combination of the rotational sensor information with the linear accelerometers is used.

The theoretical foundation for resolving this issue is based on using information from the rotational sensors to determine the orientation of the linear accelerometers to the gravitational vector, which provides an unambiguous resolution to the linear accelerometer values {Angelaki et al. 1999; Glasauer and Merfeld 1997; Merfeld 1995; Merfeld and Zupan 2002; Mergner and Glasauer 1999; Zupan et al. 2002}. The orientation of the head relative to gravity can be defined by the first order differential equation {Goldstein 1980}

$$\frac{d\vec{g}}{dt} = -\vec{\omega} \times \vec{g} \tag{2.1}$$

where $\vec{g}$ and $\vec{\omega}$ are vector representation of gravity and the angular velocity, respectively, in head-fixed coordinates. Defining the net acceleration detected by the linear sensors is the difference between the actual linear acceleration ($\vec{f}$) and gravitational ($\vec{g}$) components

$$\vec{\alpha} = \vec{f} - \vec{g} \tag{2.2}$$

the translational acceleration component can be computed by solving the vector differential equation { Angelaki et al. 1999; Hess and Angelaki 1997; Vie´ville and Faugeras 1990}

$$\frac{d\vec{f}}{dt} = \vec{\omega} \times \vec{\alpha} + \frac{d\vec{\alpha}}{dt} - \vec{\omega} \times \vec{f} \tag{2.3}$$

The difficulty for a neural circuit in solving this differential equation is the requirement that a series of cross-products (multiplications) between inputs must be performed, which is not an inherent functionality that is easily emulated in the traditional artificial neural network architecture (which is based on a weighted summation of inputs). Several models of how the nervous system could calculate such information have been proposed, leading to the general belief that a representation of the inertial acceleration, velocity and position (angular and linear) is available to provide an input to other processing circuits designed to generate compensatory motor movements.

Using Micro-Electro-Mechanical Sensor (MEMS) gyroscopes and linear accelerometers, the equivalent of the vestibular system can be realized in hardware. Implementation of the formulation for resolving the gravito-inertial ambiguity is readily available as an analytic approximation rather than using the proposed neural circuits. In the following simulations it is assumed that we have been able to extract the inertial motion from the sensor systems using traditional analytic methods, and that these representations are available to use as inputs into the EarBot control system.

3 Methods

In this study, the primary objective was to determine the artificial neural network design required to model the different plants describing the camera stages, the type of artificial neural network controller that would effectively drive these plants to reorient the camera in an optimal fashion, and evaluate the level of image stabilization that can be realized from using such a system.

3.1 Plant Models

Each axis of rotation on the camera stages is modelled as single robotic arms. The equations of motion for each arm on the “neck” stage are

$$\frac{\partial^2 \phi_1}{\partial t^2} = -V_n \frac{d\phi_1}{dt} - \left(P_n + g(\phi_2) * \sin(\phi_1) \right) + u_1 \tag{3.1}$$

$$\frac{\partial^2 \phi_2}{\partial t^2} = -V_n \frac{d\phi_2}{dt} - \left(P_n + g(\phi_1) * \sin(\phi_2)\right) + u_2 \tag{3.2}$$

with ϕ_1 and ϕ_2 being the angle off the vertical for each motor, V_n being damping constant associated with the velocity, P_n is the tendency to drive back to the neutral position of the motor, and g is the gravitational acceleration as a function of the other stages angle, and u_1 and u_2 are the applied torques to each respective motor. For the camera stage, the tilt motor a similar equation to 3.1 and 3.2

$$\frac{\partial^2 \phi_3}{\partial t^2} = -V_c \frac{d\phi_3}{dt} - \left(P_n + g(\phi_1, \phi_2) * \sin(\phi_3)\right) + u_3 \tag{3.3}$$

with the exception being that the gravitational factor is a function of both the angles associated with the neck motors.

The yaw motor of the camera does not have to compensate for gravity and is defined as

$$\frac{\partial^2 \phi_4}{\partial t^2} = -V_c \frac{d\phi_4}{dt} - \left(P_n * \sin(\phi_4)\right) + u_4 \tag{3.4}$$

Hard limits on the values of ϕ_1, ϕ_2, ϕ_3 *and* ϕ_4 were imposed to prevent them from exceeding ± 90 degrees from their initial positions while the simulation was in operation.

3.2 Neural Network Models of Plant Dynamics

A recursive, feedforward neural network architecture was used to capture the dynamics of the plant models. Using Matlab and the Simulink environment, plant responses to randomly varying inputs were generated to provide a "training" set of data to generate a neural network model of the plant operation. Responses of the plant models were sampled at 0.01 sec intervals for continuously varying pulse inputs randomly distributed over $\pm$ 10 rev/s^2, with random pulse durations of 0.02 to 2 seconds. Inputs to each network consisted of the applied input signal (in the case of our models, the torque) at a time τ, and the input from the two previous time periods, (τ-1) and (τ-2). Using both the current signal and previous signals allow an estimation of the instantaneous rate of change in the input, providing a form of "velocity" and "acceleration" control. The input also includes the output of the network model at times (τ-1) and (τ-2), providing feedback on the plant response signal previously generated, including the rate of change of the output relative to the previous inputs. The inputs are multiplied by a weighting matrix and summed, and then run through a non-linear "squashing" function to transform the output to a value in the range {0,1}. These intermediate or "hidden" units in the neural network provide the non-linear processing to capture the second order dynamics embodied in the plant models. Multiple intermediate nodes are used, and then the output from each of these intermediate units is summed to form the final estimate of the output of the neural network. The accuracy of the network in capturing the plant dynamics is nearly flawless, and is quite insensitive to the number of intermediate units once ~10 have been included. A sample of

the training data, plant response and neural network estimate of the plant response are shown in figure 3.

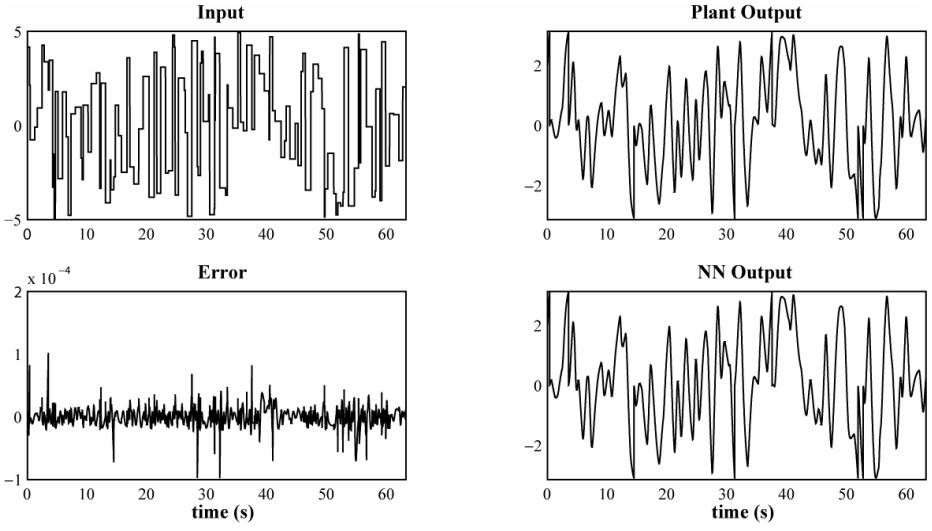


Fig. 3. Modelling of Plant dynamics with a recursive Neural Network. Input torques of random amplitude and duration are sent to the plant model of one of the camera motors, and the resulting output is recorded. A recursive neural network using two temporally successive input values, two temporally successive plant output values, and 10 intermediate nodes is trained with this data to capture the dynamic response of the system.

3.3 Robotic Simulation

The SCORPION robot is an eight-legged walking robot that is modelled after the leg design and ambulation pattern of the biological scorpion. Using a simplified controller system linking the legs to a basic motion pattern generator, the SCORPION is able to traverse complex terrain such as rocky ground, sand, stairs and varying inclines with a fair degree of stability. A camera based imaging system is mounted rigidly on the front of the SCORPION, providing the potential for significant increases in data collection capabilities since this device can penetrate areas of interest inaccessible to rover-type robotic designs.

Simulation of the SCORPION robot required the use of a physically-based modeling system from Arachi Inc, and the creation of a series of simulated accelerometer, rate-gyro, and motor torque sensors to measure the movements of the simulated robot. An idealized version of the robot was constructed adhering to the dimensions, mass, and other physical characteristics (motor damping, friction, etc.) associated with the physical device. In addition, simulated sensors to measure linear acceleration and rotational velocity were also constructed and added to the SCORPION model, allowing the simulation to test the use of proposed MEMs sensors that would be added at

these locations to construct the EarBot system. Multiple terrains were also constructed across which the SCORPION needed to traverse to test the efficacy of the EarBot controller system (figure 3).

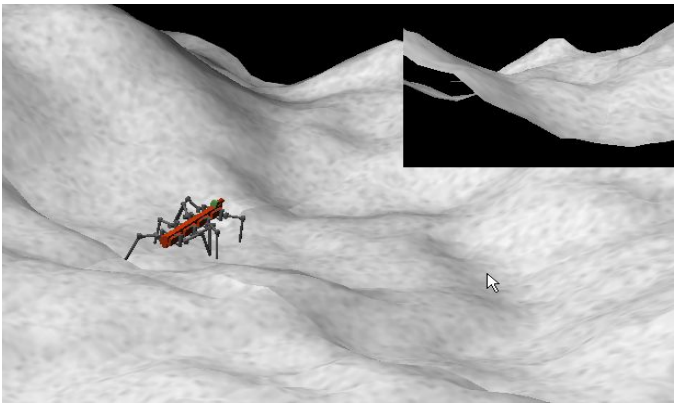


Fig. 4. Simulated SCORPION robot walking on complex terrain. Upper right-hand window is view from camera mounted on the SCORPION body.

The walking patterns used by the SCORPION robot were translated into a control system used by the simulation to drive our model. The walking patterns used in our initial tests were basic swing/stance patterns of movement for each leg, using different phase relations between the leg trajectories to drive the SCORPION's motion. Using these basic motion patterns, the typical movement of the camera mounted on the body was measured, characterizing the angles of rotation and movement distance typically encountered by the camera during simple forward motion.

4 Controller Results

Controlling the movement of the camera on the simulated SCORPION robot tested EarBot functionality. While the SCORPION was in continuous motion, the camera system mounted at the head of the SCOROPION was continuously changing its spatial orientation due to the uneven motion caused by ambulation. The EarBot had to calculate the necessary torque to drive each motor controlling camera position to maintain fixation on a target location chosen before placing the SCORPION in motion. Motion of the camera system tended to range from ± 10 degrees off of the vertical to ± 40 degrees around the vertical.

Figure 5 shows a sample output of the angular deviations from initial (desired) position and the position that the camera is driven to by the EarBot for two different neural network control architectures. The EarBot performs an adequate job of driving the camera plant dynamics to match the scorpion motion with a feedback linearization network design, but tends to lose the high frequency movements of the target destination. This would effectively move the camera system within the general area of interest, but leave viewing of the successive frames somewhat jittery. The advantages

of such a network are rapid training and minimal online computation, allowing a compact implementation in the minimal resources of a robotic device. However, the attempt to linearize the movement response of the camera systems seems to falter during rapid movements, and a more complex non-linear solution was sought.

A more generalized Model-Reference Network design (capable of doing 2nd and 3rd order and above non-linear processing on the signal) did an excellent job of matching the desired camera rotations. The Model-Reference network design uses two distinct networks: one to model the plant dynamics, and the other to model a controller that can force the plant to mimic the dynamics of a reference function. The network design for modelling the plant dynamics is the same as that used for the feedback linearization models, as described in section 3. The model reference controller network is another recursive, feedforward net. Inputs to this network include multiple sequential values from the signal driving the reference function (either 2 or 3 time steps), two sequential values from the output of the reference model, and two sequential output values from the plant model. Using the sequential time values provides the ability for the network to calculate velocity and acceleration measures on each of these signals, providing analogous information to that provided by the vestibular system. Increasing the number of recurrent inputs tended to keep the network from converging, demonstrating an unexpected sensitivity of the convergence algorithms to the composition of the input signals. Sensitivity to the number of intermediate nodes in the network was minimal, with good results typically generated using 8-13 such nodes.

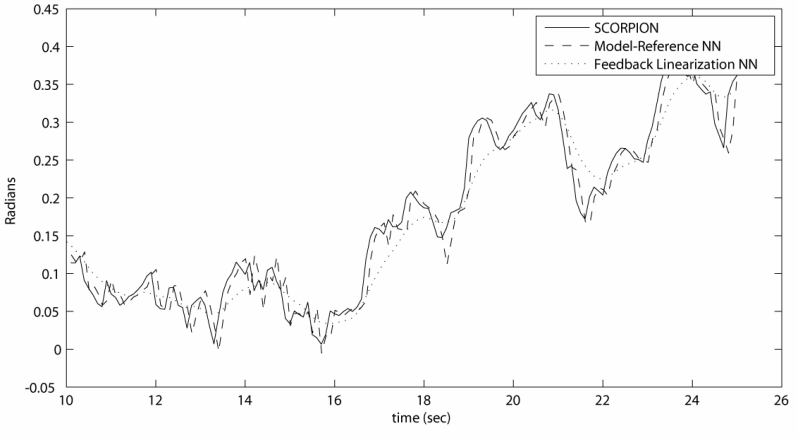


Fig. 5. Deviation from the vertical of the camera mounted on the SCORPION body during walking. Ability of a Feedback Linearization Neural Network and Model-Reference Neural Network to follow the motion of the camera is shown as an overlay of the SCORPION movement. The feedback linearization model captures fewer of the high-frequency movements than the Model-Reference design, and has a slightly longer response delay.

A lag time of several milliseconds in the response from all the controllers is due to the integration time of the system and the delay in acquiring data from the sensors, and will be prevalent to some degree depending on the physical embodiment of the

camera positioning motors and control circuitry. Similarly, delay due to sensor integration to calculate the desired positions will also affect the accuracy of the actual tracking, which is estimated to be in the 5-25 millisecond range. Corrections within the visual system to accommodate these delays are briefly discussed below.

5 Secondary Image Stabilization

Though the EarBot neural circuit can provide a robust controller for motor stabilization of the imaging devices, there will always be some degree of temporal lag between detection and adjustment to movement. Though each individual video frame may be stable and close to focusing on the desired location, the gestalt image over time will have a slight "jitter" associated with the camera continually being slightly out of the desired alignment. To compensate for this, a secondary step to stabilize the image from a camera attached to the motor plant may be performed on the images themselves based on information from the orientation sensors. Because the position of the camera may be estimated to a much higher temporal degree than the adjustments to the orientation of the plant can be made, the system may be made aware of how far out of perfect image rotational alignment it is at any moment in time. Provided that the angular errors are small, a rotational operator may be applied to the incoming image to estimate the view from the desired camera orientation.

Correction of the camera orientation may be performed with a simple rotation operator applied to the image since errors in the orientation angle are assumed to be small. Mapping pixels from the actual image plane through a rotation about the focal point of the camera to the desired image plane tends to exaggerate distortions as the operator moves farther from the focal point on the plane. This effectively limits the accuracy of the reconstruction in the outer regions of the camera, similar to the foveal regions of the eye. Similarly, if the assumption of small orientation errors is violated, then the rotation operator will not provide any reasonable approximation to the image at the desired orientation, and the system will have to rely solely on the rotation of the plant to bring it into alignment. This is similar to the need for a saccade in the visual system, allowing the eye to track with an object for a certain distance while the head is in motion but requiring a physical reset of its position once a preset rotation has been exceeded.

6 Future Work

The theoretical and simulation foundation for working with the EarBot and testing with a simulated vehicle are in place. A non-adaptive hardware implementation of the EarBot sensor system exists using a MEMS orientation sensor system, and an adaptive implementation is under design, allowing us to test the ability of the neural design to extract an accurate representation of the displacements of the cameras during motion from low-cost MEMS accelerometers and gyros. These systems should meet the low power consumption and redundancy needs for a real implementation of the EarBot to be deployed on a test vehicle such as the SCORPION, and provide the additional information required to increase the accuracy of the simulation environment to account for component failure analysis.

7 Conclusion

The EarBot has a vast potential for contributing to the control of autonomous vehicles. Many autonomous vehicles have used fairly simple and/or smooth movement behaviours, limiting the type of terrain across which they can function. By mimicking a balance system that has evolved to handle the complexities of terrain traditionally traversed by legged creatures, we can hope to control a much greater variety of autonomous vehicles in a greater number of environments. In addition to more complex terrain, greater velocities can be reached using such a system allowing for a far wider range of discovery. The EarBot simulation will also provide a blueprint for the design of new motion sensor hardware to control movement of the SCORPION and similar devices. Potential applications of this system would work with (but not be restricted to) robots on the ground, such as the DARPA funded SCORPION and its successors. Other applications under investigation are the use with undersea autonomous vehicles typically controlled remotely from the surface, using the methods for determining self-motion from external perturbations to compensate for undersea currents.

References

1. Angelaki DE, McHenry MQ, Dickman JD, Newlands SD, and Hess BJM. Computation of inertial motion: neural strategies to resolve ambiguous otolith information. *J Neurosci* 19: 316–327, 1999.
2. Glasauer S and Merfeld DM. Modelling three-dimensional vestibular responses during complex motion stimulation. In: *Three-Dimensional Kinematics of Eye, Head and Limb Movements*, edited by Fetter M, Haslwanter T, Misslisch H, and Tweed D. Amsterdam: Harwood Academic, 1997, p.387–398
3. Green AM and Angelaki DE. An integrative neural network for detecting inertial motion and head orientation. *J Neurosci* 92: 905–925, 2004.
4. Hess BJ and Angelaki DE. Inertial vestibular coding of motion: concepts and evidence. *Curr Opin Neurobiol* 7: 860–866, 1997.
5. Merfeld DM. Modeling the vestibulo-ocular reflex of the squirrel monkey during eccentric rotation and roll tilt. *Exp Brain Res* 106: 123–134, 1995.
6. Merfeld DM and Zupan LH. Neural processing of gravitoinertial cues in humans. III. Modeling tilt and translation responses. *J Neurophysiol* 87:819–833, 2002.
7. Mergner T and Glasauer S. A simple model of vestibular canal-otolith signal fusion. *Ann NY Acad Sci* 871: 430–434, 1999.
8. Vie´ville T and Faugeras OD. Cooperation of the inertial and visual systems. In: *Traditional and Non-Traditional Robotic Sensors*, edited by Henderson TC. Berlin: Springer-Verlag, 1990.
9. Zupan LH, Merfeld DM, and Darlot C. Using sensory weighting to model the influence of canal, otolith and visual cues on spatial orientation and eye movements. *Biol Cybern* 86: 209–230, 2002.

Motion Estimation with Edge Continuity Constraint for Crowd Scene Analysis

B. Zhan, P. Remagnino, S.A. Velastin, N. Monekosso[1], and L-Q. Xu[2]

[1] Digital Imaging Research Centre, Kingston University, UK
[2] British Telecom Research,Ipswich,UK
{B.Zhan, P.Remagnino}@kingston.ac.uk

Abstract. This paper presents a new motion estimation method aimed at crowd scene analysis in complex video sequences. The proposed technique makes use of image descriptors extracted from points lying at the maximum curvature on the Canny edge map of an analyzed image. Matches between two consecutive frames are then carried out by searching for descriptors that satisfy both a well-defined similarity metric and a structural constraint imposed by the edge map. A preliminary assessment using real-life video sequences gives both qualitative and quantitative results.

1 Introduction

Understanding crowd behavior is a relatively new research topic in computer vision. It can be applied to a variety of domain problems, including space optimization, ambient intelligence and visual surveillance. In this paper we describe a new technique that combines image descriptors and edge information to estimate the crowd motion of video sequences to better understand behavior analysis.

People's behavior differs when they walk, either as individuals or when they are part of a crowd. Crowds have been studied by sociologists and civil engineers which has led to physical models that are proposed to describe quantitatively complex behavior dynamics[1],[2]. Le Bon compares a crowd to a chemical compound as it displays properties quite different from those of the individual bodies that form its mass [3]. Moreover, methods that are normally employed to describe the behavior of an individual would fail in crowd situations, especially when the level of clutter is very high. Computer vision methods have been used to focus on extracting information from video sequences of crowded scenes; for instance, they are used to estimate crowd density, e.g., [4],[5]. In order to model behavior, the tracking of individuals and groups of people must be implemented. Two categories of tracking can be broadly defined: one for detecting and tracking a single person in a crowd, which requires a window of interest area to start with (see the work of [6]) and which employs the Harris interest points detector and point distribution model; the second one is for detecting and tracking a few (3 to10) people using sophisticated tracking techniques such as the particle filter [7], Markov Chains [8] and probabilistic frameworks [9]. Furthermore,

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 861–869, 2006.
© Springer-Verlag Berlin Heidelberg 2006

recent work has focused on the interpretation of densely crowded scenes [10] that employ statistical methods to extract the main paths of a crowded scene or use hidden Markov models to describe the normal behavior of a crowded scene [11].

Our main goal is to employ computer vision methods to develop a robust motion estimation method and to statistically model both the instantaneous and recurrent dynamics of a crowded scene. In this paper we propose an algorithm to estimate the crowd motion in scenes of various levels of density and clutter. Deformable object tracking techniques [12] motivated the use of the edge information and its curvature to extract descriptor points (those with local maximum curvature) as salient features of an edge. Instead of using points, edge information is maintained by "edgelet constraint" to refine the estimate. Thus we combine the advantage of not only using point features which are flexible to track, but also the advantage of using edge features which maintain structural information.

The paper is organized as follows: Section 2 describes our proposed method in detail, Section 3 presents selected results of applying the method to different dense visual scenes, and in Section 4 we give concluding remarks and discuss future work.

2 Proposed Method

The motion estimate between frames is carried out in four steps:

- A conventional Canny edge detector is run over each image frame and the edge chains are retrieved;
- the curvature is calculated for every point on an edge chain, and along each chain the interest points at maximum curvature are chosen;
- then, for all the extracted points in the first frame of each frame pair, we search for matching candidate points in the second frame;
- and finally the edgelet constraint is applied to obtain the best point matches.

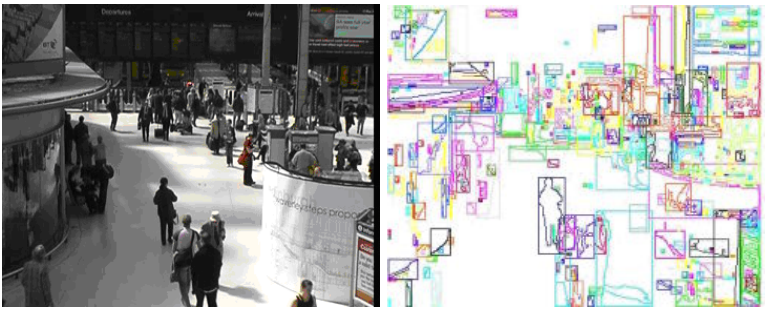


Fig. 1. Left:original frame;right:Edge chains and their bounding boxes

Sections 2.1 to 2.4 describe the above steps in detail; Section 2.5 explains the role of the background model so as to improve the performance of the proposed method.

2.1 Edge Retrieval

The Canny edge detector is employed to extract the edge information of a given frame. Each Canny edge is a chain of point Sp, and all the edges are stored in an edge list Lp. Figure1 left and right show an example image frame and the extracted edge chains with associated bounding boxes respectively. It can be observed that even in a scene which depicts a crowd of moderate density, edge chains can occlude each other, increasing the descriptor matching complexity.

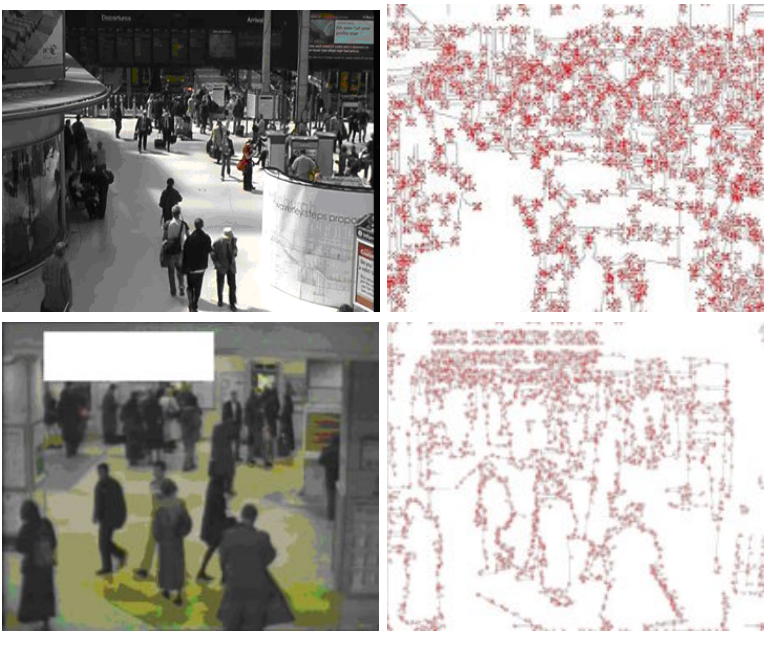


Fig. 2. Original frame (left) and extracted corner points (right), marked with red crosses on grey edges. Two scenes of different complexity levels are illustrated.

2.2 Curvature Estimation and Corner Point Extraction

Interest points, can be quickly extracted for a sequence frame with for instance the Harris corner operator [13]. Although the interest points can represent the local characteristics of an image in isolation, they cannot represent a shape. We therefore propose the extraction of interest points from the edges and then impose the constraint that they lie on a specific edge.

Each edge can be represented by a parameterized curve:

$$x = x(t) \; and \; y = y(t) \tag{1}$$

We smooth the curve with a Gaussian filter, as follows:

$$\begin{aligned} X(t) &= G(t) \otimes x(t) \\ X'(t) &= G'(t) \otimes x(t) \\ X''(T) &= G''(t) \otimes x(t) \end{aligned} \tag{2}$$

The curvature of each edgelet can then be given by [14]:

$$\kappa = \frac{X'Y'' - Y'X''}{(X'^2 + Y'^2)^{\frac{2}{3}}} \tag{3}$$

first- and second-order derivative filters, can be easily implemented using the method described in [15]. Corner points are defined and extracted as the local maxima of the absolute value of curvature on each edge.

Thus we convert the edge representation from a point sequence Sp to a corner point sequence Sc, resulting in a list Lc of Sc for all the edges of the image.

2.3 Point Matching by Curvature

Given the two frames $I(t_0)$ and $I(t)$, we estimated the motion of the points from $I(t_0)$ to $I(t)$. For each corner point $C_{t_0}(j,i)$, with coordinate (x,y) from every $S_{t_0}(t,j)$ of $L_c(t_0)$, we search in $I(t)$ the area inside a rectangular search window W centered at (x,y). A look-up table (LUT) contains corner point information of $Lc(t)$ is generated to enhance the matching. (Illustrated in Figure3). The best n matches are then selected as candidate points and considered for the next step.

The correspondence is matched by using curvature information of corner points in W in LUT against the reference point $C_{t_0}(j,i)$. The error is calculated by the curvature:

$$\epsilon(i,j) = |\kappa(C_{t_0}(j,i)) - \kappa(C_t(i',j'))| \tag{4}$$

2.4 Applying the Edgelet Constraint

Complex dynamics and frequent occlusions generated in crowd scenes make the estimation of motion a very complex task. Point matching in isolation is too fragile and prone to errors to provide a good motion estimator. If the interest points are extracted on edge chains, then the edge constraint can be imposed and used.

For an image frame $I(t_0)$, we divide every $S_{t_0}(j)$ to a uniform length edgelets represented by sub sequences $E_{t_0}(j,k)$. There are two reasons for doing this: to avoid a very long edge that could be generated by several different objects, and to enhance the matching of the edge fragments that are generated by occlusions. For each corner point Ct0 (j,k,i) in $E_{t_0}(j,k)$ we have as result from the first

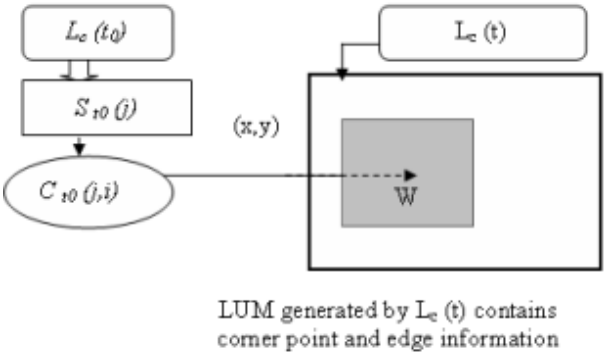


Fig. 3. The procedure of matching corner points

step, n candidate matching points. Each candidate point belongs to a sequence $S_t(j')$ in $L_c(t)$, thus we have $m(m \leq n)$ candidate matching sequences (or pieces of sequence) for each edgelet.

To find the best match of $E_{t_0}(j,k)$, we use three parameters: *energy cost, variation of displacements* and the *match length* for each candidate and combine them into a single matching score. Here we assume that the length of $E_{t_0}(j,k)$ is small enough so that it would not split again to two or more matches. This is so that their candidate points correspond to the same candidate sequence.

Energy cost $\mathcal{E}$. This refers to the deformable object match which is calculated by accumulating the errors ϵ (again calculated by the difference of the curvatures as in Equation 4 along matching point pairs of $E_{t_0}(j,k)$ and all the candidate match points that belong to the same candidate sequence.

$$\mathcal{E} = \sum \epsilon(i,j,k) = \sum |\kappa(C_{t_0}(k,j,i)) - \kappa(C_t(k',i',j'))| \tag{5}$$

Variation of displacements $\mathcal{V}$. For each matching point pair we have a displacement pair dx_i and dy_i, the combination of the variation of the two displacement vectors are as follows:

$$\mathcal{V} = \frac{1}{L_M} \times \sqrt{\sum \frac{1}{N}(dx_i - dx)^2 + \frac{1}{N}(dy_i - dy)^2} \tag{6}$$

where dx and dy are the average displacements between the matched point pairs, N is the size of the match window, L_M is the number of total matched points of from $E_{t_0}(j,k)$ to candidate sequence. Hence $\mathcal{V}$ lies in the range between 0 and 1.

And **match length parameter $\mathcal{M}$**:

$$\mathcal{M} = \frac{L_M}{L_E} \tag{7}$$

Where L_E is the total number of points on $E_{t_0}(j,k)$, M between 0 to 1.

So the overall matching score is given by:

$$\mathcal{S} = \mathcal{E} + \mathcal{V} + (1 - \mathcal{M}) \tag{8}$$

The candidate sequence of minimum matching score will be selected. However, if the match length parameter $M < 50\%$, we will discard the result.

2.5 Improving Results with Background Modelling

Background modeling is commonly employed to segment the foreground, though here we use it to reduce noise. When a scene is very crowded and people frequently occlude one another, it is not practical to segment the foreground solely with background modeling. However, there could be still some parts of the scene that are never covered by foreground objects (e.g. a ceiling) which we can describe as a permanent background. In our study we wanted to use the background model to eliminate the noise generated by the permanent background. We adopted the Gaussian mixture model proposed by Stauffer [16] which builds an adaptive and updatable background model on a pixel by pixel basis to generate a foreground (FG) mask. In some cases a FG mask may lose some edge points, which could cause some of the FG edges to be broken by the background and lose their consistency. To avoid this we did not apply the FG mask directly on the image, but applied the mask when we extracted the corner points. In other words we eliminated the corner points that fell into the background while keeping the connection information.

3 Experimental Results

We tested our algorithm against some of the video data which contained different crowd scene densities, with different frame rates, and with different perspectives

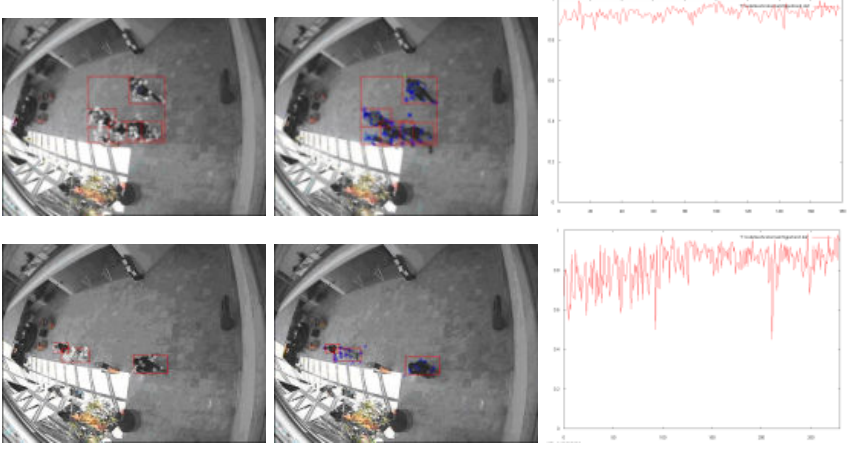


Fig. 4. Two test data sets. The left two sample initial frames from both data set, with corner points indicated by white dots inside ground truth box; the middle two are the matched frame, with correct matched points CRM marked by blue circle and incorrect matched points ICRM marked by cross, the right two are the correct match rate R along the frames of the sequences.

of pedestrians (there were different camera set-ups and therefore perspectives). We chose to assess some of the results quantitatively, those including video sequences with as few as 3 to 4 people, for which the ground truth is known. Whilst for others we chose to asses qualitatively, thereby illustrating the flow of people from crowded scenes.

3.1 Motion Estimation of Multiple People

The video data we used is from the European project CAVIAR [17]; the ground truth information for this data is provided in XML format.

To test the result, for $I(t_0)$ we estimated the foreground object position (by means of a bounding box) and translated every corner point in the bounding



Fig. 5. Visualized Results from different types of video sequences (left ones), against the results from optical flow method run on same sequences (right ones)

box to the matching frame $I(t)$ by its estimated motion, we then counted all the translated points still in the box as a correct match CR_M, and those falling outside of the box as an incorrect match ICR_M , and the correct match rate is calculated as:

$$R = \frac{CR_M}{CR_M + ICR_M} \tag{9}$$

The optimal situation R should always be equal to 1.

3.2 Motion Estimation of More Complex Video Sequences

We tested our algorithm on different types of video sequences, including different number of people and frame rate. Top of Figure 5 is a snapshot of a high frame rate se-quence with medium density of people while middle and bottom of Figure 5 are low frame rate sequence (typically 1 frame/sec or less) with high density of people. Results are com-pared against those obtained using optical flow method run on the same sequences. A detailed illustration from bottom sequence of Figure 5 is given in Fig. 6, with one of the two consecu-tive frames being overlaid on top of the other, which gives us an expression of loca-tion of the pedestrians in the two frames. The corner points in both frames and their correspondence are indicated on the figures.

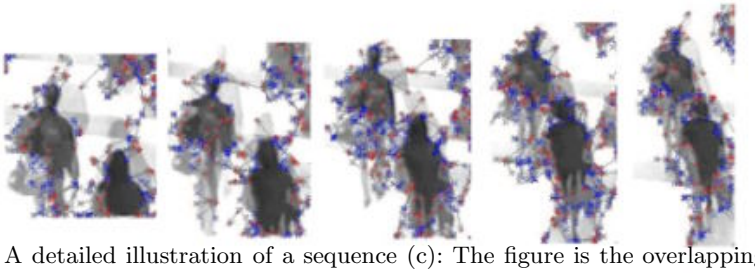


Fig. 6. A detailed illustration of a sequence (c): The figure is the overlapping of two consecutive images, with the initial corner points indicated by blue crosses and the matched points indi-cated by red circles

4 Conclusions and Future Work

In this paper we have introduced a novel method to estimate the motion of a crowd in complex scenes and thus provided the basis for a high level description of a crowd for interpretation and modeling. The method in its essence relies on edge information for the extraction of image descriptors and their matching between frames. Corner points are selected as salient features and edgelets are employed to maintain the local edge information, an adaptive background model is associated with the system to reduce the noise. Future work will entail the clustering of the extracted corner points and the edgelets to groups that represent a pedestrian or a group with a common movement, and thus the method can be extended to the actual tracking of individuals or groups of people in very complex scenes.

Acknowledgements

The first author (Beibei Zhan) is grateful to BT Group PLC for partially funding her PhD research.

References

1. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical Review **51** (1995) 4282–4286
2. Pan, X., Han, C.S., Law, K.H.: Human and social behavior in computational modeling and analysis of egress. Building Future Council Doctoral Program (2005)
3. Le Bon, G.: The Crowd. Cherokee Publishing Company (1895)
4. Marana, A., Velastin, S., Costa, L.d.F., Lotufo, R.: Automatic estimation of crowd density using texture. Safety Science **28** (1988) 165–175
5. Ma, R., Li, L., Huang, W., Tian, Q.: On pixel count based crowd density estimation for visual surveillance. In: IEEE Conference on Cybernetics and Intelligent Systems, IEEE (2004) 170–173
6. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. In: British Machine Vision Conference, Oxford (2005)
7. Venegas, S., Knebel, S., Thiran, J.: Multi-object tracking using particle filter algorithm on the top-view plan. (2003)
8. Smith, K., Gatica-Perez, D., Odobez, J.: Using Particles to Track Varying Numbers of Interacting People. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on **1** (2005)
9. Marques, J., Jorge, P., Abrantes, A., Lemos, J.: Tracking Groups of Pedestrians in Video Sequences. IEEE WoMOT **9** (2003) 101
10. Zhan, B., Remagnino, P., Velastin, S.: Mining paths of complex crowd scenes. Lecture notes in computer science (2005) 126–133
11. Andrade, E., Blunsden, S., Fisher, B.: Hidden markov models for optical flow analysis in crowds. In: the 18th international conference on pattern recognition. (2006)
12. Cohen, I., Ayache, N., Sulger, P.: Tracking points on deformable objects using curvature information. Proceedings of the Second European Conference on Computer Vision (1992) 458–466
13. Harris, C., Stephens, M.: A combined corner and edge detector. Alvey Vision Conference (1988) 147–151
14. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. IEEE Trans. Pattern Anal. Mach. Intell. **14** (1992) 789–805
15. YOUNG, I., VAN VLIET, L.: Recursive implementation of the Gaussian filter. Signal processing **44** (1995) 139–151
16. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2** (1999) 246–252
17. EC: Funded caviar project http://homepages.inf.ed.ac.uk/rbf/caviar/ (2002-2005)

Perceptual Grouping Based on Iterative Multi-scale Tensor Voting

Leandro Loss[1], George Bebis[1], Mircea Nicolescu[1], and Alexei Skourikhine[2]

[1] Computer Vision Laboratory, University of Nevada, Reno
[2] Space and Remote Sensing Sciences Group, Los Alamos National Laboratory
{loss, bebis, mircea}@cse.unr.edu, alexei@lanl.gov

Abstract. We propose a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Segments are represented as second-order tensors and communicate with each other through a voting scheme that incorporates the Gestalt principles of visual perception. An iterative scheme has been devised which removes noise segments in a conservative way using multi-scale analysis and re-voting. We have tested our approach on data sets composed of real objects in real backgrounds. Our experimental results indicate that our method can segment successfully objects in images with up to twenty times more noise segments than object ones.

1 Introduction

Grouping processes, which "organize" the given data by eliminating the irrelevant items and sorting the rest into groups, each corresponding to a particular object, are indispensable in computer vision. Determining groups in a given set of points or edgels can be a very difficult task, as the actual measurement of compatibility within a sub-set is not well defined. There has been considerable research work in this area over the last two decades.

In [1], Lowe discusses the importance of the Gestalt principles of collinearity, co-curvilinearity and simplicity for perceptual grouping. Ahuja and Tuceryan [2] have introduced methods for clustering and grouping sets of points having an underlying perceptual pattern. Mohan and Nevatia [3] have assumed *a-priori* knowledge of the contents of the scene (i.e., aerial images). A model of the desired features was then defined, and groupings were performed according to that model. Ullman [4] has suggested that a curve joining two edge fragments is formed by a pair of circular arcs that minimizes the integral of the square of the curvature. He also proposed a network model, but no results were shown. Clearly, elliptical curves cannot be constructed by joining only a pair of circular arcs. Also, this scheme cannot be easily generalized to a set of three or more edge fragments, and does not allow for outliers. Parent and Zucker [5] have described a relaxation labeling scheme where local kernels were used to estimate tangent and curvature. These kernels used support functions based on co-circularity. Somewhat similar kernels are used in our methodology, but applied in a different way. Ullman and Sha'ashua [6] have proposed the use of a saliency measure to

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 870–881, 2006.
© Springer-Verlag Berlin Heidelberg 2006

guide the grouping process, and eliminate erroneous features in the image. Their scheme prefers long curves with low total curvature by using an incremental optimization scheme.

More recently, Williams and Thornber [7] have proposed a probabilistic approach based on *Closed Random Walks*. In their approach, saliency is defined relatively to the number of times an edge is visited by a particle in a random walk. The main restriction assumed in that work was that the movement has to start and finish on the same edge (i.e., closed random walk). This reduces the number of paths to consider along with the complexity of the problem, however, it imposes a restriction that is not practical. Their technique was compared with five other methods in the literature and found to outperform them considering a benchmark of real objects.

The use of voting for salient feature inference from sparse and noisy data was introduced by Guy and Medioni [8] and then formalized into a unified tensor voting framework [9]. Tensor voting represents input data as tensors and interrelates them through voting fields which are built from a saliency function that incorporates the Gestalt laws of proximity and continuation. The methodology has been used in 2D for curve and junction detection and for figure completion. It has also been applied in 3D for dense reconstruction from stereo or multiple views and for tracking.

In this paper, we propose a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Specifically, we have devised an iterative scheme that removes noise segments using multi-scale analysis and re-voting. In contrast to traditional tensor voting approaches, that use hard thresholding and single-scale analysis, our method removes noise segments conservatively according to their behavior across a range of scales, and applies re-voting on the remaining segments to estimate saliency information more reliably. Our experimental results illustrate that this iterative, multi-scale thresholding scheme, coupled with re-voting, improves segmentation results remarkably.

The rest of this paper is organized as follows: Section 2 reviews the tensor voting framework, discusses the main challenges in employing tensor voting for grouping, and presents the new approach. Section 3 describes the datasets used in our experiments and the evaluation methodology. Section 4 presents our experimental results and comparisons. Finally, conclusion and directions for future work are presented in Section 5.

2 Perceptual Grouping Using Tensor Voting

2.1 The Tensor Voting Framework

In the framework proposed by Guy and Medioni [8], input data is encoded as elementary tensors. Support information (including proximity and smoothness of continuity) is propagated from tensor to tensor by vote casting. Tensors that lie on salient features (such as curves in 2D, or curves and surfaces in 3D) strongly support each other and deform according to the prevailing orientation, producing generic tensors. Each such tensor encodes the local orientation of features (given

by the tensor orientation), and their saliency (given by the tensor shape and size). Features can be then extracted by examining the tensors resulted after voting.

In 2D, a generic tensor can be visualized as an ellipse. It is described by a 2×2 eigen-system, whose eigenvectors e_1, e_2 give the ellipsoid orientation, while eigenvalues λ_1, λ_2 (with $\lambda_1 \geq \lambda_2$) give its shape and size. The tensor is represented as a matrix S:

$$S = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T \tag{1}$$

There are two types of features in 2D - curves and points (junctions) - that correspond to two elementary tensors. A curve element can be intuitively encoded as a *stick tensor* where one dimension dominates (along the curve normal), while the length of the stick represents the curve saliency (confidence in this knowledge). A point element appears as a *ball tensor* where no dimension dominates, showing no preference for any particular orientation.

Input tokens are encoded as such elementary tensors. A point element is encoded as a ball tensor, with e_1, e_2 being any orthonormal basis, while $\lambda_1 = \lambda_2 = 1$. A curve element is encoded as a stick tensor, with e_1 being normal to the curve, while $\lambda_1 = 1$ and $\lambda_2 = 0$. Tokens communicate through a voting process, where each token casts a vote at each token in its neighborhood. The size and shape of this neighborhood, and the vote strength and orientation are encapsulated in predefined voting fields (kernels), one for each feature type - there is a stick voting field and a ball voting field in the 2-D case. Specific details about the voting generation process can be found in [8].

At each receiving site, the collected votes are combined through simple tensor addition, producing generic tensors that reflect the saliency and orientation of the underlying salient features. Local features can be extracted by examining the properties of a generic tensor, which can be decomposed in its stick and ball components:

$$S = (\lambda_1 - \lambda_2) \cdot e_1 e_1^T + \lambda_2 \cdot (e_1 e_1^T + e_2 e_2^T) \tag{2}$$

Each type of feature can be characterized as: (a) *Curve* (saliency is $(\lambda_1 - \lambda_2)$, normal orientation is e_1), and (b) *Point* (saliency is λ_2, no preferred orientation). After voting, curve elements can be identified as they have a large curve saliency $\lambda_1 - \lambda_2$ (and appear as elongated tensors), junction points have a local large point saliency λ_2 and no preferred orientation (appear as large ball tensors), while noise points have low point saliency. Therefore, the voting process infers curves and junctions simultaneously, while also identifying outliers (tokens that receive little support). The method is robust to considerable amounts of outlier noise and does not depend on critical thresholds, the only free parameter being the scale factor σ which defines the voting fields.

2.2 Applying Tensor Voting for Grouping

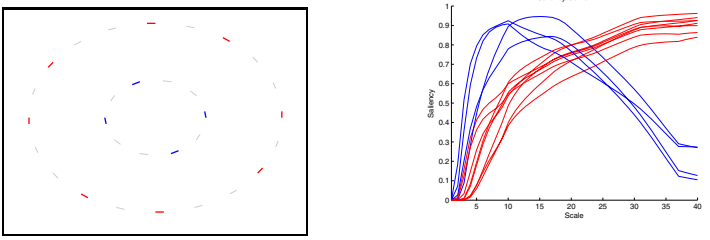
Although the tensor voting framework has only one free parameter, the scale σ, several other issues must be considered when using it for perceptual grouping

and segmentation. The voting dimensionality, the features to be used as tokens, and the encoding of the input tokens are examples of issues that need our attention. The voting dimensionality is determined by the number of features to be used to represent the information at hand. Ideally, a small number of features with maximal representation capability is desired. This raises the issue of what features to use as input tokens. Token encoding has considerable impact on the performance of the framework. In the case of edges, one can choose several different tensor representations. For example, one way would be assigning a ball tensor at each pixel of the contour. Alternatively, one can assign a stick tensor at each pixel location with position and orientation information computed according to its adjacent neighbor; since tensors have no direction, the choice of the neighbor does not matter. Finally, one could also select representatives of the contour and initialize them as stick tensors. In this study, we follow this last approach. In our case, we re-sample the contour using a fixed sampling step, and initialize the framework using the sampled pixels which are encoded as stick tensors. Their position is determined by the position of the sample and their orientation by using the direction proportional to the gradient.

Another issue that needs careful consideration is the selection of the scale parameter σ. In [10], it was found that the framework has low sensitivity with respect to σ. However, finding the appropriate σ value might not be easy in practice. It is well known that small scales capture local structures while large scales capture global configurations. In a real scenario, it is unlikely that we would have any *a-priori* information about the size of objects in the scene, making the choice of the scale parameter a "trial-and-error" process. In general, the choice of the scale parameter will vary from application to application, or even worse, from image to image. Moreover, analyzing information at a single scale can compromise or make hard the detection of structures having different sizes.

This situation can be illustrated using an image with two similar figures, one smaller than the other, shown in Fig. 1. To help the visualization, we have plotted "Scale versus Saliency" curves, thereafter called *saliency curves*. Specifically, a saliency curve is created by voting in different scales and computing the saliency of each segment in each scale. We then normalize the saliency curves according to the average saliency of all segments in the image. This is done in order to prevent a monotonically increasing curve. Such a situation could result from the fact that, as the voting neighborhood increases, the segment saliency increases simply because new segments are considered.

As the voting neighborhood increases, the smaller circle starts becoming more salient since more of its segments are considered in the voting process. Its saliency maximum is reached when the voting neighborhood covers all its segments, (i.e., when scale σ is around 10). After this point, not having any more segments to strengthen its saliency, the smaller circle starts "losing" saliency for the larger one, which becomes more salient as more of its segments are included the voting neighborhood. After the larger circle reaches its saliency maximum, at scale σ around 35, its saliency curves stabilizes since there are no more segments to consider beyond this scale.



(a) (b)

Fig. 1. Two circles having different size and saliency. (a) Segments having their saliency behavior analyzed. (b) Normalized saliency curves corresponding to the segments chosen in (a). The saliency of the smaller circle grows until the voting neighborhood covers all of its segments. After this point, the saliency of the larger circle surpasses the saliency of the smaller circle and keeps growing until it reaches its own maximum.

Another important issue when segmenting a figure from background is the choice of a threshold to filter out non-figure segments. It is reasonable to expect that if the saliencies of the figure are quite different from those of the background, then it would be easy to find a threshold value to separate them completely. Fig. 2 shows a simple example. In this example, we consider a well-formed circle surrounded only by random noise at a signal-to-noise ratio (SNR) equal to 70%. If we apply tensor voting to the image segments and plot the saliency histogram, it is easy to see that eliminating segments with saliencies below a threshold value T equal to 45% filters out noise completely.

However, this is hardly the case in practice. Let us consider, for example, the image shown in Fig. 3(a). If we apply tensor voting to its segments and plot the corresponding saliency curves or saliency histogram at a large scale, we can easily verify that there is no way to get a perfect segmentation of the figure from the background. This would be also true at different scales. Moreover, even if we choose an optimal threshold value using the saliency histogram, the number of misclassified segments would be unavoidably large (i.e., see Fig. 3(d), 3(e), 3(f)).

2.3 Iterative Multi-scale Tensor Voting Scheme

Aiming at eliminating the largest number of background segments while preserving most figure ones, we have developed an iterative scheme based on multi-scale analysis and re-voting. The main idea is removing segments from the image conservatively in an iterative fashion, and applying re-voting on the remaining segments, to estimate saliency information more reliably. Improvements in figure segmentation come from the fact that, after each iteration, low salient segments are filtered out and, after the subsequent re-voting sessions, the support to background segments is reduced. After a conservative elimination of segments, the difference in saliency between figure and ground segments becomes much more pronounced.

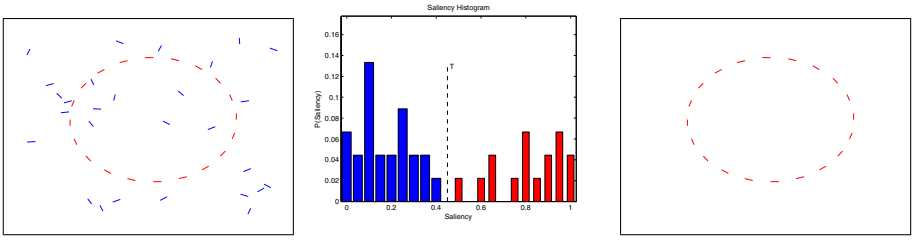


Fig. 2. A simple example where figure and background can be separated by using a single threshold. (a) Original image. (b) Saliency histogram and choise of threshold T. (c) Resulted segmentation.

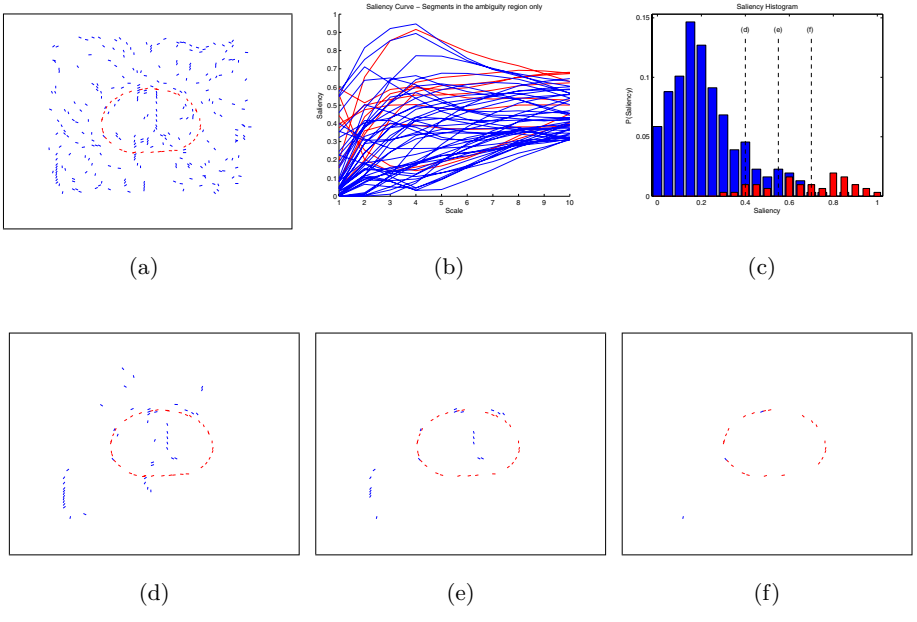


Fig. 3. An image with SNR equal to 15% processed by different threshold values. A fixed threshold value (T), cannot result in a good segmentation at any scale. (a) Original image. (b) Saliency curves corresponding to segments of the figure and background in the ambiguity region. (c) Saliency histogram with various threshold choices. (d) Thresholding at 40%. (e) Thresholding at 55%. (f) Thresholding at 70%.

In our methodology, the conservative elimination is done by applying a low threshold value T_s, which removes almost always background segments. A new application of tensor voting follows so that a new saliency map is obtained, without considering the eliminated segments. After re-voting, the threshold value is increased to adapt to the strengthening of figure saliency due to elimination

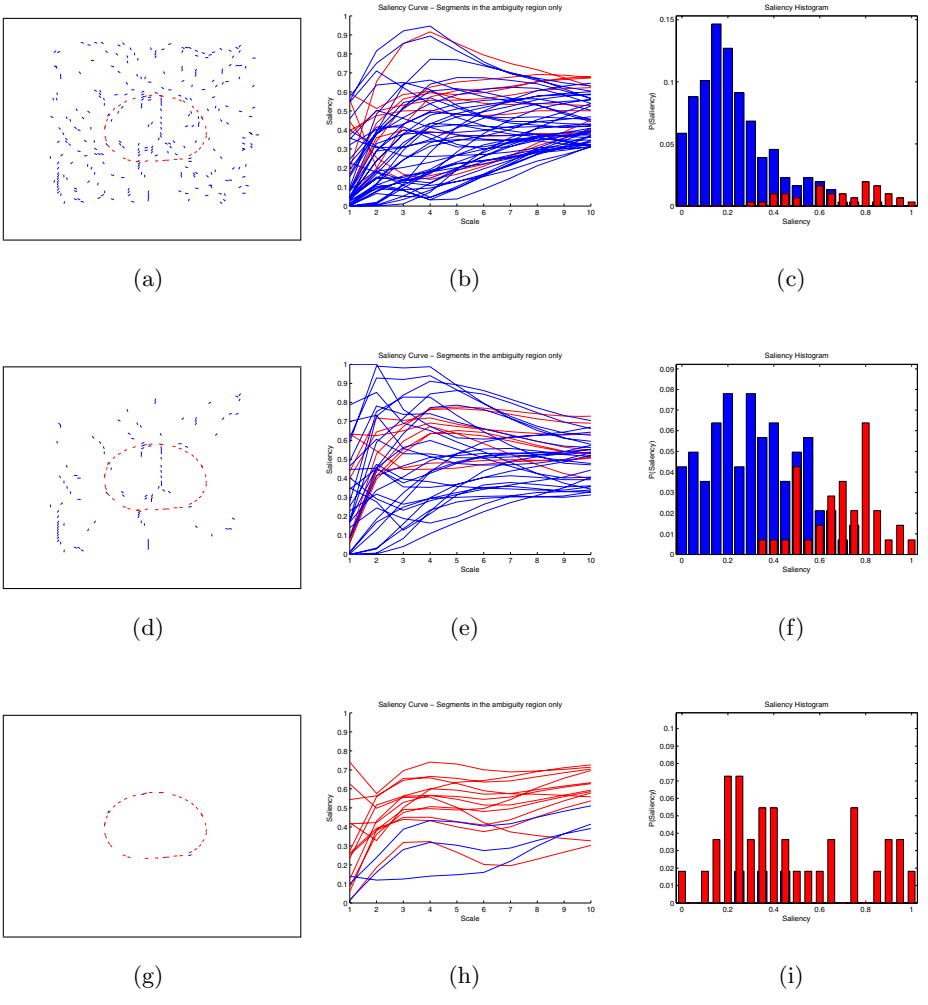


Fig. 4. Image with SNR equal to 15% processed by our iterative, multi-scale thresholding scheme. By conservatively eliminating segments, the saliency difference between figure and background starts becoming more pronounced. (a) Original image. (b) Saliency curves of background segments in the ambiguity region. (c) Saliency histogram at the largest scale. (d) Resulting image using $T_s = 20\%$. (e) Saliency curves of the segments in the ambiguity region for image (d). (f) Saliency histogram at the largest scale for image (d). (g) Resulting image using $T_s = 40\%$. (h) Saliency curves of the segments in the ambiguity region for image (g). (i) Saliency histogram at the largest scale for image (g).

of noise. In practice, we increase T_s after each re-voting session by a fixed step $Step_s$.

Multi-scale analysis is incorporated to this scheme by voting in a number of scales and thresholding according to the behavior of saliency along them.

Specifically, the saliency curve is computed by voting in different scales and computing the saliency of each segment in each scale. Segments are then eliminated if they do not present any significant peaks of saliency across a range of scales. This preserves salient segments of any size. Algorithmically, this is implemented by counting how many times the saliency curve of a segment is above the threshold T_s. If the number of times does not exceed another threshold T_σ, we consider that the segment does not have a significant saliency peak and it is eliminated. As mentioned in the previous section, we normalize the saliency curves according to the average saliency of all segments in the image.

Fig. 4 shows the behavior of figure (red) and background (blue) saliencies as the multi-scale, adaptive thresholding is applied. The image has a SNR equal to 15% (i.e., about 7 times more background segments than figure ones). The threshold values T_s goes from 10% up to 40% with $Step_s$ equal to 10%. The voting was performed with a σ ranging from 1 (5% of image size) to 20 (100% of image size). The improvements over the using a fixed threshold and single scale (i.e., naïve approach) are remarkable.

3 Datasets and Evaluation Methodology

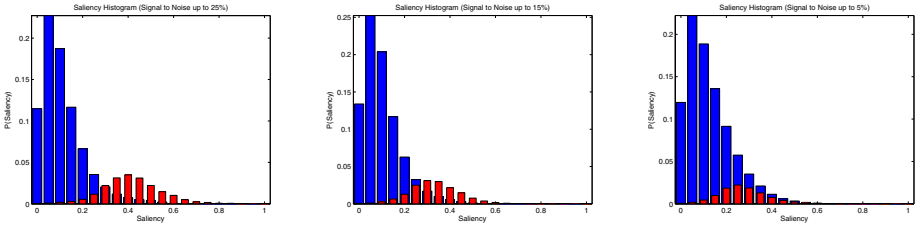
Experiments were performed based on the set of fruit and texture sampled silhouettes provided in [7]. As in [7], each benchmark image was created from a pair of sampled silhouettes belonging to a fruit or a vegetable (thereafter called figure) and textured background (thereafter called background). Nine figure silhouettes were re-scaled to an absolute size of 32x32 and placed in the middle of nine 64x64 re-scaled ground windows. We experimented with five different SNRs in order to reduce the number of figure segments proportionally to the number of background segments. Further details regarding this benchmark can be found in [7]. Fig. 7 shows some examples of benchmark images at different SNRs. The total number of images used in our experiments was 405.

Quantitative evaluations and comparisons of different methods were performed using Receiver Operational Characteristic (ROC) curves, (i.e. False Positives (FP) versus False Negatives (FN)). A FN is a figure segment detected as background; conversely a FP is a background segment detected as figure. The ROC curves presented are average curves over all images in the dataset. In order to allow a direct comparison with Williams and Thornber's method (WT) [7], SNR vs FP and SNR vs FN graphs are also shown.

4 Experimental Results and Comparisons

Saliency histograms were plotted for the different SNRs used in [7] (see Fig. 5). For each histogram, we used 81 images (9 figures and 9 backgrounds).

It can be noted that, as SNR decreases, figure (red) and background (blue) start overlapping up to a point where figure becomes indistinguishable from background. The larger the overlap between figure and background, the harder is to visually separate the object in the image. At some point (for instance,

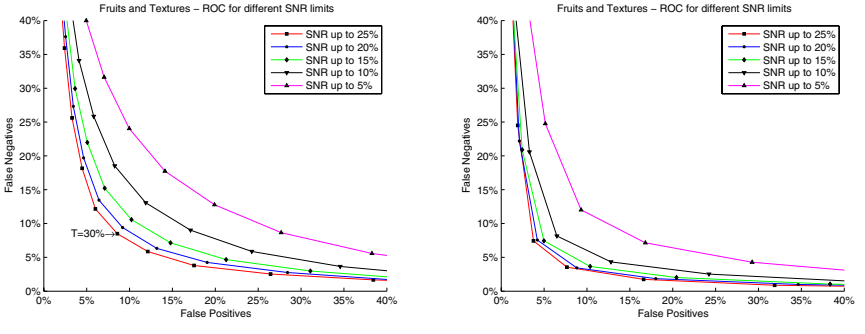


(a) SNR at 25% (b) SNR at 15% (c) SNR at 5%

Fig. 5. Saliency behaviors assuming various SNRs (red corresponds to figure and blue corresponds to background). σ was set to 20 (i.e., voting field covers the entire image). As SNR decreases, background and figure start to overlapping up to the point where figure becomes indistinguishable from background.

when SNR is below 10%) the structures of the background are visually more distinguishable than the figure itself. This effect is mainly due to the use of sampled textures (leaves, bricks, etc) as background instead of random noise.

Figure 6(a) shows the ROC curves assuming a single-scale and a fixed threshold. The scale was chosen based on knowledge of the benchmark images (i.e., σ was set equal to 20, yielding a voting field that covers the entire image).



(a) (b)

Fig. 6. (a) ROC curves for different SNR limit acceptance. At SNR below 10%, the perception of the figure becomes more difficult. This is reflected by the overlapped curves in their histograms of Fig. 5, and the worst performance for the ROC curves for SNR up to 10% and 5%. (b) ROC curves for different SNR limits according to the multi-scale, adaptive threshold with Step_s equal to 5%. It is possible to note improvements in all ROC curves comparing to the curves for the naïve approach (single-scale, fixed threshold) shown in (a). In addition, the curve for SNR up to 10% is closer to the higher ones (25%, 20% and 15%), showing that the iterative, multi-scale adaptive thresholding deals better with cluttered images.

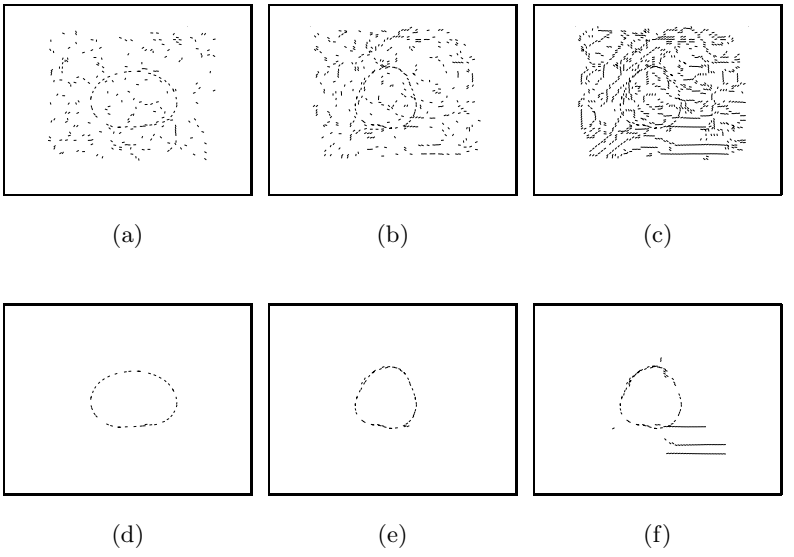
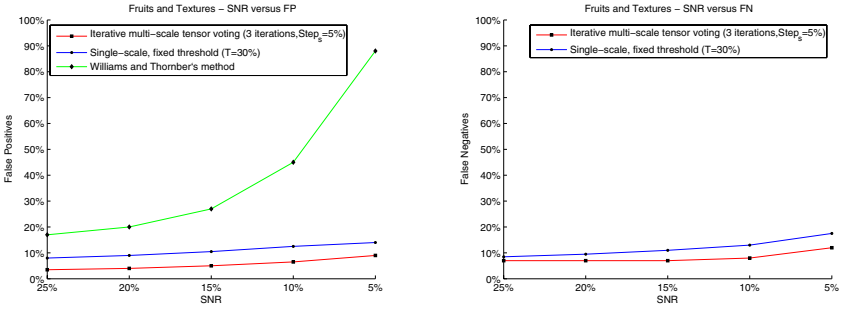


Fig. 7. Representative results based on our methodology: (a,d) avocado on bark at SNR equal to 20%, (b,e) pear on wood background at SNR equal to 15%, (c,f) pear on wood at SNR equal to 5%

Figure 6(b) shows the ROC curves using the proposed iterative, multi-scale adaptive thresholding scheme. The scale parameter σ varies from 2 to 20 (covering from 5% to 100% of the image), Step_s was equal to 5%, and T_σ was equal to 50% (i.e., the saliency curve must be above T_s in at least half of the processed scales). This allows structures to pop out in any region of the scale range. Significant improvements can be noted by comparing Figure 6(b) to Figure 6(a). In addition, the curve corresponding to SNR equal to 10% is closer to the ones corresponding to higher SNRs (i.e., 25%, 20% and 15%). This indicates that the iterative multi-scale adaptive thresholding approach deals with cluttered scenes much better. Fig. 7 shows three representative results using the iterative multi-scale tensor voting approach.

To compare our results with those given in [7], we have created plots of SNR vs FP, shown in Fig. 8(a). Specifically, Fig. 8(a) compares results obtained using the naïve approach (single-scale, fixed threshold at T=30% - Fig. 6(a)), the best result obtained by our iterative, multi-scale tensor voting scheme (i.e., 3 iterations using Step_s=5% - Fig. 6(b)), and the results reported in [7]. Since the results in [7] were not provided explicitly, we used a ruler over a hard copy of their plots to infer the values shown for their method in Fig. 8(a).

Fig. 8(b) is a plot of SNR vs FN. In this case, a direct comparison with [7] is not possible since they do not report FN rates. As it can be seen from the graphs, our iterative, multi-scale tensor voting approach shows improvements of more than 14% over [7] when SNR is equal to 25%, and improvements of almost 90% when SNR is equal to 5%, while keeping a low FN rate (i.e., eliminates noise



(a) (b)

Fig. 8. Plots of (a) SNR vs FP and (b) SNR vs FN. The iterative, multi-scale tensor voting approach outperforms the method in [7] (WT) as well as the naïve approach. Also, it has a low FN rate and performs consistently as SNR decreases).

without compromising figure information). Compared to the naïve approach, the iterative multi-scale approach improves figure vs noise discrimination by 5% on the average for all SNRs considered. The graphs also show a significantly smaller depreciation of performance as SNR decreases.

5 Conclusions and Future Work

We have presented a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Our approach uses an iterative scheme that removes noise conservatively using multi-scale analysis and re-voting. We tested our approach on data sets composed of real objects in real backgrounds. Our experimental results indicate that our method can segment successfully objects in images with up to twenty times more noise segments than object ones.

For future work, we plan to test our method on more challenging data sets including objects with open contours as well as multiple objects of the same and of different sizes. Moreover, we plan to devise a procedure for choosing Step_s and T_σ automatically at each iteration. Although the choice of Step_s did not seem to be very critical in our experiments, we feel that choosing this parameter in a more optimal way would probably help in certain situations.

References

1. Lowe, D.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence **31** (1987) 355–395
2. Ahuja, N., Tuceryan, M.: Extraction of early perceptual structure in dot patterns: integrating region, boundary, and component gestalt. CVGIP (1989) 304–356

3. Mohan, R., Nevatia, R.: Segmentation and description based on perceptual organization. CVPR (1989) 333–341
4. Ullman, S.: Filling-in the gaps: The shape of subjective contours and a model for their generation. Biological Cybernetics **25** (1976) 1976
5. Parent, P., Zucker, S.W.: Trace inference, curvature consistency, and curve detection. IEEE Trans. PAMI **11** (1989) 823–839
6. Ullman, S., Sha'ashua, A.: Structural saliency: The detection of globally salient structures using a locally connect network. 2nd. International Conference on Computer Vision - ICCV'88 (1988)
7. Williams, L., Thornber, K.: A comparison measures for detecting natural shapes in cluttered background. International Journal of Computer Vision **34** (2000) 81–96
8. Guy, G., Medioni, G.: Inference of surfaces, 3-d curves, and junctions from sparse, noisy 3-d data. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 1265–1277
9. Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Segmentation and Grouping. Elsevier Science (2000)
10. Medioni, G., Kang, S.B.: A Computational Framework for Segmentation and Grouping - Chapter 5. The Tensor Voting Framework. Prentice Hall (2005)

Fast Prediction Mode Decision Algorithm for H.264 Based on Hierarchical Mode Classification Framework

Cheng-dong Shen and Si-kun Li

School of Computer Science, National University of Defense Technology, ChangSha, China
shencd@163.com

Abstract: The latest video compression standard H.264 supports many intra and inter prediction modes. The mode with the optimal rate-distortion performance is determined using the powerful Lagrangian minimization tool such as rate-distortion optimization, which achieves highest possible coding efficiency but results in much higher computational complexity. In this paper, a fast prediction mode decision algorithm for H.264 is proposed. It is based on hierarchical mode classification framework which assists fast mode decision by pre-selecting the class for macroblock using the extracted spatial and temporal features of macroblock. Since tests for many modes of non-selected classes will be skipped, much computation of rate-distortion optimization can be saved. Experimental results show that the proposed method can reduce the execution time of mode decision by 85% on the average without perceivable loss in coding rate and quality.

Keyword: Mode decision, rate-distortion optimization, video coding, H.264.

1 Introduction

The newest H.264 video coding standard [1] can achieve much higher coding efficiency than the previous standards such as H.263 and MPEG-4. This is mainly due to more complicated approaches employed by H.264 encoder, especially the advanced prediction method supporting various inter and intra prediction mode. For the inter prediction, H.264 allows blocks of 7 different motion-compensation sizes and shapes which are 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4. And for intra prediction, H.264 offers 9 prediction modes for 4×4 luminance blocks and 4 prediction modes for 16×16 luminance blocks. To achieve as highest coding efficiency as possible, H.264 encoder uses complex mode decision technique based on rate-distortion optimization (RDO) that requires high computational complexity. Using RDO, all possible coding modes are undergone exhaustively resulting in dramatically increased computational complexity compared to previous standards. This makes H.264 difficult for real-time applications with low computational capability, such as mobile devices.

To reduce computational complexity of mode decision, recently many attempts have been made in developing fast intra mode prediction algorithms and fast inter mode prediction algorithms. F. Pan, et al. [2] proposed a fast intra-mode decision scheme with a pre-processing technique, which measures the edge direction of a given block

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 882 – 890, 2006.
© Springer-Verlag Berlin Heidelberg 2006

using Sobel operator and their histogram and thus adopts the edge direction to predict the possible mode, so as to reduce the number of modes to be checked. It assumes high correlation between the edge direction and intra mode but this assumption of edge direction is not always true. Kim et al. [3] uses two features, i.e. the sum of absolute residual difference (SATD) and the sum of absolute gradients (SAG), and a simplified RDO method to determine the best mode for 4x4 blocks. Some other works consider fast inter mode decision to speed up the variable block size motion estimation and RDO mode decision in H.264. Lim [4] selected several possible inter modes based on the texture homogeneity of the macroblock or sub-block. The performances of these methods highly depend on the threshold used and the video contents. Tu [5] proposed a method by first doing full search on four 8x8 sub-blocks. Two sub-blocks may then be merged according to the estimated error surface by experiments, which however is under the unimodal assumption and may vary sequence by sequence.

Different from the aforementioned methods, which individually develop fast intra or inter mode decision algorithms, this paper proposed a new prediction mode decision algorithm which is based on hierarchical mode classification framework. In the framework, spatial and temporal features of macroblock are exploited and according to some extracted feature parameters the most possible class can be selected before RDO mode decision is done. By skipping testing for prediction modes of the non-selected classes the computational complexity of mode decision can be significantly reduced while maintaining similar coding performance. The rest of the paper is organized as follows. In Section 2, we first review in brief prediction method and mode decision in H.264. Then in Section 3, proposed hierarchical framework for fast macroblock prediction mode decision is presented. The simulation results are shown in Section 4 and conclusions are made in Section 5.

2 Prediction Method and Mode Decision in H.264

Traditionally, motion estimation (ME) is performed only on fixed-size block such as 16×16 macroblock or 8*8 blocks. However, when the macroblock contains multiple objects and every object moves in different directions or when the macroblock lies on the boundary of a moving object, the motion vector (MV) getting from the simple ME will not be enough to represent true motions. So in H.264 multiple various block sizes for ME are supported in order to improve the prediction accuracy. It uses tree-structured hierarchical macroblock partitions as depicted in Fig.1. 16×16 pixel macroblocks can be broken into macroblock partitions, of sizes 16×8, 8×16, or 8×8. 8×8 partitions are also known as sub-macroblocks. Sub-macroblocks can be further broken into sub-macroblock partitions, of sizes 8×8, 8×4, 4×8, and 4×4. In general, selecting a mode with a large partition size means that a small number of bits for motion information are required. However, motion estimation may not be accurate resulting in generating a large number of bits for sending transform coefficients. On the other hand, selecting a mode with a small partition size may require a small number of bits needed to signal residual information but produce a large number of bits for motion vectors and side information. Therefore, the choice of coding mode has a significant impact on compression efficiency.

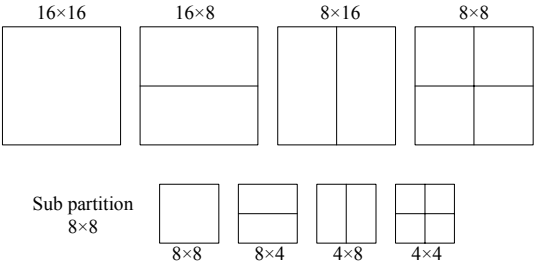


Fig. 1. Block modes for inter prediction

In contrast to previous standards, where intra prediction was conducted in the transform domain, intra prediction in H.264 is always conducted in the spatial domain using boundary pixels of upper and left blocks, which are previously decoded. In H.264 there are two intra prediction types called Intra4×4 and Intra16×16 which include 9 and 4 kinds of intra prediction modes respectively. In Intra4×4 prediction type, which is selected in non-homogeneous area, each 4×4 block of one macroblock is predicted from spatially neighboring samples as illustrated on the left-hand side of Fig.2. A prediction mode is a way to generate 16 predictive pixel values (named a to p) using some or all of the neighboring pixels A to M. The arrows and the numbers in the right-hand side of Fig.2 indicate directions of prediction and mode numbers respectively, where DC prediction (Mode 2) that is not directional mode is carried out using the average of A to L. For Intra16×16 prediction type, which are selected in relatively homogeneous area, 4 prediction modes such as mode 0 (vertical prediction), mode 1 (horizontal prediction), mode 2 (DC prediction) and mode 3 (plane prediction) are supported.

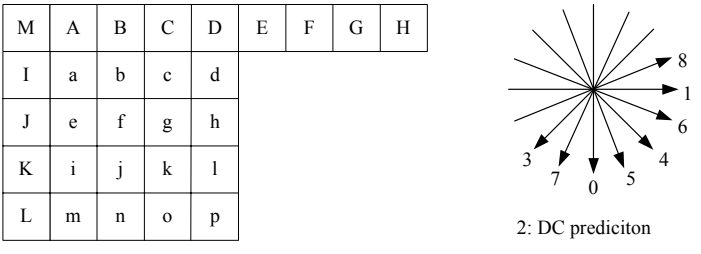


Fig. 2. Intra4×4 prediction modes

When dealing with mode decision for interframes, besides the multiple inter-modes with regards to block types, H.264 also supports skip mode and intra mode. The current JVT reference software [6] is based on a RDO framework for mode decision which tests all possible modes by evaluating the cost associated with each mode, and then selects the best mode with the smallest cost. The cost is formulated by following Lagrangian function:

$$J(s,c,MODE \mid QP,\lambda_{\mathrm{mod}e}) = SSD(s,c,MODE \mid QP) + \lambda_{\mathrm{mod}e} \cdot R(s,c,MODE \mid QP)$$

where QP is the quantization parameter, λ_{mode} is the Lagrange multiplier for mode decision, SSD is the sum of the squared differences between the original block luminance (denoted by s) and its reconstruction c, MODE indicates a mode out of a set of potential macroblock modes:{SKIP, 16×16, 16×8, 8×16, 8×8, Intra4×4, Intra16×16}, and R(s, c, MODE | QP) represents the number of bits associated with the chosen mode, including the bits for the macroblock header, the motion and all DCT coefficients. Note that, the RDO routine also operates on the sub-macroblock mode decision where candidate modes include 8×8, 8×4, 4×8, and 4×4, and Intra4×4 mode decision where candidate modes include all of nine modes as depicted in Fig.2. Using above RDO mode decision method, the computational cost is very high. Even without RDO, the complexity is still high since the encoder checks all inter and intra modes.

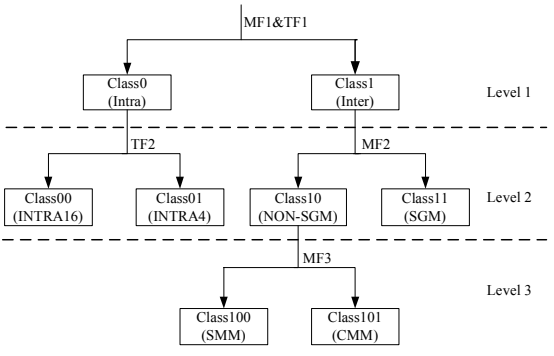
3 Proposed Fast Mode Decision Algorithm

The reason that H.264 supports so many prediction modes as introduced above is to adapt different temporal and spatial correlations that exist in various macroblocks, frames and video sequences, so as to better exploit these correlations and achieve smallest RDO cost. According to some features which are defined to represent the temporal and spatial correlations, all of the modes can be hierarchically classified by three levels. At each level, one class can be selected for the current macroblock by evaluating some extracted feature parameters. Finally, only modes in selected class should be tested in the RDO mode decision procedure instead of full modes test. This section will elaborate the mode classification framework, especially the selection and extraction of feature parameters, and the fast mode decision algorithm based on this framework.

3.1 Prediction Mode Classification

Since different modes will be used to exploit different temporal and spatial correlations, intuitively speaking, they can be classified in term of temporal and spatial features. We define five features to help mode classification, that is, three motion feature (MF) and two texture feature (TF), which represent temporal and spatial correlations respectively, denoted by MF1, MF2, MF3, TF1, and TF2. As depicted in Fig.3, the whole modes can be hierarchically classified by three levels using these defined features. At the first level joint motion and texture feature, namely MF1&TF1, is used to divide all the modes into two classes (Class0 and Class1) which correspond to intra prediction mode and inter prediction mode respectively. At the second level, the classification for Class0 and Class1 continues. For Class0, modes of intra type are further divided into Class00 (INTRA16) and Class01 (INTRA4) using texture feature TF2. And for Class1, modes of inter type are further divided into Class11 and Class10 using motion feature MF2, which will be used for macroblocks with stationary or global motion (SGM) and macroblocks without stationary or global motion (NON-SGM) respectively. Ultimately, modes of Class10 (NON-SGM) are classified as

two classes using motion feature MF3: Class100 which includes modes used for macroblock with simple motion (SMM, simple motion mode) and Class101 which includes modes for macroblock with complex motion (CMM, complex motion mode). Modes belong to class INTRA16, INTRA4, SGM, SMM, and CMM are listed in Fig.3. Note that, modes in class CMM are for sub-macroblocks. The goal of mode classification is that macroblock to be encoded can be assigned to one certain class before full mode decision by evaluating the motion feature and/or texture feature parameters of macroblock, so that only partial modes should be tested and lots of computation of RDO can be saved. Details about the selection and extraction of feature parameters will be presented in the following subsection.



INTRA16={Vertical, Horizontal, DC, Plane};
INTRA4={Vertical, Horizontal, DC, Diagonal down-left, Diagonal down-right, Vertical-right, Horizontal-down, Vertical-left, Horizontal-up};
SGM={SKIP, INTER16×16 (MV=0 or MV=PMV)};
SMM={INTER16×16, INTER16×8, INTER8×16};
CMM={INTER8×8, INTER8×4, INTER4×8, INTER4×4}

Fig. 3. Prediction mode classification

3.2 Extraction of Motion and Texture Feature Parameters

The key problem for mode classification-based fast mode decision algorithm is to select the appropriate feature parameters which are used to decide an appropriate mode class for current macroblock during the mode decision procedure. Good feature parameters should be easily computed with little computation overhead whereas spatial and/or temporal correlation is well captured. Based on this rule, we proposed the extraction method of parameters for texture features (TF1 and TF2) and motion features (MF1, MF2 and MF3), which are described respectively as follows.

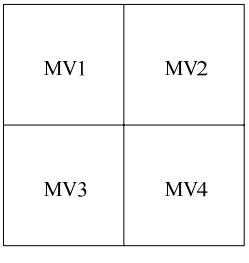
MF1: To get the feature to reflect the temporal domain correlation for Class1, motion search using 8×8 macroblock partition mode is performed to get four best matched blocks. To be more specific, four motion vectors are obtained using EPZS [7] algorithm with the quarter pixel accuracy. The four prediction residual blocks are 1/2 down sampled to 4×4 arrays and then Hadamard transform of 4×4 matrix size is applied to them. The reason for using Hadamard transform is due to its simplicity. The sums of absolute transform difference (SATD) of the four prediction residual blocks are computed respectively and the sum of them is defined as parameter for MF1, denoted as f_1.

TF1: To characterize the spatial domain correlation for Class0, we compute the sum of SATD of intra prediction residual using Hadamard transform. Representative modes from Class00 and Class01 are selected respectively to perform intra predictions and get the mode with the smallest prediction error. For Class00, three mode (DC, vertical and horizontal) are selected; and for Class01, five modes including DC, vertical, horizontal, diagonal down-left and diagonal down-right are selected. The sum of 16 SATD values for each 4×4 prediction residual blocks of one macroblock is computed as the TF1 parameter, denoted as f_2.

TF2: Based on the above test results of representative modes for Class00 and Class01, the texture features for mode decision between Class00 and Class01 are extracted as follows. From the test results of three modes of Class00, the mode with the smallest prediction error is selected and the sum of 16 SATD values of macroblock prediction residual is computed, denoted as f_3, which is used to reflect the prediction accuracy of Class00 and regarded as one texture feature parameter of TF2. In the same way another texture feature parameter f_4 is obtained from test result of Class01.

MF2: The motion features used to discriminate between class SGM and class NON-SGM can be extracted by computing the cost of SKIP mode and the cost of zero MV for 16×16 inter prediction mode, which are denoted as f_5 and f_6 respectively.

MF3: During the extraction of parameters for MF1, fast motion estimation is carried out for 8×8 macroblock partition mode. The resulting four motion vectors can be used to determine whether the motion of current macroblock is simple or complex. As depicted in Fig.4, the four motion vectors are MV1, MV2, MV3 and MV4, the distance of neighboring MVs is computed, denoted as MVD1, MVD2, MVD3 and MVD4. Then parameters for MF3 can be represented by f_7 and f_8, which are computed as the sum of horizontal neighboring MVs' distances and that of vertical neighboring MVs' distances respectively.



$$MVD1 = |MV1_x - MV2_x| + |MV1_y - MV2_y|$$
$$MVD2 = |MV3_x - MV4_x| + |MV3_y - MV4_y|$$
$$MVD3 = |MV1_x - MV3_x| + |MV1_y - MV3_y|$$
$$MVD4 = |MV2_x - MV4_x| + |MV2_y - MV4_y|$$
$$f_7 = MVD1 + MVD2$$
$$f_8 = MVD3 + MVD4$$

Fig. 4. Extraction of motion feature (MF3)

3.3 Fast Mode Decision Algorithm

Based on the feature-based mode classification, the most probable modes can be hierarchically determined by evaluating some motion and texture feature parameters prior to full search of all modes. This is the key idea of our proposed hierarchically framework for fast mode decision algorithm. This subsection will discuss the fast mode

decision algorithm which is based on proposed mode classification framework. As depicted in Fig.5, the critical part of fast mode decision algorithm is hierarchically selecting the proper mode class for current macroblock. The main process of the fast mode decision algorithm is summarized as follows.

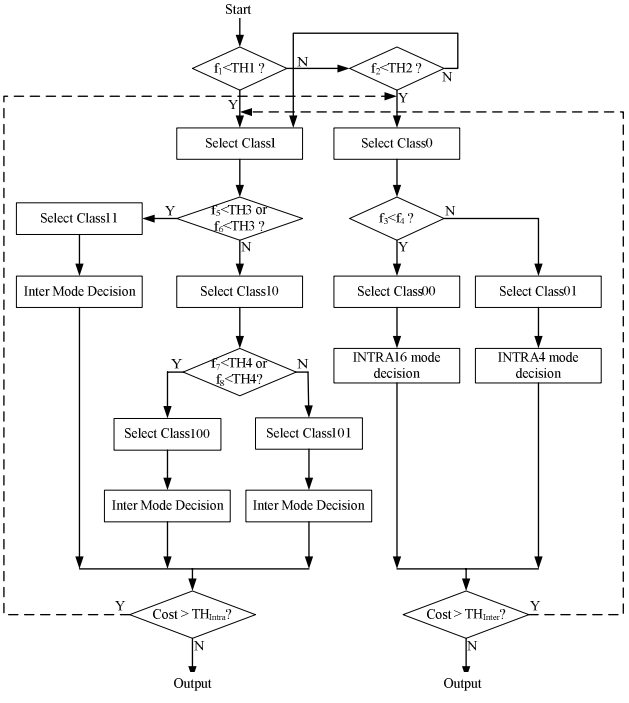


Fig. 5. Fast macroblock mode decision algorithm

Step1: Using the joint motion and texture feature, f_1 and f_2, to discriminate between Class0 and Class1. Generally, the cost of inter mode is comparative smaller than that of intra mode. If f_1 is smaller than one threshold (denotes as *TH1*) Class1 is selected and goto Step3; otherwise, compare f_2 with threshold *TH2*, if f_2<*TH2*, Class0 is selected and goto Step2, otherwise, Class1 is again selected and goto Step3.
Step2: Continue to select the class that the intra modes belong to. Compare f_3 with f_4, if the former is smaller Class00 is selected, goto Step5; otherwise, choose Class01 and goto Step6.
Step3: Compare f_5 and f_6 with one threshold (denoted as *TH3*), if one of them is smaller than *TH3* the macroblock can be treated as stationary or with global motion and Class11 is selected, goto Step7; otherwise, select Class10 and continues.
Step4: Compare f_7 and f_8 with one threshold (denoted as *TH4*), if one of them is smaller than *TH4* Class100 is selected; otherwise Class101 is selected, goto Step7.
Step5: Select the mode with smallest cost during extraction of f_3 and DC mode as the candidate intra prediction modes; compute their RDO costs and goto Step8. It should be noted that if the mode with f_3 is DC then only DC mode should be checked.

Step6: Select the mode with f_4, two modes adjacent to it and the DC mode as the candidate intra prediction modes; compute their RDO costs and goto Step8. Here note that if the mode with f_4 is DC then only DC mode should be checked.
Step7: The modes with regard to the class selected are chosen; compute their RDO costs and continues.
Step8: If the mode with smallest cost belongs to Class1, compare it with one threshold TH_{Intra}, if it is larger than TH_{Intra} it is probable that selecting Class1 (Inter) is erroneous and reselect Class0, return to Step2. If the mode with smallest cost belongs to Class0, compare it with one threshold TH_{Inter}, if it is larger than TH_{Inter} it is probable that selecting Class0 (Intra) is erroneous and reselect Class0, return to Step2.
Step9: The mode with the smallest cost is the final prediction mode.

It should be noted that fast mode decision for Intra prediction is performed in Step 5 and Step 6, due to heavy computation load of RDO mode decision for Intra modes. Also note that, the threshold values used in the algorithm are initially determined by test sequences learning practice and can be slightly modified during after completing one frame using the mode checking results.

In Step 8, we employ class switch scheme for Class0 and Class1. The use of this scheme can effectively alleviate the influence of erroneous class decision at first level.

Essentially, the above fast mode decision process is a demonstration of proposed mode classification framework. In the practical application of the framework, mode classification can be made at any level and other fast mode decision schemes can be utilized on selected mode class. For instance, mode classification is applied only on first level and fast mode decision algorithms proposed in other literatures for Class0 or Class1 are used.

4 Experimental Results

The fast mode decision algorithm has been integrated with the reference JVT software JM8.6 [6]. Test sequences and major encoding options are shown in Table 1. We compare proposed mode decision method with the RDO mode decision of JM8.6 in terms of computational complexity which is counted by encoding time for mode decision routine. The time saving results are given in Table 2. To demonstrate the encoding performance in terms of PSNR and bitrate, we also give the average PSNR gain (BDPSNR) and bitrate reduction (BDBR), which are recommended by [8]. The result is listed in Table 2.

Table 1. Test sequences and encoding conditions used in experiments

Test Sequence (QCIF, 300 frames)	Encoding Options
Foreman	QP: 28, 32, 36, 40 Search Range: 16 Number of Reference: 5 Coding Structure: IBBPBBPBB Others: RDO, CABAC, Hadamard transform
Stefan	
Mobile	
Container	
News	

Table 2. Performance of proposed method

Sequence	Foreman	Stefan	Mobile	Container	News	Average
BDBR (%)	2.36	1.55	3.12	0.88	1.25	1.83
BDPSNR(dB)	-0.11	-0.09	-0.12	-0.06	-0.08	-0.09
TimeSaving (%)	89.13	85.78	78.32	91.35	82.46	85.41

From these results we observe that proposed method is very effective in reducing the RDO mode decision time. 78.32% to 91.35% (average 85.41%)complexity reduction can be achieved versus the reference software, while its compression efficiency loss in terms of BDBR and BDPSNR [8] is very negligible, which are only 1.83% and 0.09dB, respectively.

5 Conclusions

In this paper we developed a hierarchical framework for fast macroblock prediction mode decision based on mode classification. This framework can be effectively used in fast macroblock mode algorithms. Experimental results demonstrate that the fast mode decision method using the proposed framework can reduce the execution time significantly while maintaining coding efficiency.

References

1. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 I ISO/IEC 14496-10 AVC)", ITU-T ISO/IEC , 2003.
2. Feng Pan and Xiao Lin et. al., “Fast mode decision for intra prediction”, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 7th Meeting Pattaya II, Thailand, March 2003.
3. C. Kim, H. Shih, and C. Kuo, “Multistage mode decision for intra prediction in H.264 codec,” in Proc. SPIE Symp. Visual Communications and Image Processing 2004, 2004, p 355-363
4. K.P. Lim, S. Wu, D.J. Wu, S. Rahardja, X. Lin, F. Pan, and Z.G. Li, "Fast Inter Mode Selection", JVT-I020, Joint Video Team (JVT) of ISOIIEC MPEG & ITU-T VCEG, 9th meeting, 2003
5. Y.K.Tu, J.F.Yang, Y.N.Shen, M.T.Sun, “Fast variable size block motion estimation using merging procedure with an adaptive threshold”, Proc. Int. Conf. Multimedia and Expo, Vol.2, pp.789-792, Jul. 2003.
6. H.264 JVT reference software, http://bs.hhi.de/~suehring/tml/download.
7. A. Tourapis, O. C. Au, and M. L. Liou, “Highly efficient predictive zonal algorithm for fast block-matching motion estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 12, pp. 934-947, Oct. 2002.
8. G. Bjontegaard, “Calculation of Average PSNR Differences between RD-curves,” Doc. VCEG-M33, Apr. 2001.

Camera Self-calibration in Scheimpflug Condition for Air Flow Investigation

Hanene Louhichi[1,3], Thierry Fournel[1], Jean Marc Lavest[2], and Habib Ben Aissia[3]

[1] Laboratoire Traitement du Signal et Instrumentation, UMR 5516 CNRS, Université Jean Monnet, 18 rue du Pr B. Lauras 42000 Saint Etienne, France
[2] Laboratoire de Sciences des Materiaux pour l'Electronique et d'Automatique, UMR 6602 CNRS, Université Blaise Pascal, 63177 Aubiere Cedex, France
[3] Unité de Métrologie en Mécanique des Fluides et Thermiques, UR 11-17, Université de Monastir, Iben Eljazzar, 5019 Monastir, Tunisie

Abstract. Camera calibration is a necessary preliminary step to ensure high quality measurement in a section of a flow. The bundle adjustment technique is based on redundancy resulting from multi-viewing the calibration target. In this paper we proposed a well-adapted protocol for self-calibrating a Scheimpflug model. Such a protocol allows calibration by hand positioning a laser printed target. Experimental results indicate that with only 5 views we can accurately calibrate a Scheimpflug camera in Stereoscopic Particle Image Velocimetry condition.

Keywords: self-calibration, calibration, Scheimpflug condition, Stereoscopic Particle Image Velocimetry.

1 Introduction

The Scheimpflug condition enables to focus a video-camera when oblique viewing at low aperture number. It is used in Stereoscopic Particle Image Velocimetry (SPIV) for measuring the three-dimensional velocity field in a section of a flow [1]. This condition is performed by tilting the sensor (or the lens) such that the object plane, the lens plane and the image plane intersect in a common line (figure 1). In this case, the object plane is set as close as possible to the measurement plane which is the median plane of a light sheet, and the three velocity components are reconstructed from mapping functions. One mapping function per camera is used to relate any 3D location to its image. The parameters of each mapping function are recovered at the calibration stage.

A precise target is usually placed in a reference plane for the calibration of SPIV cameras. The residual misalignment between the reference plane and the measurement plane has to be corrected when the studied flow is non laminar [2]. If the target is a planar one, it has to be accurately moved to different locations in depth. Other multi-level targets, on which different z-positions are present, are available for the special case of SPIV calibration. In [3], accurate axial displacements are required by the calibration protocol. All these calibration

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 891–900, 2006.
© Springer-Verlag Berlin Heidelberg 2006

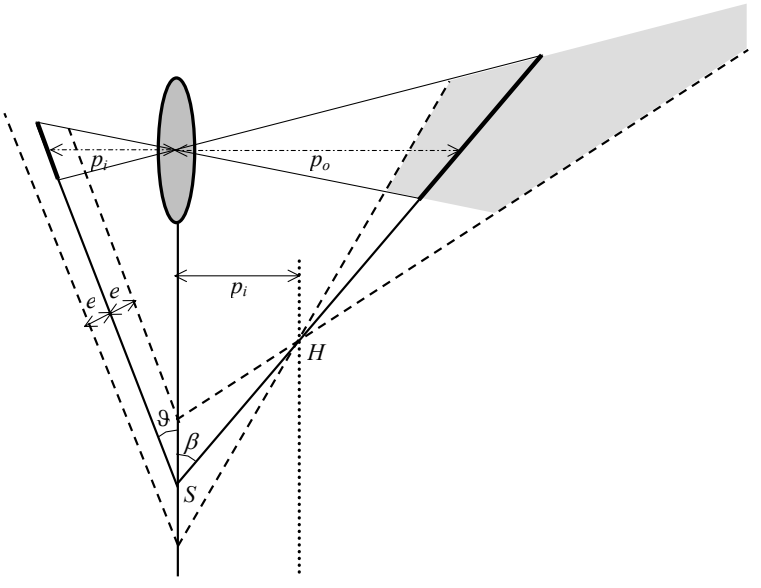


Fig. 1. Cross-section of Scheimpflug set-up: the image plane (rotated with angle θ), the lens plane and the object plane (viewed under angle β) intersect in Scheimpflug line (S). The near and far limits of the depth of field (dashed lines) intersect in Hinge line (H). These planes are respectively conjugated to limiting planes parallel and either side of the image plane at distance e given by the length of acceptable confusion: for images focused at either of the limiting planes, the shape of confusion on the image plane will be the maximum allowable.

methods may be unstable if the positioning of the target is not accurate enough. Besides a method using a single image of a dot grid target was proposed in [4] but given the focal length and sensor pixel pitches of the PIV camera.

Self-calibration does not use such additional accurate informations on the recording system. A non linear least mean square procedure allows the (bundle) adjustment of the vector of the calibration parameters after their rough initialization. The coordinates of the calibration points are included in this vector. Thus, metrological targets and controlled remote XYZ devices are no more required when self-calibrating Scheimpflug cameras. This paper aims to propose a protocol well-adapted to Scheimpflug self-calibration. The bundle adjustment technique in Scheimpflug condition is first described in section 2. This technique is based on redundancy resulting from multi-viewing the calibration target. Section 3 explains how recording a minimum of views for convergence. Experimental results are reported in section 4: firstly the influence of the number of views are studied by simulation, secondly the quality of real reconstructions is discussed.

2 Bundle Adjustment Technique in Scheimpflug Condition

Bundle adjustment is a videogrammetric technique used for (self-)calibrating video-cameras or recovering the 3D structure of a scene [5]. Each mapping function corresponds here to a camera model determined by intrinsic and extrinsic camera parameters. The intrinsic ones are the parameters of the camera itself. The extrinsic ones are the parameters of the location of the camera with respect to the world coordinate system. The distinction between intrinsic and extrinsic parameters allows high redundancy when recording a set of views of the target. Indeed the difference between the number of equations and the number of unknown parameters increases with the number of views because the intrinsic parameters remains constant. Thus redundancy becomes high enough to self-calibrate i.e. to introduce the coordinates of the calibration points into vector $\mathbf{\Phi}$ of parameters to be adjusted by least mean squares. Thus the calibration points are estimated simultaneously with the camera parameters by minimizing image residual errors as:

$$\begin{cases} \varepsilon_U = u - u \\ \varepsilon_V = v - v \end{cases} \tag{1}$$

where (u, v) are the pixel coordinates of the projection of a calibration point given by the camera model and (u, v) are the measured ones.

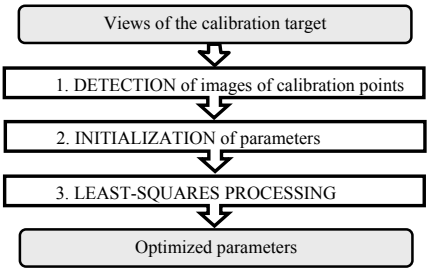


Fig. 2. Steps of a bundle adjustment technique

2.1 Camera Model

The camera model has to be determined as coordinates (u, v) in (1) depends on its parameters. The Scheimpflug camera model can be derived from the pinhole one. According to the pinhole model, point $\mathbf{X} = (X, Y, Z, 1)^T$ (in world coordinate system) is projected to $\mathbf{x_P} = (x, y, z, 1)^T$ in the (first) camera coordinate system (figure 3) as:

$$\mathbf{x_P} \approx \mathbf{P_P} \begin{pmatrix} \mathbf{R_P} & \mathbf{t_P} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{X} + \delta\mathbf{x} \tag{2}$$

where the equivalence relation denoted by symbol $\approx$ means equality up to a non zero scale factor. $\mathbf{R_P}$, $\mathbf{t_P}$ are the rotation and translation matrices relating the

world coordinate system to that of the camera, $\mathbf{P_P}$ is the perspective projection to the frontal plane and $\delta\mathbf{x}$ is the distortions term. In usual conditions, distortions are usually reduced to radial ones given by adimensional coefficients a_1,a_2,a_3 [6] such as:

$$\begin{cases} \delta x = x(a_1 r^2 + a_2 r^4 + a_3 r^6) \\ \delta y = y(a_1 r^2 + a_2 r^4 + a_3 r^6) \end{cases} \tag{3}$$

where $r = \sqrt{x^2 + y^2}/p_i$ is the normalized distance from the image centre.

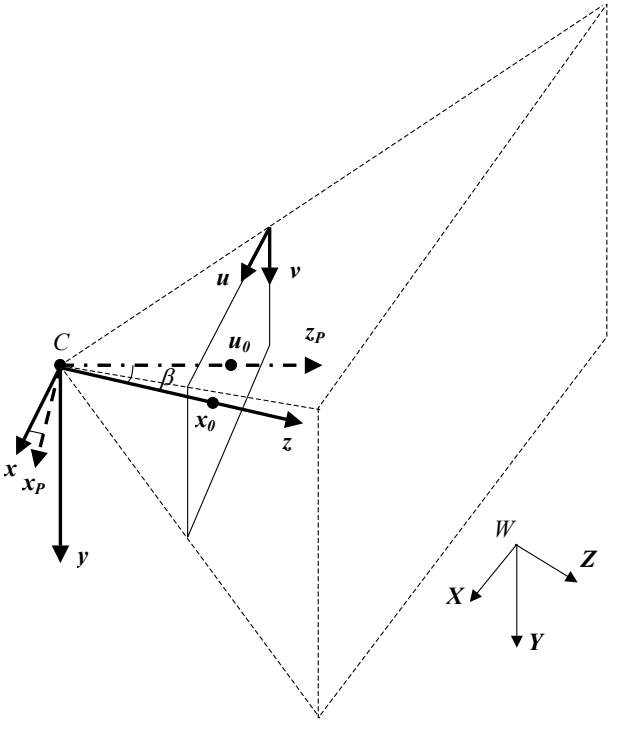


Fig. 3. Coordinate systems in Scheimpflug condition according to the "painter convention": the image plane is placed front of projection centre C by symmetry with respect to this point

The Scheimpflug model induces tilt angle θ of the image plane as supplementary parameter. The tilt is performed according to the Scheimpflug condition. This condition satisfied when the image plane, the lens plane and the object plane intersect in a common line, ensures that conjugaison formula $\frac{1}{f} = \frac{1}{p_i} + \frac{1}{p_o}$ is verified. Where p_i is the image distance and p_o is the working distance. With nominal magnification $M = \frac{p_i}{p_o}$ and view angle β, tilt θ is given by (figure 3):

$$\theta = \arctan(M \tan(\beta)) \tag{4}$$

Image $\mathbf{x}$ can be deduced from $\mathbf{x_P}$ by:

$$\mathbf{x} \approx \begin{pmatrix} \mathbf{R_S} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{P_S}\, \mathbf{x_P} \tag{5}$$

where $\mathbf{P_S}$ is the perspective projection in the image plane and $\mathbf{R_S}$ is the Scheimpflug rotation of the camera coordinate system which leads the optical axis to merge in the normal of the tilted image plane. The tilted optical axis intersects the image plane in Scheimpflug image centre $\mathbf{x}_0$ and defines an ideal pinhole camera without distortions:

$$\mathbf{x} \approx \mathbf{P_{PS}} \begin{pmatrix} \mathbf{R_{PS}} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{X} \tag{6}$$

where $\mathbf{P_{PS}} = \mathbf{P_S}\, \mathbf{P_P}$ and $\mathbf{R_{PS}} = \mathbf{R_S}\, \mathbf{R_P}$.

2.2 Multi-views and Optimization of Calibration Points

Residual errors (1) are minimized by refinement of the following parameters: $3n$ coordinates (X_j, Y_j, Z_j) of n calibration points and the intrinsic / extrinsic parameters of the camera model (mapping function). From the last section 2.1, the intrinsic parameters consist in tilt angle θ and the pinhole parameters: image distances p_{ix}, p_{iy} expressed in pixel, pixel coordinates (u_0, v_0) of the image centre and distortions coefficient. m different positions of the target give $6m$ extrinsic parameters: Euler angles $\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)}$ of rotation $\mathbf{R_P}$ and components $t_x^{(k)}, t_y^{(k)}, t_z^{(k)}$ of translation $\mathbf{t_P}$. The set of the parameters can now be listed: the intrinsic parameters of the camera, the extrinsic ones and the coordinates of the calibration points.

$\mathbf{\Phi_{8+6m+3n}} = (p_{ix}, p_{iy}, u_0, v_0, a_1, a_2, a_3, \theta,\ \alpha^{(1)}, \beta^{(1)}, \gamma^{(1)}, t_x^{(1)}, t_y^{(1)}, t_z^{(1)}, \cdots,$
$\alpha^{(m)}, \beta^{(m)}, \gamma^{(m)}, t_x^{(m)}, t_y^{(m)}, t_z^{(m)},\ X^{(1)}, Y^{(1)}, Z^{(1)} ..., X^{(m)}, Y^{(m)}, Z^{(m)})^T$

Redundancy given by the difference between the number of equations of type (1) and the number of unknown parameters is equal to:

$$r = 2mn - (8 + 6m + 3n) \tag{7}$$

Thus, even if the calibration point coordinates are included in vector $\mathbf{\Phi}$ of parameters, recording between 3 and 10 views (depending on number n of the calibration points) garantees a redundancy relevant for least mean square procedure.

2.3 Solving the Problem

The problem is to estimate the vector $\mathbf{\Phi}$ by minimizing the sum of the square residual errors:

$$S = \sum_{i=1}^{n} \sum_{j=1}^{m} (\varepsilon_{u_{i,j}}^2 + \varepsilon_{v_{i,j}}^2) \tag{8}$$

from image coordinates $(u_{i,j}, v_{i,j})$ of n calibration points (X_j, Y_j, Z_j), measured in m views. As residual errors are non-linear functions of $\mathbf{\Phi}$, the minimization is a non-linear optimization problem. One way of solving the problem is to linearize (1) with some initial value $\mathbf{\Phi}^{(0)}$ and solve for $\Delta\mathbf{\Phi}$ then to add $\Delta\mathbf{\Phi}$ to $\mathbf{\Phi}^{(0)}$ as the new initial value and repeat the process until a certain convergence is satisfied. The $2mn$ linearized residual error equations (1) can be written in matrix form:

$$\mathbf{E} = \mathbf{L} + \mathbf{J}\,\Delta\mathbf{\Phi} \tag{9}$$

where $\mathbf{L}$ is the residual error matrix at $\mathbf{\Phi} = \mathbf{\Phi}^{(0)}$ and $\mathbf{J}$ its jacobian matrix at this point. The least squares solution to (9) is given by:

$$\Delta\mathbf{\Phi}(\mathbf{J}^{\,T}\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^{\,T}\mathbf{W}\mathbf{L} \tag{10}$$

where $\mathbf{W}$ is the weight matrix of the measurements. The estimates of the residual vector $\mathbf{E}$ can be computed as

$$\hat{\mathbf{E}} = [\mathbf{I} + \mathbf{J}(\mathbf{J}^{\,T}\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^{\,T}\mathbf{W}]\mathbf{L} \tag{11}$$

and the estimate of the so called *standard error of unit weight*, which is the *a posteriori* estimation of the noises σ_0 of the image coordinates if the model is correct and there are no system errors:

$$\hat{\sigma}_0^2 = \frac{\hat{\mathbf{E}}^T\mathbf{W}\hat{\mathbf{E}}}{r} \tag{12}$$

and the estimate of the *covariance* matrix of the parameters $\mathbf{\Phi}$:

$$\hat{\mathbf{C}}_{\Phi} = (\mathbf{J}^{\,T}\mathbf{W}\mathbf{J})^{-1} \tag{13}$$

For each individual parameter Φ_k, the estimate of its precision or the standard deviation is given by:

$$\hat{\sigma}_{\Phi_k}^2 = \hat{\sigma}_0^2\,\hat{c}_{kk} \tag{14}$$

3 Proposed Protocol for Scheimpflug Self-calibration

In Scheimpflug condition, the planes of the near and far limits of the depth of field are no more parallel to the plane of sharp focus. The region of acceptable definition is in fact wedge-shaped [7]. The sharp edge of the wedge is the Hinge line defined as the intersection between the object plane and the plane parallel to the image plane containing the projection centre (figure 1). In standard PIV conditions, images remain focused provided that the calibration target is not rotated more than about 10° in the centre of the field of view. This limit decreases when the target is moved towards the Hinge line side. Thus, the target can be hand-positioned at this side in the depth of field, in a corner of the field of view. Then, in-plane translations can be roughly applied on the target in order to cover the other corners and the centre of the field of view. m_2 views recorded at the corners complete m_1 views recorded at the centre, in order to get a good

estimate of distortions coefficients a_1, a_2, a_3 (figure 4). The views recorded at the center can be deduced from each other by in-plane rotation. A simple printed flat target of n dots can serve as calibration object for the bundle adjustment. The coordinates of the calibration points are optimized together with the camera parameters from the $m = m_1 + m_2$ views of the target. A Levenberg-Marquardt procedure can be applied to initial values of the parameters after detection of the images of the calibration points (accordingly to the scheme of figure 2).

For initialization, coordinates $X^{(i)}$, $Y^{(i)}$, $Z^{(i)}$ of the calibration points are directly read from the print file of the target. For the parameters of the camera model (see 2.1) one can proceed as follows. Image distance p_{ix}, p_{iy} are obtained by dividing specified focal length f by pixel sizes dx, dy respectively. Image centre u_0, v_0 are initialized with the pixel coordinates of the middle of the matrix sensor. Distortion coefficients a_1, a_2, a_3 are set to zero. Scheimpflug tilt θ is computed from (4). Extrinsic parameters $\alpha^{(j)}$, $\beta^{(j)}$, $\gamma^{(j)}$, $t_x^{(j)}$, $t_y^{(j)}$, $t_z^{(j)}$ are obtained with DeMenthon and Davis' algorithm [8].

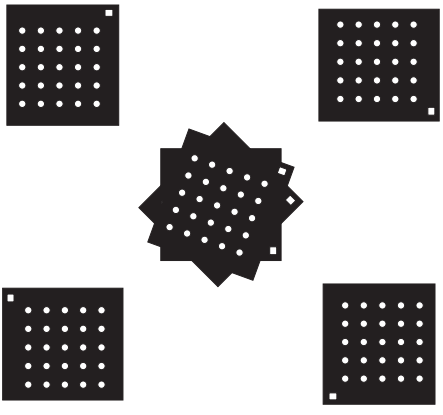


Fig. 4. Superimposition of side-views and central views of the calibration target in the field

4 Experimental Results

Bundle adjustment has been claimed to simplify the calibration of Scheimpflug cameras by handpositioning a printed flat target in a sheet of the depth of field. However, a critical points has to be studied for this purpose: how many views are really useful ? The effectiveness of the proposed protocol was tested with a 1/2" CCD camera having 1024×768 pixels of size $6.25\,\mu$ m $\times 6.25\,\mu$ m. Magnification was set to $M = 0.05$ and viewangle to $\beta = 45°$ giving Scheimpflug tilt $\theta = 2.86°$. A flat target of 5×5 dots provided $n = 25$ calibration points and redundancy $r = 44m - 83$. Two views ($m = 2$) are then required at minimum. Simulated data computed for $f = 50$ mm ($f/dx = 8000$ pixel) allowed testing the sensitivity of both the location error on calibration point coordinates and the number of views.

The protocol was applied in similar conditions on real data, with a laserprinted calibration target. In this case, a $f = 24$ mm lens ($f/dx = 3840$ pixel) having higher optical distortions was also used for testing the quality of the calibration.

4.1 Simulated Data

Handpositioning was simulated by randomly selecting the pose of the calibration target within 5° in rotation and 10 mm in translation with respect to the focus plane. Distortions rate was fixed such that it represents an amount of 2.5 pixel at the edges of images ($a_1 = 3$). For each test, 100 trials were done: initial values were randomly choosen according to a uniform distribution in the range of [40°, 50°] for β, [45 mm, 55mm] for f and in a circle of radius 30pixels around the image centre for u_0, v_0.

For this study, the image detection error was here assumed to be $\sigma_{uv} = 0.02$ pixel. The error on calibration points was set to σ_{XYZ}=1 mm, this value being easy to reach in practice. An amount of $m = m_1 + m_2$ views were computed, with a single view at the center until $m = 5$ views and with a single view per corner and several rotations at the centre for $m > 5$ views. As previously, the estimates of both image distance p_{i_x} and Scheimpflug tilt θ are reported with their standard deviations in table 1. The parameters are well recovered from $m = 5$ views. For $m = 5$, we obtained $a_1 = 2.71 \pm 0.30$, $a_2 = 0.03 \pm 0.01$, $a_3 = 0.05 \pm 0.02$ and $u_0 = 511.81 \pm 2.13$ pixel, $v_0 = 384.68 \pm 1.06$ pixel. The results are similar when using $m = 10$ views.

Table 1. Sensitivity to the number of views for σ_{XYZ}=1 mm and σ_{uv}=0.02 pixel

m	p_{i_x}(pixel)	$\sigma_{p_{i_x}}$(pixel)	θ(degree)	σ_θ(degree)	k
2	8364.92 ± 128.34	8.61 ± 0.45	3.74 ± 0.24	0.21 ± 0.16	0.91 ± 0.04
5	8002.29 ± 6.36	0.57 ± 0.06	2.86 ± 0.01	0.27 ± 0.02	1.01 ± 0.00
10	8001.92 ± 11.13	0.79 ± 0.03	2.86 ± 0.02	0.19 ± 0.03	1.01 ± 0.00

4.2 Real Data

The 25 calibration dots were laserprinted at 600 dpi on a paper sheet which was sticked on a metal plate. This target was hand positioned at different locations in the field as mentioned in section 3. At each position, one view was recorded: one view per corner ($m_2 = 4$) and six views at the centre after rotations by step of 60°. Self-calibration was done by using all the views ($m = 10$) then by using a subset reduced to the four side-views and only one central view ($m = 5$). Tests were performed at nominal magnification $M = 0.05$ with increasing distortions rate, by mounting a standard $f = 50$ mm lens then a standard $f = 24$ mm lens. The values obtained for the intrinsic parameters are given in tables 2 and 3 respectively. For f=50 mm and f=24 mm, the residuals are about 0.02 pixel for $m = 10$ views and 0.03 pixel for $m = 5$ views. The results indicate a good precision in the estimate of each intrinsic parameter. The confidence intervals

Table 2. Results of calibration for f=50mm

$m = 10$			$m = 5$		
Parameters		σ	Parameters		σ
p_{i_x}(pixel)	8075.32	0.81	p_x(pixel)	8168.12	0.43
p_{i_y}(pixel)	8075.32	0.81	p_{i_y}(pixel)	8168.12	0.43
u_0(pixel)	500.36	0.64	u_0(pixel)	502.12	0.52
v_0(pixel)	368.61	0.35	v_0(pixel)	367.26	0.39
a_1	3.91	0.84	a_1	4.52	0.39
a_2	0.82	0.03	a_2	0.91	0.06
a_3	0.26	0.04	a_3	0.31	0.01
θ(degree)	2.87	0.13	θ(degree)	3.01	0.17
Residual=0.021 pixel			Residual=0.034 pixel		

Table 3. Results of calibration for f=24mm

$m = 10$			$m = 5$		
Parameters		σ	Parameters		σ
p_{i_x}(pixel)	3801.37	0.41	p_{i_x}(pixel)	3819.42	0.23
p_{i_y}(pixel)	3801.37	0.41	p_{i_y}(pixel)	3819.42	0.23
u_0(pixel)	508.25	0.38	u_0(pixel)	510.22	0.52
v_0(pixel)	383.93	0.27	v_0(pixel)	389.17	0.12
a_1	8.93	0.37	a_1	9.23	0.42
a_2	1.49	0.21	a_2	1.61	0.14
a_3	1.38	0.14	a_3	1.08	0.17
θ(degree)	2.69	0.27	θ(degree)	2.65	0.21
Residual=0.028 pixel			Residual= 0.031 pixel		

obtained for $m = 5$ and $m = 10$ views intersect well for u_0, v_0, a_1, a_2, a_3, θ. The image distance estimated from $m = 5$ views differs from that estimated from $m = 10$ views with a deviation inferior to 1.2 % of the focal length for the f=50 mm lens (0.5 % for the f=24 mm lens). Concerning the calibration points, all the optimized Z-coordinates are inferior to 2 μm which agrees with the flatness of the actual calibration target.

5 Conclusion

Scheimpflug condition is required when oblique imaging at low aperture number of an object plane. The obvious advantages of the self-calibration method are that is not necessary to know the target motion in space and to avoid metrological target whose 3D coordinates are known with high precision. It has proven to be the most robust and accurate calibration as in SPIV application. As in standard condition of computer vision a precision of 0.02 pixel [9] in image detection is required for bundle adjustment of the Scheimpflug model. A laser printed target can be used, calibration points are adjusted together with camera parameters.

This paper propose an easy protocol for Scheimpflug self-calibration. Because images remain focused in a thin volume, multi-viewing can be here reduced to in-plane translations applied on the target in order to cover the other corners and the centre of the field of view. The obtained experimental results confirm that only 5 views are sufficient to accurately calibrate the Scheimpflug model.

References

1. K.D. Hinsch, H. Hinrichs, A. Roshop and F. Dreesen, Holograpic and stereoscopic advance in 3D PIV. Holographic Particle Image Velocimetry *Proc. of Fluids Engineering Division, American Society of mechnical Engineers* vol 148, ed E P Rood (Washington, DC:ASME) p 33-6, 1993
2. S. Coudert and J.P. Schon, Back-projection algorithm with misalignment correction for 2D3C stereoscopic PIV *Meas. Sci. Technol.***12** 1371-81, 2001
3. S. Soloff, R. Adrian and Z.C. Liu, Distortion Compensation for generalised stereoscopic particle image velocimetry *Meas. Sci. Technol.* **8** 1441-54, 1997
4. G. Qunot, A. Rambert, F. Lusseyran and P. Gougat, Simple and accurate PIV camera calibration using a single target image and camera focal length *4th Int. Symp. on Particle Image Velocimetry, Gottingen* paper 1040, 2001
5. W. Triggs, P. Mclauchlan, R. Hartly and Fitzgibbon, Bundle adjustment: a modern synthesis (Verlag: springer), 2000
6. American Society for Photogrammetry Manual of photogrammetry *4th edition*, 1984
7. M. Merklinger, View Camera Focus and Depth of Field: VIEW CAMERA magazine pp 56-58, 1996
8. L.S.Davis and D. Dementhon, Model based object pose in 25 lines of code*Int Journal of computer vision* **15(2)** 123-141, 1995
9. JM. Lavest, M.Viala and M. Dhome, Do we really need an accurate calibration pattern to acheive a reliable camera calibration *Proc. of ECCV98* (Freiburg: Germany) p 158-174, 1998

Author Index

Abdi, Hervé II-447
Abidi, Besma I-401
Abidi, Mongi I-401
Adamo-Villani, Nicoletta I-70
Agu, Emmanuel I-505
Ahn, Jaewoo II-79
Aissia, Habib Ben II-891
Akgül, Ceyhun Burak II-393
Al Muhit, Abdullah I-28
Al-Osaimi, Faisal R. I-90
Albu, Alexandra Branzan I-771
Albuquerque, Antonia Lucinelma Pessoa II-657
Amini, Nerssi Nasiri I-660
Andres, Eric II-514
Aquilio, Anthony S. I-891
Arias-Estrada, Miguel II-502, II-564
Arseneau, Shawn I-363
Awad, George I-495
Aykac, Deniz II-824

Bab-Hadiashar, Alireza I-445
Baek, Nakhoon II-49
Bailey, Ken II-427
Bais, Abdul I-202
Baker, Chris II-534
Baker, Jason I-293
Barat, C. II-364
Barcellos-Hoff, Mary Helen I-518
Barman, Sarah II-427
Barneva, Reneta I-548
Bascle, Bénédicte I-793, I-803
Bashar, M.K. II-136
Basu, Anup I-212, I-222, I-253
Basu, D.K. I-353
Bayerl, Pierre I-741
Bazin, A.I. I-141
Bebis, George I-40, I-122, II-870
Beck, Cornelia I-741
Behrendt, Stephan II-467
Beneš, Bedřich I-70, II-637
Bennamoun, Mohammed I-90, I-860
Bennett, Mike II-457
Berssenbrügge, Jan I-343
Bert, Julien I-411
Bertoline, Gary R. II-637
Bertram, M. II-186
Bhattarai, Bibek Dev II-678
Bielamowicz, Steven I-761
Blake, Edwin I-465
Blauensteiner, Philipp II-607
Bobach, T. II-186
Boudon, F. I-751
Boyer, Vincent I-393
Boyle, Richard I-122, II-851
Brimkov, Valentin E. I-548
Brisbin, Matt I-70
Brooks, Jeremy C. I-891
Brox, Thomas II-59
Brun, Luc II-306
Buatois, Luc I-383
Burgeth, Bernhard I-455

Calway, Andrew II-276
Cassagnabère, Christophe II-265
Caumon, Guillaume I-383
Chai, Yanmei I-80, I-579
Chan, Ming-Yuen I-273, I-528
Chang, Hang I-518, II-586
Chang, Tzyh-Chyang II-31
Chanwimaluang, Thitiporn II-19
Chekhlov, Denis II-276
Chen, Baoquan I-50
Chen, Cheng II-790
Chen, Chi-Jen II-31
Chen, Shuyi I-161
Cheng, Irene I-253
Chi, Zheru I-579
Chien, Yu-Ren I-170
Chiou, Jia-Wei II-780
Cho, Hwan-Gue II-576
Cho, Seongwon II-196
Choi, Tae-Sun II-166
Chong, Jong-Wha II-554
Chu, Chee-hung Henry I-19
Chuah, Teong Chee I-28
Chum, O. II-802

Chun, Bong-Kyung II-576
Chung, Ronald I-303
Chung, Sun-Tae II-196
Clarke, James II-427
Cœurjolly, David II-205, II-514
Cohen, Isaac I-783
Cole, Ronald I-731
Cole, T. I-141
Colombano, Silvano II-851
Coogan, Thomas I-495
Cooperstock, Jeremy R. I-363
Corlin, R. I-60
Costantini, Roberto II-245
Cox, Susan I-373

Da Silva, D. I-751
Daggubati, Lakshmi Pavan II-698
Daliri, Mohammad Reza I-712
Damelin, S.B. I-1
Damiand, Guillaume II-235
Darbon, Jérôme II-393
de Aguiar, Edilson I-485
de Calignon, Martin Braure II-306
de Vieilleville, François II-688
Debled-Rennesson, I. II-480
Debrunner, Chris II-534
Dembélé, Sounkalo I-411
Dexet, Martine II-514
Dhand, Harsh II-698
Diepold, Klaus II-596
Direkoğlu, Cem I-243
Domenjoud, E. II-480
Donate, Arturo II-750

Eikermann, Martin I-343
Eom, Doo-seop II-146
Erol, Ali I-122

Fahad, Ahmed II-286
Fan, Guoliang II-19, II-490, II-790
Fang, Hui II-227
Felsberg, Michael II-374
Feng, David I-579
Fihl, P. I-60
Florez, Jose F. I-421
Fournel, Thierry II-364, II-891
Fuchs, Laurent II-235

Gain, James I-465
Gall, Juergen II-59
Gao, Shanshan II-106
Gharai, S. I-353
Gee, Andrew P. II-354
Gee, Timothy F. II-835, II-843
Ghent, John II-334
Ghys, Charlotte I-793, I-803
Gibson, Gavin J. II-405
Gleason, Shaun S. II-824
Goddard, James II-835
Godin, C. I-751
Goebel, Randy I-253
Goh, Wooi-Boon I-475
Gooding, Sean II-534
Gorbach, Alexander M. II-835
Goto, Satoshi II-554
Govindasamy, V. Priya II-824
Gregor, Jens II-824
Gu, Jason I-202
Guan, Weiguang I-102
Guo, Fenghua II-98

Haghjoo, Mostafa I-660
Hahn, Hernsoo II-116, II-126
Hahn, James K. I-761, I-901
Han, Dongil II-39
Han, Ju I-518
Han, JungHyun II-146
Han, Junwei I-495
Han, Youngjoon II-116, II-126
Hanbury, Allan II-255, II-607
Hanson, Andrew J. I-232
Harding, Chris II-417
Haroz, Steve II-215
He, Qiang I-19
He, Xiangjian II-176
He, Zhiyu I-191
Herman, Gabor T. I-263
Hernández, Jorge I-822
Hernandez-Marin, Sergio II-405
Hickey, Timothy J. II-617
Hintz, Tom II-176
Hoff, William II-534
Holtzman-Gazit, Michal I-648
Hong, Hyunki I-881
Hoseinnezhad, Reza I-445
Hossain, Md. Altab I-841
Hu, Jinhui I-9
Huang, Chung-Hsien II-31
Hubbold, Roger I-434, I-558

Hur, Namho II-176
Hwang, In Yong II-294
Hwang, Jane I-692
Hwang, Won-Il II-576
Hwang, Yongho I-881
Hyland, Bryce I-70

Imiya, Atsushi I-608

Jamet, D. II-480
Jang, Howook II-326
Jang, Jungshik II-146
Jeong, Hong I-334
Jeong, Karpjoo II-9
Jia, Jingping I-80, I-579
Jia, Wenjing II-176
Jiang, Fang II-447
Jiang, Jianmin II-227
Jin, Ge I-761, I-901
Jin, Huan II-647
Johnson, Philip I-293
Jung, Ho Gi II-294
Jung, Moon-Ryul I-831

Kamberov, George II-802
Kamberova, Gerda II-802
Kampel, Martin II-607
Karnowski, Thomas P. II-824
Kett, B. I-141
Khiat, Abdelaziz II-88
Kidode, Masatsugu II-88
Kim, Beomjin I-293
Kim, Daewhan II-196
Kim, Daijin I-323
Kim, Eun Yi II-9
Kim, Gerard J. I-692
Kim, Haedong II-326
Kim, J.C. I-133
Kim, Jaemin II-196
Kim, Jaihie II-294
Kim, Jane Hwang Namgyu I-692
Kim, Jee-In II-9, II-383
Kim, Jinwoong II-176
Kim, Jin Young I-731
Kim, Joohee II-1
Kim, Namgyu I-692
Kim, Soo-jeong II-9
Kim, Sung Bae II-731
Kim, Taekyung I-871
Kim, Taeyoung II-146
Kim, Y.S. I-133
Kirkup, Don II-427
Kjellin, Andreas I-180
Kolb, Andreas II-524
Kong, Hui I-283, I-628
Kong, T. Yung I-263
Korah, Thommen II-718
Koračin, Darko II-544
Kostková, J. II-802
Krüger, V. I-538
Kuester, Falko I-191
Kundu, M. I-353
Kuno, Yoshinori I-841
Kwan, Chiman I-112

Lachaud, Jacques-Olivier II-306, II-688
Lahiri, S. I-353
Lam, Tony II-770
Langton, John T. II-617
Lavest, Jean Marc II-891
Lee, Byoungmoo II-39
Lee, Byung-Uk II-70
Lee, Hee-Won II-70
Lee, Hyung-Soo I-323
Lee, Inho II-326
Lee, Jaeman I-881
Lee, Jiann-Der II-31
Lee, Joonwhoan I-133
Lee, Jun II-383
Lee, Ki-Youn I-313
Lee, Namkyung II-49
Lee, Sang-Joon I-761
Lee, Yun Hee II-294
Lefort-Piat, Nadine I-411
Lemuz-López, Rafael II-502, II-564
Lesage, David II-393
Lévy, Bruno I-383
Lewis, Daniel II-215
Li, Baoxin I-112, I-421
Li, Helen K. I-421
Li, Sheng I-161
Li, Si-kun II-882
Li, Wenjing II-627
Li, Xiaokun I-112
Lietsch, Stefan I-343
Lin, Yi I-151
Lind, Mats I-180
Lindner, Marvin II-524

Lindsay, Clifford I-505
Liu, Jing-Sin I-170
Liu, Jun I-434, I-558
Loss, Leandro II-870
Louhichi, Hanene II-364, II-891
Loviscach, Jörn II-437

Ma, Kwan-Liu II-215
Ma, Yunqian I-1, I-783
Mahlknecht, Stefan I-202
Malik, Aamir Saeed II-166
Mandal, Mrinal I-222
Mannan, S.M. II-166
Mansur, Al I-841
Martinec, D. II-802
Masoud, O. I-1
Matas, J. II-802
Mayol-Cuevas, Walterio II-276, II-354
Mayo, Simon II-427
McDonald, John II-334
Melo, Rubens II-657
Menudet, J.F. II-364
Mian, Ajmal I-90, I-860
Min, Dongbo II-316
Min, Kyeong-Yuk II-554
Min, Kyungha I-831, I-850, II-79
Mittal, Rajat I-761
Moeslund, T.B. I-60
Monekosso, N. II-861
Monger, Patricia I-102
Moon, Naureen II-678
Morris, Tim II-286

Na, Seung You I-731
Nasipuri, M. I-353
Neumann, Heiko I-741
Neumann, Ulrich I-9
Newcomb, Matt II-417
Nicolescu, Mircea I-40, II-870
Nilufar, Sharmin I-253
Nisar, Humaira II-166
Nixon, Mark S. I-141, I-243, II-156
Null, Bradley D. II-668

Obdržálek, Š. II-802
Ogasawara, Tsukasa II-88
Ohnishi, N. II-136
O'Toole, Alice J. II-447
Owen, G. Scott I-891, II-741
Owens, Robyn I-860

Paek, Inho I-599
Paik, Joonki I-599, I-871
Pajdla, T. II-802
Papanikolopoulos, N. I-1
Paragios, Nikos I-793, I-803
Park, Changhan I-599
Park, E.J. I-133
Park, Rae-Hong I-313, I-589, II-731
Park, S. I-60
Park, Sungchan I-334
Park, Sungjun II-383
Parvin, Bahram I-518, II-586
Paul, Jean-Claude I-672, I-682
Peltier, Samuel II-235
Pettersson, Lars Winkler I-180
Phillips, P. Jonathon II-447
Pinheiro, Sergio II-657
Poirson, Allen II-627
Price, Jeffery R. II-824
Prieto, Flavio I-822
Prinz, Astrid A. II-617
Puech, O. I-751
Pupilli, Mark II-276
Pyen, Seung-Man II-554

Qahwaji, Rami II-227
Qian, Gang II-647
Qu, Huamin I-273, I-528
Quigley, Aaron II-457

Rajko, Stjepan II-647
Rasmussen, Christopher II-718
Reilly, Jane II-334
Remagnino, Paolo II-427 , II-861
Renaud, Christophe II-265
Ribeiro, Eraldo II-750
Rogers, Timothy J. II-637
Rosenhahn, Bodo II-59
Rosten, Edward I-373
Rousselle, François II-265
Ruszala, Simon II-814
Ruwwe, Christian I-702
Ryu, Dong-Sung II-576
Ryu, Kwan Woo II-49

Sablatnig, Robert I-202
Sakai, Tomoya I-608

Salgian, Andrea II-709
Šára, R. II-802
Sarioz, Deniz I-263
Sarkis, Michel II-596
Sato, Kosuke I-608
Sbaiz, Luciano II-245
Schaefer, Gerald II-814
Schultz, Thomas I-455
Seidel, Hans-Peter I-485, II-59
Seipel, Stefan I-180
Severson, William II-534
Shaik, J. I-813
Shen, Cheng-dong II-882
Sing, J.K. I-353
Singh, Meghna I-222
Singh, Rahul II-678, II-770
Sinoquet, H. I-751
Sinzinger, Eric D. II-668
Skoglund, Johan II-374
Skourikhine, Alexei II-870
Smith, C. I-751
Sobczyk, Dominique I-393
Sohn, Kwanghoon II-316
Sun, Jia-Guang I-569, I-672, I-682
Suo, Xiaoyuan II-741
Süsstrunk, Sabine II-245
Suter, David I-445
Sutherland, Alistair I-495
Szumilas, Lech II-255

Tavakkoli, Alireza I-40
Teoh, Eam Khwang I-283, I-628, I-638
Thakur, S. I-353
Theobalt, Christian I-485
Tiesel, Jan-Phillip II-437
Tobin, Kenneth W. II-824
Torre, Vincent I-712
Tougne, Laure II-205
Tousek, Michael I-401
Toutant, J.-L. I-618
Trivedi, M.M. I-60
Twombly, Xander I-122, II-851

Umlauf, G. II-186
Usabiaga, Jorge I-122

Vacavant, Antoine II-205
Vanteru, B. I-813
Velastin, S.A. II-861
Vellore, Ramesh II-544

Wall, Jonathan II-824
Wallace, Andrew M. II-405
Walsh, Raymond I-761
Wang, Guoping I-161
Wang, Hengliang II-835
Wang, Qing I-80
Wang, Tao I-212
Wang, Wen-Ke I-569
Weickert, Joachim I-455
Wetzel, Melanie II-544
Wildenauer, Horst II-607
Wilkin, Paul II-427
Winterbottom, Cara I-465
Wittenberg, Veit I-343
Wu, Qiang II-176
Wu, Shwu-Jiuan II-31
Wu, Tony II-31
Wu, Yingcai I-273, I-528

Xu, L.-Q. II-861
Xu, Pengfei I-283, I-628
Xu, Xinyu I-421

Yang, Chuan-Kai II-760, II-780
Yang, Hui-Lin II-760
Yang, Qing I-518
Yang, Xingqiang II-344
Yang, Yi-Jun I-672, I-682
Yao, Yi I-401
Yavneh, Irad I-648
Yeasin, M. I-813
Yong, Jun-Hai I-569, I-672, I-682
Yoo, Jang-Hee II-156
Yoon, Pal Joo II-294
Yoon, SangUn II-316
You, Suya I-9
Yous, Sofiane II-88
Yu, Qian I-783
Yuan, Chunrong I-721
Yuan, Ding I-303
Yuan, Xiaoru I-50
Yun, Jae-Ho I-589

Zabel, Henning I-343
Zeng, Wei I-682
Zhan, B. II-861

Zhang, Caiming II-98, II-106, II-344
Zhang, Hui I-232, I-569, I-672, I-682
Zhang, Xin II-490
Zhao, Rongchun I-80, I-579
Zhao, Zheen I-638
Zhong, Li II-106
Zhou, Hong I-273, I-528
Zhu, Ying I-891, II-741
Zölzer, Udo I-702