Lecture Notes in Computer Science 3752

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

Editorial Board

Nikos Paragios Olivier Faugeras
Tony Chan Christoph Schnörr (Eds.)

Variational, Geometric, and Level Set Methods in Computer Vision

Third International Workshop, VLSM 2005
Beijing, China, October 16, 2005
Proceedings

Springer

Volume Editors

Nikos Paragios
C.E.R.T.I.S.
Ecole Nationale des Ponts et Chaussées
Champs sur Marne, France
E-mail: nikos.paragios@cermics.enpc.fr

Olivier Faugeras
I.N.R.I.A
2004 route des lucioles, 06902 Sophia-Antipolis, France
E-mail: Olivier.Faugeras@sophia.inria.fr

Tony Chan
University of California at Los Angeles
Department of Mathematics
Los Angeles, USA
E-mail: chan@math.ucla.edu

Christoph Schnörr
University of Mannheim
Department of Mathematics and Computer Science, Germany
E-mail: schnoerr@uni-mannheim.de

Library of Congress Control Number: Applied for

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

ISSN 0302-9743
ISBN-10 3-540-29348-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29348-4 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11567646 06/3142 5 4 3 2 1 0

Preface

Mathematical methods has been a dominant research path in computational vision leading to a number of areas like filtering, segmentation, motion analysis and stereo reconstruction. Within such a branch visual perception tasks can either be addressed through the introduction of application-driven geometric flows or through the minimization of problem-driven cost functions where their lowest potential corresponds to image understanding.

The 3rd IEEE Workshop on Variational, Geometric and Level Set Methods focused on these novel mathematical techniques and their applications to computer vision problems. To this end, from a substantial number of submissions, 30 high-quality papers were selected after a fully blind review process covering a large spectrum of computer-aided visual understanding of the environment.

The papers are organized into four thematic areas: (i) Image Filtering and Reconstruction, (ii) Segmentation and Grouping, (iii) Registration and Motion Analysis and (iiii) 3D and Reconstruction. In the first area solutions to image enhancement, inpainting and compression are presented, while more advanced applications like model-free and model-based segmentation are presented in the segmentation area. Registration of curves and images as well as multi-frame segmentation and tracking are part of the motion understanding track, while introducing computational processes in manifolds, shape from shading, calibration and stereo reconstruction are part of the 3D track.

We hope that the material presented in the proceedings exceeds your expectations and will influence your research directions in the future. We would like to acknowledge the support of the Imaging and Visualization Department of Siemens Corporate Research for sponsoring the Best Student Paper Award.

Nikos Paragios
Olivier Faugeras
Tony Chan
Christoph Schnoerr

Organizing Committee

Organizing Board

Olivier Faugeras	General Chair, INRIA, France
Nikos Paragios	Organizer & General Co-chair, École Nationale des Ponts et Chaussées, France
Tony Chan	Program Chair, University of California, Los Angeles, USA
Christoph Schnoerr	Program Chair, University of Mannheim, Germany

Programme Committee

Luis Alvarez	University of Las Palmas, Spain
Benedicte Bascle	France Télécom R&D, France
Andrew Blake	Microsoft Research, UK
Alfred Bruckstein	Technion University, Israel
Vincent Caselles	University of Pompeu Fabra, Spain
Yunmei Chen	University of Florida, USA
Laurent Cohen	University of Paris – Dauphine, France
Rachid Deriche	INRIA, France
Eric Grimson	Massachusetts Institute of Technology, USA
Frederic Guichard	DxO, France
Marie-Pierre Jolly	Siemens Corporate Research, USA
Ioannis Kakadiaris	University of Houston, USA
Renaud Keriven	École Nationale des Ponts et Chaussées, France
Ron Kimmel	Technion University, Israel
Hamid Krim	University of North Carolina, USA
Petros Maragos	National Technical University of Athens, Greece
James W. MacLean	University of Toronto, Canada
Dimitris Metaxas	Rutgers University, USA
Mads Nielsen	University of Copenhagen, Denmark
Mila Nikolova	Ecole Normale Supériore de Cachan, France
Stanley Osher	University of California, Los Angeles, USA
Emmanuel Prados	University of California, Los Angeles, USA
Dimitris Samaras	State University of New York, Stony Brook, USA
Guillermo Sapiro	University of Minnesota, USA
Nir Sochen	University of Tel Aviv, Israel

Stefano Soatto	University of California, Los Angeles, USA
George Tziritas	University of Crete, Greece
Allen Tannenbaum	Georgia Institute of Technology, USA
Jean-Philippe Thiran	EPFL, Switzerland
Anthony Yezzi	Georgia Institute of Technology, USA
Baba Vemuri	University of Florida, USA
Joachim Weickert	University of the Saarland, Germany
James Williams	Siemens Corporate Research, USA
Luminita Vese	University of California, Los Angeles, USA
Hongkai Zhao	University of California, Irvine, USA
Steven Zucker	Yale University, USA

Table of Contents

Image Filtering and Reconstruction

Segmentation and Grouping

Registration and Motion Analysis

3D and Reconstruction

A Study of Non-smooth Convex Flow Decomposition

Jing Yuan, Christoph Schnörr, Gabriele Steidl, and Florian Becker

Department of Mathematics and Computer Science,
University of Mannheim, 68131 Mannheim, Germany
www.cvgpr.uni-mannheim.de
kiwi.math.uni-mannheim.de

Abstract. We present a mathematical and computational feasibility study of the variational convex decomposition of 2D vector fields into coherent structures and additively superposed flow textures. Such decompositions are of interest for the analysis of image sequences in experimental fluid dynamics and for highly non-rigid image flows in computer vision.

Our work extends current research on image decomposition into structural and textural parts in a twofold way. Firstly, based on Gauss' integral theorem, we decompose flows into three components related to the flow's divergence, curl, and the boundary flow. To this end, we use proper operator discretizations that yield exact analogs of the basic continuous relations of vector analysis. Secondly, we decompose simultaneously both the divergence and the curl component into respective structural and textural parts. We show that the variational problem to achieve this decomposition together with necessary compatibility constraints can be reliably solved using a single convex second-order conic program.

1 Introduction

The representation, estimation, and analysis of non-rigid motions is relevant to many scenarios in computer vision, medical imaging, remote sensing, and experimental fluid dynamics. In the latter case, for example, sophisticated measurement techniques including pulsed laser light sheets, modern CCD cameras and dedicated hardware, enable the recording of high-resolution image sequences that reveal the evolution of spatial structures of unsteady flows [1].

In this context, two issues are particularly important. Firstly, the design and investigation of variational approaches to motion estimation that are well-posed through regularization but do not penalize relevant flow structures are of interest. A corresponding line of research concerns the use of higher-order regularizers as investigated, for example, in [2,3,4]. Secondly, representation of motions by components that capture different physical aspects are important for most areas of application mentioned above. Referring again to experimental fluid dynamics, for example, the extraction of coherent flow structures which are immersed into additional motion components at different spatial scales [5], poses a challenge for image sequence analysis.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 1–12, 2005.

The decomposition of images has become an interesting and active area of research quite recently. Based on the seminal paper [6] introducing total variation based image denoising, and on the use of norms that are suited for representing oscillating patterns [7], a range of novel variational and computational approaches have been suggested for decomposing images of general scenes into basic components related to geometry, texture, and noise; e.g., [8,9,10,11].

In the present paper, we focus on function decomposition from the viewpoint of non-rigid variational motion analysis, and based on our recent work [12]. Specifically, we consider Meyer's [7] variational model

$$\min \mathrm{TV}(f^s)\,, \quad \text{s.t.} \quad f^s + f^t = f\,,\ \left\|f^t\right\|_G \leq \delta \tag{1}$$

as a representative approach to the decomposition of a function f into its basic structural and textural parts f^s, f^t, and study the feasibility of an extension to the decomposition of motion vector fields. Our objective is the *simultaneous* decomposition of a vector field into physically relevant components related to its divergence and curl, *and* the decomposition of these components into parts with intrinsic variations at different scales.

In section 2, we introduce the discrete representation of vector fields by its basic components related to divergence, curl, and boundary values. Based on an accurate discretization employing various finite-dimensional spaces and corresponding operators, a variational model for the simultaneous decomposition of these components is proposed in section 3. From the computational point of view, we prefer to reformulate our variational problem as a convex conic program in subsection 4 because all compatibility constraints defining our decomposition can be included at once. While conic programming has found widespread applications in all branches of computational science, it has only recently been suggested for the decomposition of scalar-valued image functions [13]. Numerical experiments demonstrate the feasibility of our approach in section 5.

2 Vector Field Representation

2.1 Flow Discretization

For discretizing the relevant differential operators we apply the *mimetic finite difference method* introduced by Hyman and Shashkov in [14]. This method preserves the integral identities satisfied by the continuous differential operators by appropriately defining their discrete analogues simultaneously with respect to two grids which we call primal and dual grid. Then we define

H_P: space of scalar fields on vertices,
H_V: space of scalar field on cells,
H_S: space of vector fields defined normal to sides,
H_E: space of vector fields defined tangential to sides,

and H_P^o, H_S^o, H_E^o as their restricted versions of inner scalar/vector fields, see Fig. 1. Likewise, we consider the restriced spaces H_P^o, H_S^o, H_E^o also as naturally embedded in H_P, H_S, H_E with zero boundaries. While H_P and H_V are equipped

with the usual Euclidian norm, the norms on H_S and H_E include boundary weights, see appendix. The discrete versions of the first order operators ∇, div and curl with respect to the primal and dual grid are given by

$$\mathbb{G} : H_P \to H_E,\ \mathbb{D}iv\ : H_S \to H_V,\ \mathbb{C}url\ : H_E \to H_V,$$
$$\overline{\mathbb{G}} : H_V \to H_S,\ \overline{\mathbb{D}iv}\ : H_E^o \to H_P^o,\ \overline{\mathbb{C}url}\ : H_S^o \to H_P^o.$$

Reshaping the scalar/vector fields columnwise into vectors of appropriate lengths, our first-order operators act on the corresponding vector spaces as the matrices specified in the appendix.

Finally, for discretizing $\mathrm{n} \cdot u|_{\partial\Omega}$, we introduce the boundary operator $\mathbb{B}_\mathrm{n} : H_S \to \partial H_S := H_S \backslash H_S^o$, which restricts the vector field to the vectors at the grid's boundary multiplied by the outer normal vectors. For the matrix form of the operator, we refer to the appendix.

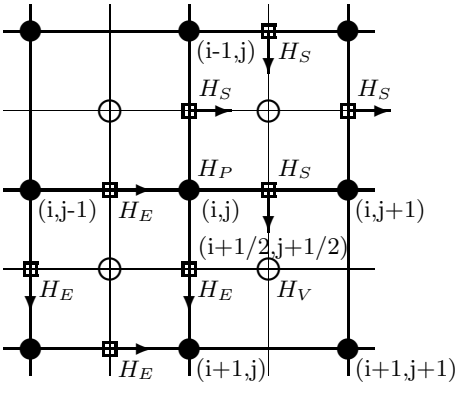


Fig. 1. Spaces H_P, H_V, H_S and H_E

2.2 Flow Representation

For the flow vectors $u \in H_S$, we see by definition of $\mathbb{D}iv$ and $\mathbb{B}_\mathrm{n}$ that

$$\mathbf{1}^\mathrm{T}_{\dim H_V}\, \mathbb{D}iv\, u = \mathbf{1}^\mathrm{T}_{\dim \partial H_S}\, \mathbb{B}_\mathrm{n} u, \tag{2}$$

where $\mathbf{1}_n$ denotes the vector consisting of n ones. This is just the discrete version of the *Gaussian Integral Theorem* $\int_\Omega \operatorname{div} u \,\mathrm{d}x = \int_{\partial\Omega} n \cdot u \,\mathrm{d}l$. Conversely, we say that $\rho \in H_V$ and $\nu \in \partial H_S$ fulfill the *compatibility condition* if

$$\mathbf{1}^\mathrm{T}_{\dim H_V}\, \rho = \mathbf{1}^\mathrm{T}_{\dim \partial H_S}\, \nu \tag{3}$$

Besides the flow representation $u \in H_S$, we will apply a second flow representation. To this end, consider the operator $A : H_S \to H_V \oplus H_P^o \oplus \partial H_S$ given in matrix form by

$$A := \begin{pmatrix} \mathbb{D}iv \\ \hline \mathbb{C}url \\ \mathbb{B}_\mathrm{n} \end{pmatrix} \in \mathbb{R}^{\dim H_S + 1, \dim H_S}, \tag{4}$$

where the $\overline{\mathbb{C}url}$ operator is naturally extended to the whole space H_S here. The operator A has full rank $\dim H_S$. Moreover, we see by (2) that $(\rho, \omega, \nu)^{\mathrm{T}}$ is in the image of A iff ρ and ν fulfill the compatibility condition (3). In this case u can be obtained from given $(\rho, \omega, \nu)^{\mathrm{T}}$ by $u = A^{\dagger}(\rho, \omega, \nu)^{\mathrm{T}}$, where $A^{\dagger} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ denotes the pseudoinverse of A.

Proposition 1. *There exists a one–to–one correspondence between the spaces H_S and*

$$V_S := \{(\rho, \omega, \nu)^{T} : \mathbf{1}^{T}_{\dim H_V}\, \rho = \mathbf{1}^{T}_{\dim \partial H_S}\, \nu\}\ ,$$

where $\rho = \mathbb{D}iv\ u$, $\omega = \overline{\mathbb{C}url}\ u$, $\nu = \mathbb{B}_{\mathrm{n}} u$, and conversely $u = A^{\dagger}(\rho, \omega, \nu)^{T}$.

3 Variational Approaches

3.1 Flow Decomposition

In this section, we want to decompose flow vectors $u \in H_S$, resp., $(\rho, \omega, \nu)^{\mathrm{T}} \in V_S$ in a meaningful way. To this end, let c_ρ denote the mean of the divergence of u and c_ω the mean of the curl of u, i.e.,

$$c_\rho := \mathbf{1}^{\mathrm{T}}_{\dim H_V}\, \rho \,/\, \dim H_V \;=\; \mathbf{1}^{\mathrm{T}}_{\dim H_V}\, \mathbb{D}iv\ u \,/\, \dim H_V\ , \tag{5}$$

$$c_\omega := \mathbf{1}^{\mathrm{T}}_{\dim H^o_P}\, \omega \,/\, \dim H^o_P \;=\; \mathbf{1}^{\mathrm{T}}_{\dim H^o_P}\, \mathbb{C}url\ u \,/\, \dim H^o_P\ . \tag{6}$$

These are the discrete versions of $|\Omega|^{-1} \int_\Omega \mathrm{div}(u) dx$ and $|\Omega|^{-1} \int_\Omega \mathrm{curl}(u) dx$. Then we can decompose the flow $(\rho, \omega, \nu)^{\mathrm{T}} \in V_S$ as

$$(\rho, \omega, \nu) = (c_\rho, c_\omega, \nu) + (\rho^o, \omega^o, 0), \tag{7}$$

where $\mathbf{1}^{\mathrm{T}}_{\dim H_V} \rho_o = \mathbf{1}^{\mathrm{T}}_{\dim H^o_P} \omega_o = 0$. Obviously, we have that $(c_\rho, c_\omega, \nu)^{\mathrm{T}}$, $(\rho^o, \omega^o, 0)^{\mathrm{T}} \in V_S$ again, so that $u = u^c + u^o$ is the corresponding decomposition of $u \in H_S$, where $u^c := A^{\dagger}(c_\rho, c_\omega, \nu)^{\mathrm{T}}$ and $u^o := A^{\dagger}(\rho^o, \omega^o, 0)^{\mathrm{T}}$. The vector u^c, resp. (c_ρ, c_ω, ν), represents the basic pattern of the non-rigid flow and its boundary behaviour while u^o, resp. $(\rho^o, \omega^o, 0)$, is related to the variant flow pattern. Now we want to further decompose the intrinsic flow variation u^o into a structural part u^s and a texture part u^t, i.e., $u^o = u^s + u^t$. By proposition 1, this corresponds to the decomposition

$$(\rho^o, \omega^o, 0) = (\rho^s, \omega^s, 0) + (\rho^t, \omega^t, 0).$$

In summary, our task consists in the decomposition of a given flow field $u \in H_S$ as

$$u = u^c + u^s + u^t. \tag{8}$$

We can apply A to u which provides us, by using in addition (5) and (6), with $(c_\rho, c_\omega, \nu)^{\mathrm{T}}$ and $(\rho^o, \omega^o, 0)^{\mathrm{T}}$. Then, inspired by Meyer's approach (1), we may compute $(\rho^s, \omega^s, 0)$ and $(\rho^t, \omega^t, 0)$ as solutions of the minimization problem

$$J(\rho^s, \omega^s, \rho^t, \omega^t) = \lambda_d \mathrm{TV}(\rho^s) + \lambda_c \mathrm{TV}(\omega^s), \tag{9}$$

$$\text{s.t.} \quad \rho^s + \rho^t = \rho^o, \quad \omega^s + \omega^t = \omega^o, \quad \left\|\rho^t\right\|_G \le \delta_d\,, \quad \left\|\omega^t\right\|_G \le \delta_c,$$

where the discrete TV functionals and the discrete versions of the G norm are defined in the appendix. This variational approach extends Meyer's model for the decomposition of scalar-valued functions to the *simultaneous* decomposition of vector fields into basic flow patterns. Finally, we may formally obtain u^s and u^t by solving the linear systems $(A^{\mathrm{T}}A)u^s = A^{\mathrm{T}}(\rho^s, \omega^s, 0)^{\mathrm{T}}$ and $(A^{\mathrm{T}}A)u^t = A^{\mathrm{T}}(\rho^t, \omega^t, 0)^{\mathrm{T}}$. However, these systems are very ill-conditioned so that we prefer to compute the components of u directly by minimizing the corresponding functional

$$\begin{aligned} & J(u^c, u^s, u^t) = \lambda_d \mathrm{TV}(\mathbb{D}iv\ u^s) + \lambda_c \mathrm{TV}(\overline{\mathbb{C}url}\, u^s) \qquad (10) \\ \text{s.t.} \quad & u^c + u^s + u^t = u, \\ & \overline{\mathbb{G}}\mathbb{D}iv\ u^c = 0, \quad \mathbb{G}\overline{\mathbb{C}url}\, u^c = 0, \quad \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \overline{\mathbb{C}url}\, u^s = 0, \\ & \mathbb{D}iv\ u^t = \rho^t, \quad \overline{\mathbb{C}url}\, u^t = \omega^t, \quad \left\|\rho^t\right\|_G \le \delta_d\,, \quad \left\|\omega^t\right\|_G \le \delta_c. \end{aligned}$$

This approach also fits into our flow estimation model in the next section. We note that the third constraint is related to the decomposition (7). While $\mathbf{1}^{\mathrm{T}}_{\dim H_V} \mathbb{D}iv\ u^o = 0$ is automatically fulfilled by the compatibility condition, we have to take care about $\mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \overline{\mathbb{C}url}\, u^o = 0$. However, by the G norm constraint we have $\overline{\mathbb{C}url}\, u^t = \mathbb{D}iv\ p$ for some p which again, by the compatibility condition, and since $\overline{\mathbb{C}url}$ maps to H_P^o, implies that $\mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \overline{\mathbb{C}url}\, u^t = 0$. As a result, we have only to take u^s into account.

Finally, we point out that as in the scalar-valued case, some variations of the approach (10) are easily conceivable. Referring to [8,10], for instance, the constraint $u^c + u^s + u^t = u$ in (10) could be replaced by a L^2 penalty term. This would imply L^2 penalty terms for each component in the decomposition.

3.2 Optical Flow Estimation Through Flow Decomposition

In this section, we combine the usual optical flow estimation method with the structure-texture flow decomposition (8). For a given image sequence $\{g\} \in H_V$, we want to compute the components u^c with constant divergence and curl, the large-scale patterns u^s of divergence and curl with bounded BV-norms, and the small-scale patterns u^t of divergence and curl with bounded G-norms, by solving

$$\begin{aligned} J(u^{c,s,t}) = & \left\|\overline{\mathbb{G}}g \cdot (u^c + u^s + u^t) + g_t\right\|_2^2 + \lambda_d \mathrm{TV}(\mathbb{D}iv\ u^s) + \lambda_c \mathrm{TV}(\overline{\mathbb{C}url}\, u^s) \quad (11) \\ \text{s.t.} \quad & \overline{\mathbb{G}}\mathbb{D}iv\ u^c = 0, \quad \mathbb{G}\overline{\mathbb{C}url}\, u^c = 0, \quad \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \overline{\mathbb{C}url}\, u^s = 0, \\ & \mathbb{D}iv\ u^t = \rho^t, \quad \overline{\mathbb{C}url}\, u^t = \omega^t, \quad \left\|\rho^t\right\|_G \le \delta_d\,, \quad \left\|\omega^t\right\|_G \le \delta_c. \end{aligned}$$

Here g_t denotes the discretization of the time derivative by a forward difference and the inner product is taken with respect to H_S. We refer to (11) as TV–G model. However, for the image areas where $\nabla g = 0$, the data term disappears such that the local constraints through the two G-norm terms lead to unbounded solutions. Hence, the flow estimation by solving problem (11) is not well-posed. Therefore, we propose to replace the TV–G model by a TV–L_2 model where the texture flow patterns u^t have divergence and curl with bounded L_2-norms:

$$J(u^{c,s,t}) = \left\|\overline{\mathbb{G}}g \cdot (u^c + u^s + u^t) + g_t\right\|_2^2 + \lambda_d \mathrm{TV}(\mathbb{D}iv\ u^s) + \lambda_c \mathrm{TV}(\overline{\mathbb{C}url}\ u^s) \quad (12)$$
$$+ \gamma_d \left\|\mathbb{D}iv\ u^t\right\|_2^2 + \gamma_c \left\|\overline{\mathbb{C}url}\ u^t\right\|_2^2$$
$$\text{s.t.} \quad \overline{\mathbb{G}}\mathbb{D}iv\ u^c = 0, \quad \mathbb{G}\overline{\mathbb{C}url}\ u^c = 0, \quad \mathbf{1}^{\mathrm{T}}_{\dim H_P^o}(\overline{\mathbb{C}url}\ u^s + \overline{\mathbb{C}url}\ u^t) = 0.$$

Our experiments show that this approach works well although the superiosity of the G–norm over the L_2–norm in capturing (scalar) oscillating patterns was experimentally shown in [15].

3.3 Incompressible Optical Flow Estimation

Incompressible flows which are divergence-free are common in computational fluid dynamics and 2D turbulence. According to the Helmholtz decomposition, a 2D vector field can be decomposed into an irrotational part and a soleniodal part which is divergence–free. The discete counterpart of the Helmholtz decomposition with respect to our mimetic finite difference discretization has been introduced in [12]. Specifically, we obtain that a divergence-free vector $u \in H_S$ can be written as $u = \mathbb{G}^{\perp}\psi$ for some $\psi \in H_P$, where the operator $\mathbb{G}^{\perp} : H_P \to H_S$ is defined in the appendix. By definition of $\mathbb{G}^{\perp}$, it is easy to check that $\mathbb{D}iv\ \mathbb{G}^{\perp} = \mathbf{0}$, and that the restricted operator $\mathbb{G}^{\perp}|_{H_P^o}$ maps to H_S^o. Now we want to estimate the components u^c, u^s and u^t of a divergence–free flow $u = \mathbb{G}^{\perp}\psi$, i.e.,

$$u = u^c + u^s + u^t = \mathbb{G}^{\perp}\psi^c + \mathbb{G}^{\perp}\psi^s + \mathbb{G}^{\perp}\psi^t, \quad (13)$$

where, by regarding the boundary conditions, $\psi^c \in H_P$ and $\psi^s, \psi^t \in H_P^o$. Let $\triangle_c := \overline{\mathbb{C}url}\,\mathbb{G}^{\perp}|_{H_P^o} : H_P^o \to H_P^o$ and $\triangle := \overline{\mathbb{C}url}\,\mathbb{R}^{H_S}_{H_S^o}\,\mathbb{G}^{\perp} : H_P \to H_P^o$, where $R^{H_S}_{H_S^o}$ denotes the restriction of H_S to H_S^o by boundary cutting. Then we can rewrite our TV–G approach (11) with respect to (13) as

$$J(\psi^{c,s,t}) = \left\|\overline{\mathbb{G}}g \cdot \mathbb{G}^{\perp}(\psi^c + \psi^s + \psi^t) + g_t\right\|_2^2 + \lambda_c \mathrm{TV}(\triangle_c \psi^s) \quad (14)$$
$$\text{s.t.} \quad \mathbb{G}\triangle\psi^c = 0,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P}\psi^c = 0, \quad \mathbf{1}^{\mathrm{T}}_{\dim H_P^o}\triangle_c\psi^s = 0, \quad \triangle_c\,\psi^t = \omega^t,\ \left\|\omega^t\right\|_G \le \delta_c,$$

and our TV–L_2 approach (12) as

$$J(\psi^{c,s,t}) = \left\|\overline{\mathbb{G}}g \cdot \mathbb{G}^{\perp}(\psi^c + \psi^s + \psi^t) + g_t\right\|_2^2 + \lambda_c \mathrm{TV}(\triangle_c\psi^s) + \gamma_c \left\|\triangle_c\psi^t\right\|_2^2 \quad (15)$$
$$\text{s.t.} \quad \mathbb{G}\triangle\psi^c = 0, \quad \mathbf{1}^{\mathrm{T}}_{\dim H_P}\psi^c = 0, \quad \mathbf{1}^{\mathrm{T}}_{\dim H_P^o}(\triangle_c\psi^s + \triangle_c\psi^t) = 0.$$

We will see that in areas where $\|\nabla g\| \ll 1$, the solution to (14) becomes sensitive to small pertubations while (15) gives reasonable results.

4 Optimization

Our computational approach to solving (10) is based on second-order cone programming (SOCP) [16]. This amounts to minimizing a linear objective function

subject to the constraints that several affine functions of the variables have to lie in a *second order cone* $\mathcal{L}^{n+1} \subset \mathbb{R}^{n+1}$ defined as the convex set

$$\mathcal{L}^{n+1} = \left\{ (x;\ t) = (x_1, \ldots, x_n, t)^\top \;\middle|\; \|x\|_2 \le t \right\}. \tag{16}$$

With this notation, the general form of a SOCP is given by

$$\inf_{x \in \mathbb{R}^n} \ f^\top x\,,\ \text{s.t. } \left(A_i x + b_i;\ c_i^T x + d_i\right) \in \mathcal{L}^{n+1}\,,\ i = 1, \ldots, m. \tag{17}$$

Problem (17) is a convex program for which efficient, large scale solvers are available [17]. In this paper, we used the SeDuMi-package [18]. In connection with TV–based image decomposition the application of SOCPs, was recently suggested in [13].

Using the notation given in the appendix, we reformulate the variational approach (10) as a SOCP:

$$J(u^c, u^s, u^t) = \lambda_d \mathbf{1}^{\mathrm{T}}_{\dim H_V} \boldsymbol{v} + \lambda_c \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \boldsymbol{w} \tag{18}$$

$$\text{s.t.} \quad u^c + u^s + u^t = u\,,\ \overline{\mathbb{G}}\mathbb{D}iv\ u^c = 0\,,\ \mathbb{G}\overline{\mathbb{C}url}\ u^c = 0\,,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \overline{\mathbb{C}url}\ u^s = 0\,,$$

$$\mathbb{D}iv\ u^t = \mathbb{D}iv\ p_d\,,\ \overline{\mathbb{C}url}\ u^t = \mathbb{D}iv\ p_c\,,\ \left((\overline{\mathbb{G}}\mathbb{D}iv\ u^s)_{V_{i,j}};\ v_{V_{i,j}}\right) \in \mathcal{L}^5\,,$$

$$\left((\overline{\mathbb{G}\mathbb{C}url}\ u^s)_{P^o_{i,j}};\ w_{P^o_{i,j}}\right) \in \mathcal{L}^5\,,\ \left((p_d)_{V_{i,j}};\ \delta_d\right) \in \mathcal{L}^5\,,\ \left((p_c)_{P^o_{i,j}};\ \delta_c\right) \in \mathcal{L}^5$$

In order to incorporate the quadratic terms of the variational approaches to optical flow estimation (11), (12), (14), and (15), we use the following rotated version of the standard cone:

$$\mathcal{R}^{n+2} := \left\{ \left(\boldsymbol{x}, x_{n+1}, x_{n+2}\right)^\top \in \mathbb{R}^{n+2}\,,\ x_{n+1} x_{n+2} \ge \frac{1}{2} \|\boldsymbol{x}\|^2\ ,\ x_{n+1} + x_{n+2} \ge 0 \right\}$$

Fixing $x_{n+2} = 1/2$, we have $x_{n+1} \ge \|\boldsymbol{x}\|^2$. Below, we confine ourselves to rewriting (14), and (15) as SOCPs. The SOCPs corresponding to (11), (12) look very similar.

The incompressible flow estimation approach (14), rewritten as a SOCP, reads

$$J(\psi^{c,s,t}) = v + \lambda_c \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \boldsymbol{w} \tag{19}$$

$$\text{s.t.} \quad \mathbb{G}\triangle\psi^c = 0\,,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P} \psi^c = 0\,,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \triangle_c \psi^s = 0\,,\ \triangle_c \psi^t = \mathbb{D}iv\ p_c$$

$$\left((\overline{G}\triangle_c \psi^s)_{V_{i,j}};\ w_{V_{i,j}}\right) \in \mathcal{L}^5\,,\ \left((p_c)_{P^o_{i,j}};\ \delta_c\right) \in \mathcal{L}^5\,,$$

$$\left(\overline{\mathbb{G}} g \cdot \mathbb{G}^\perp(\psi^c + \psi^s + \psi^t) + g_t;\ v,\ 1/2\right) \in \mathcal{R}^{\dim H_V + 2}$$

Approach (15), on the other hand, becomes

$$J(\psi^{c,s,t}) = v + \gamma_d t + \lambda_c \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \boldsymbol{w} \tag{20}$$

$$\text{s.t.} \quad \mathbb{G}\triangle\psi^c = 0\,,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P} \psi^c = 0\,,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \triangle_c \psi^s = 0\,,\ \mathbf{1}^{\mathrm{T}}_{\dim H_P^o} \triangle_c \psi^t = 0$$

$$\left((\overline{G}\triangle_c \psi^s)_{V_{i,j}};\ w_{V_{i,j}}\right) \in \mathcal{L}^5\,,\ \left(\overline{\mathbb{G}} g \cdot \mathbb{G}^\perp(\psi^c + \psi^s + \psi^t) + g_t;\ v, 1/2\right) \in \mathcal{R}^{\dim H_V + 2}\,,$$

$$\left(\triangle_c \psi^t;\ t, 1/2\right) \in \mathcal{R}^{\dim H_P^o + 2}$$

5 Numerical Experiments

In this section, we show some experiments with flow decomposition and flow restoration.

Flow Decomposition. Figure 2 shows a turbulent flow field u as ground truth, along with its divergence ρ and curl ω. Figures (3) and (2) show the variational decomposition computed with the approach (10). Note that the structural and textural components recovered the interesting motion patterns at different scales, which are not easily visible in the flow u itself. The decomposed velocities are shown below in Fig. (4).

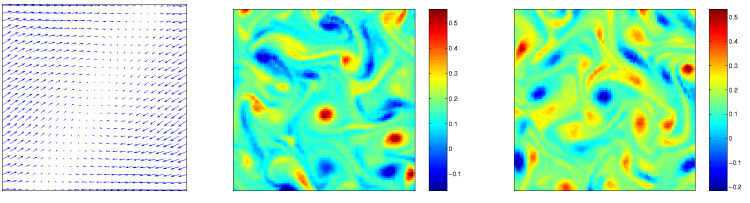


Fig. 2. Ground truth data to be decomposed: flow field u (left), its divergence field ρ (center), and its curl field ω (right)

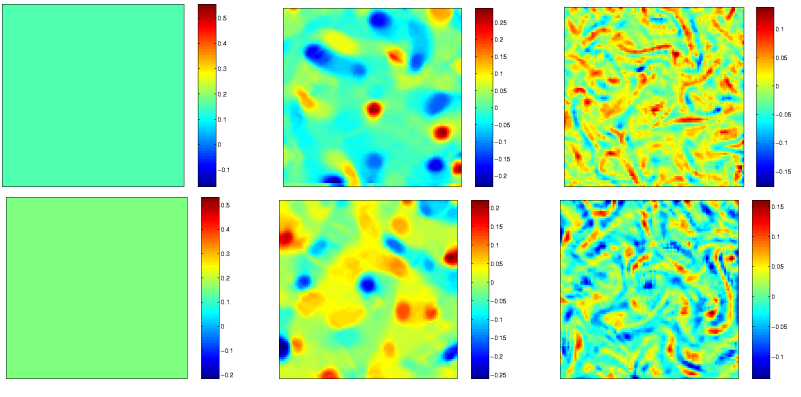


Fig. 3. Decomposition of u from Fig. 2 with the approach (10). From left to right. Top: ρ^c, ρ^s, ρ^t. Bottom: $\omega^c, \omega^s, \omega^t$. The structure and texture components reveal turbulent flow patterns at different scales which are not easily visible in the flow u itself.

Flow Estimation. We report result validating the flow estimation models (14) and (15). We first created a divergence-free ground truth flow field u by superimposing a dominant laminar flow (both divergence- and curl-free) with some turbulent vortices structures, see Fig. 5. Using this flow, an artificial image sequence was created for which $|\nabla g(x)| \neq 0$, $\forall x \in \Omega$.

Figures 6, 7 and 8 show the decomposition-based optical flow estimates. The u^c component nicely recovered the laminar flow, whereas the structural and tex-

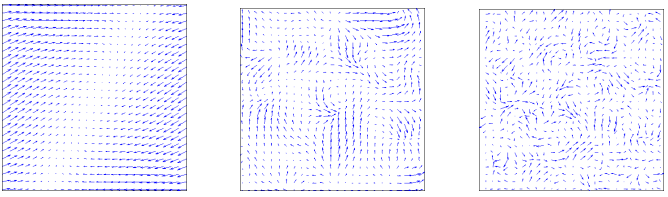


Fig. 4. The components of the flow u from Fig. 2: u^c (left), u^s (middle), and u^t (right). The vectors of u^s, u^t are scaled-up for better visibility. Note that despite $|u| \approx |u^c|$, structural and texture part u^s and u^t are recovered well.

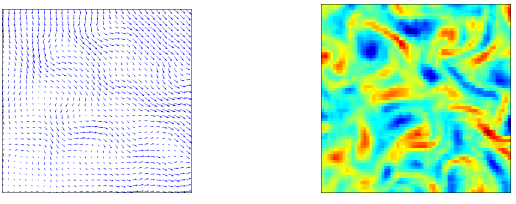


Fig. 5. Ground truth data u and its curl to be estimated from a corresponding artificially created image sequence. u is a superposition of a laminar flow (div- and curl-free) and turbulent vortices.

tural components reveal the turbulent curl field. Furthermore, the TV $- L_2$ regularizer turned out to be more robust than the TV $- G$ model in connection with the degenerate data term commonly used for variational optical flow estimation.

6 Conclusion and Further Work

Along the lines of current research on variational convex decomposition of image functions, we presented a range of variational models extended to the decomposition and estimation of vector fields which represent image motions. Using proper discretizations, these models achieve a twofold decomposition: three components of the flow field representing flow variations at different scales, along with a further decomposition of the divergence and the curl into a structural and a textural part, respectively. We also presented a variational model for the decomposition-based estimation of divergence-free flows which is of interest for experimental fluid dynamics. Numerical results conducted by convex second-order cone programming showed the feasibility of our approach as well as promising results with respect to the processing and analysis of complex flow patterns in real-world applications.

Our further work concerns the study of various TV $- *$ combinations of regularizers for flow field decomposition which in comparison to image decomposition may behave differently due to the data term and corresponding image pre-processing. Furthermore, we will investigate more robust models for using G-norm regularization in connection with the (mathematically) degenerate data term for optical flow estimation.

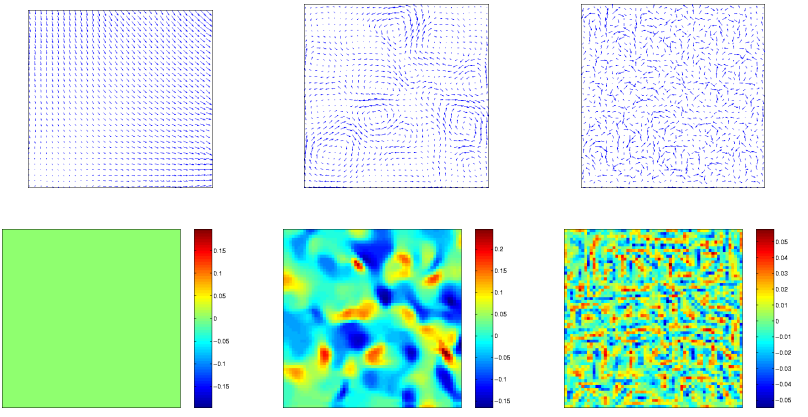


Fig. 6. Estimated and decomposed flow corresponding to Fig. 5, using the approach (14). From left to right. Top: u^c, u^s and u^t. Bottom: ω^c, ω^s and ω^t. Note, that the laminar component is almost completely represented by u^c, ω^c, whereas the turbulent patterns are captured by the remaining components at two different scales. The texture components u^t, ω^t reflect the lack of robustness of G-norm regularization in combination with the degenerate data term for optical flow estimation.

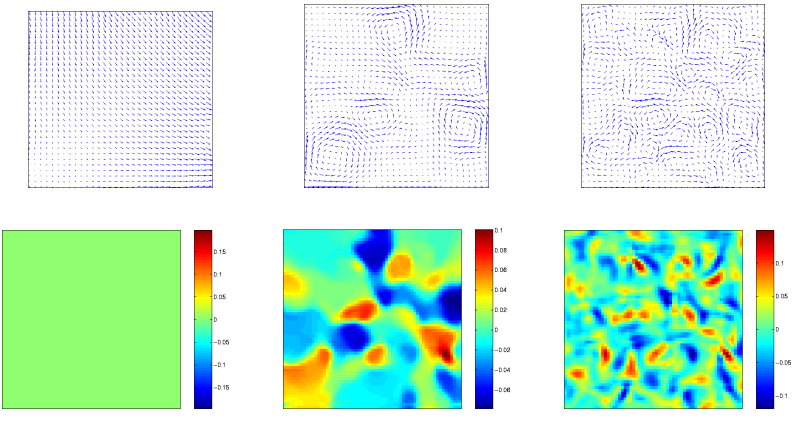


Fig. 7. Results analogous to Fig. 6, computed with $TV - L_2$ regularization (15), however. The sensitivity of the texture part (left column) has been removed.

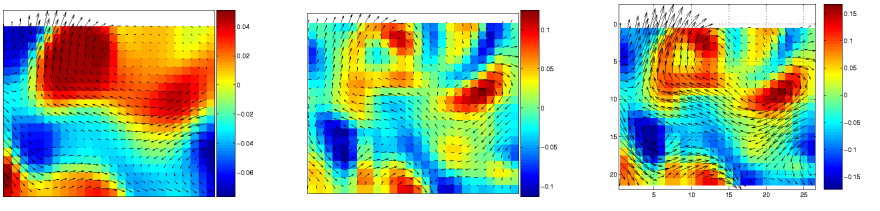


Fig. 8. Close-up view of a section of Fig. 7. From top to bottom: $\omega^s, \omega^t, \omega^s + \omega^t$ with the corresponding flows as overlays.

References

1. M. Raffel, C. Willert, and J. Kompenhans. *Particle Image Velocimetry*. Springer, 2nd edition, 2001.
2. D. Suter. Motion estimation and vector splines. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 939–942, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.
3. T. Corpetti, E. Mémin, and P. Pérez. Dense estimation of fluid flows. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(3):365–380, 2002.
4. T. Kohlberger, E. Mémin, and C. Schnörr. Variational dense motion estimation using the helmholtz decomposition. In L.D. Griffin and M. Lillholm, editors, *Scale Space Methods in Computer Vision*, volume 2695 of *LNCS*, pages 432–448. Springer, 2003.
5. J. Lewalle, J. Delville, and J.-P. Bonnet. Decomposition of mixing layer turbulence into coherent structures and background fluctuations. *Applied Scientific Research*, 64(4):301–328, 2000.
6. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
7. Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations*, volume 22 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2001. The fifteenth Dean Jacqueline B. Lewis memorial lectures.
8. L.A. Vese and S.J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Scient. Computing*, 19(1-3):553–572, 2003.
9. L.A. Vese and S.J. Osher. Image denoising and decomposition with total variation minimization and oscillatory functions. *J. of Math. Imag. Vision*, 20(1/2):7–18, 2004.
10. J.F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *J. of Math. Imag. Vision*, 22(1):71–88, 2005.
11. J.F. Aujol and A. Chambolle. Dual norms and image decomposition models. *Int. J. of Comp. Vision*, 63(1):85–104, 2005.
12. J. Yuan, P. Ruhnau, E. Mémin, and C. Schnörr. Discrete orthogonal decomposition and variational fluid flow estimation. In *Scale-Space 2005*, volume 3459 of *lncs*, pages 267–278. Springer, 2005.
13. W. Yin, D. Goldfarb, and S. Osher. Total variation based image cartoon-texture decomposition. Technical Report CORC TR-2005-01, Columbia University, 2005. submitted to *Inverse Problems*.
14. James M. Hyman and Mikhail J. Shashkov. The orthogonal decomposition theorems for mimetic finite difference methods. *SIAM J. Numer. Anal.*, 36(3):788–818 (electronic), 1999.
15. Jean-Franois Aujol and Antonin Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):85 – 104, 2005.
16. Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.
17. H.D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Math. Programming, Series B*, 95(2):407–430, 2003.
18. J. F. Sturm. Using SeDuMi 1.02, a MATLAB for optimization over symmetric cones. Technical Report, Communications Research Laboratory, McMaster University, 1280 Main Street West, Hamilton, Ontario L8S 4K1, Canada, August 1998.

Appendix

Let our primal grid consist of $m \times n$ vertices. Reshaping the scalar/vector fields columnwise into vectors, we can identify

$$H_P = \mathbb{R}^{mn},\ H_P^o = \mathbb{R}^{(m-2)(n-2)}, H_V = \mathbb{R}^{(n-1)(n-1)},$$
$$H_S = \mathbb{R}^{m(n-1)+n(m-1)}, H_S^o = \mathbb{R}^{(m-1)(n-2)+(n-1)(m-2)},$$

and finally H_E, H_E^o as H_S, H_S^o. While H_P and H_V are equipped with the usual Euclidian norm, the norm on H_S and H_E^o are given cell adapted as follows: for $u \in H_S$ and $i = 1, \dots, m-1; j = 1, \dots, n-1$, let

$$u_{V_{i,j}} := \frac{1}{\sqrt{2}} \left(u_{i,j+\frac{1}{2}}, u_{i+1,j+\frac{1}{2}}, u_{i+\frac{1}{2},j}, u_{i+\frac{1}{2},j+1}\right)^{\mathrm{T}}.$$

and

$$\|u\|_{H_S}^2 := \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \|u_{V_{i,j}}\|_2^2 = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \frac{1}{2}(u_{i,j+\frac{1}{2}}^2 + u_{i+1,j+\frac{1}{2}}^2 + u_{i+\frac{1}{2},j}^2 + u_{i+\frac{1}{2},j+1}^2).$$

Similarly, we introduce the norm on H_E^o with respect to $u_{P_{i,j}^o}$. Further we define the TV functional for $\rho \in H_V$ as $\mathrm{TV}(\rho) := |\overline{\mathbb{G}}\,\rho|_{H_S}$, where

$$|u|_{H_S} = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \|u_{V_{i,j}}\|_2 = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \sqrt{\frac{1}{2}(u_{i,j+\frac{1}{2}}^2 + u_{i+1,j+\frac{1}{2}}^2 + u_{i+\frac{1}{2},j}^2 + u_{i+\frac{1}{2},j+1}^2)}$$

and for $\omega \in H_P^o$ as $\mathrm{TV}(\omega) := |\mathbb{G}|_{H_P^o}\,\rho|_{H_E^o}$. Finally, the discrete G norms are given by

$$\|\rho\|_G := \inf_{\rho = \mathbb{D}iv\ p} \| \left(\|p_{V_{i,j}}\|_2\right)_{i,j} \|_\infty, \quad \|\omega\|_G := \inf_{\omega = \overline{\mathbb{D}iv}\ p} \| \left(\|p_{P_{i,j}^o}\|_2\right)_{i,j} \|_\infty.$$

Let

$$\boldsymbol{D}_m := \begin{pmatrix} -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{m-1,m}, \quad \tilde{\boldsymbol{D}}_m := \begin{pmatrix} 2 & & & & \\ -1 & \ddots & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ 0 & \dots & 0 & -1 & 1 \\ 0 & \dots & 0 & 0 & -2 \end{pmatrix} \in \mathbb{R}^{m+1,m},$$

Then the discrete first order operators can be identified with the following matrices:

$$\mathbb{G} = \begin{pmatrix} \boldsymbol{I}_n \otimes \boldsymbol{D}_m \\ \boldsymbol{D}_n \otimes \boldsymbol{I}_m \end{pmatrix}, \qquad \overline{\mathbb{G}} = \begin{pmatrix} \boldsymbol{I}_{n-1} \otimes \tilde{\boldsymbol{D}}_{m-1} \\ \tilde{\boldsymbol{D}}_{n-1} \otimes \boldsymbol{I}_{m-1} \end{pmatrix},$$

$$\mathbb{D}iv = \left(\boldsymbol{I}_{n-1} \otimes \boldsymbol{D}_m, \boldsymbol{D}_n \otimes \boldsymbol{I}_{m-1}\right), \quad \overline{\mathbb{D}iv} = \left(\boldsymbol{I}_{n-2} \otimes \boldsymbol{D}_{m-1}, \boldsymbol{D}_{n-1} \otimes \boldsymbol{I}_{m-2}\right),$$
$$\mathbb{C}url = \left(\boldsymbol{D}_n \otimes \boldsymbol{I}_{m-1}, -\boldsymbol{I}_{n-1} \otimes \boldsymbol{D}_m\right), \quad \overline{\mathbb{C}url} = \left(\boldsymbol{D}_{n-1} \otimes \boldsymbol{I}_{m-2}, -\boldsymbol{I}_{n-2} \otimes \boldsymbol{D}_{m-1}\right),$$

where $\otimes$ denotes the Kronecker product of matrices. The operator $\mathbb{G}^\perp : H_P \to H_S$ is defined by

$$\mathbb{G}^\perp = \begin{pmatrix} -\boldsymbol{D}_n \otimes \boldsymbol{I}_m \\ \boldsymbol{I}_n \otimes \boldsymbol{D}_m \end{pmatrix},$$

It is easy to check that the restricted operator $\mathbb{G}^\perp|_{H_P^o}$ maps to H_S^o. Finally, the boundary operators are given by

$$\mathbb{B}_{\mathrm{n}} = \begin{pmatrix} \boldsymbol{I}_{n-1} \otimes \boldsymbol{B}_m & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_n \otimes \boldsymbol{I}_{m-1} \end{pmatrix}, \ \boldsymbol{B}_m := \begin{pmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{2,m}.$$

where $\boldsymbol{0}$ are zero matrices of appropriate sizes.

Denoising Tensors via Lie Group Flows

Y. Gur and N. Sochen

Department of Applied Mathematics, Tel-Aviv university,
Ramat-Aviv, Tel-Aviv 69978, Israel
{yanivg, sochen}@post.tau.ac.il

Abstract. The need to regularize tensor fields arise recently in various applications. We treat in this paper tensors that belong to matrix Lie groups. We formulate the problem of these $SO(N)$ flows in terms of the principal chiral model (PCM) action. This action is defined over a Lie group manifold. By minimizing the PCM action with respect to the group element, we obtain the equations of motion for the group element (or the corresponding connection). Then, by writing the gradient descent equations we obtain the PDE for the Lie group flows. We use these flows to regularize in particular the group of N-dimensional orthogonal matrices with determinant one i.e. SO(N). This type of regularization preserves their properties (i.e., the orthogonality and the determinant). A special numerical scheme that preserves the Lie group structure is used. However, these flows regularize the tensor field isotropically and therefore discontinuities are not preserved. We modify the functional and thereby the gradient descent PDEs in order to obtain an anisotropic tensor field regularization. We demonstrate our formalism with various examples.

1 Introduction

For more than a decade PDE's are widely used to tackle many image processing problems such as image restoration, segmentation, image enhancing and much more. Especially interesting are the nonlinear PDE's which in the context of image restoration has been proved to have remarkable denoising, deblurring as well as edges preserving properties. We will mention some of these works such as the pioneering work by Perona and Malik [19] on image denoising, the work by Osher and Rudin [16] on image enhancement and many others which are discussed extensively in [1,31,11,23]. Some of the image processing problems may be formulated in terms of Lagrangian actions where the variation of the Lagrangian leads to the equations of motion. The gradient descent equations then defines the PDE's that we wish to apply to images in order to obtain the desired result [15,21] (i.e, segmentation, denoising, etc).

Earlier studies dealt with scalar valued images. It was later generalized to vector-valued images (see for example [35,2,24,22,29] and references therein). Works on constrained regularization of vector-valued image were treated in the literature as well in [20,28,3,26,12].

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 13–24, 2005.

In the last years new methods which consider tensor-valued images have emerged. In these new methods at each point of the two (or three) dimensional image space a tensor is attached rather than a scalar or a vector. This tensor field might be noisy and therefore one has to regularize it in order to extract its original texture. Moreover, the tensor field has certain properties that we wish to preserve along the flow (e.g., orthogonality, unit norm, etc) and it might lie on a non-flat manifold. In order to regularize these fields and preserve their original properties one has to adopt new methods, both analytical and numerical.

A solution to the problem of orthogonal tensor field regularization was proposed by Deriche et al. [30,5]. In their formalism, the orthogonality of the tensor field is preserved by adding a constraint term to the *unconstraint* gradient descent equation using Lagrange multipliers. The constraint term preserve the orthonormality of the vector basis along the flow. The unconstraint gradient descent equation was obtained by minimizing the unconstrained ϕ functional. Different methods for regularization of tensor fields were proposed recently by [32,4,18,17,13]. Their work is mainly in relation with the DT-MRI application.

In this work we suggest a novel and natural framework to the problem of tensor field regularization. We assume here that the tensor at each point is a Lie group element and construct a regularization flow that respects the group's structure. The constrained gradient descent equation will be derived directly from a Lagrangian without any additional constraint e.g. without a lagrange multiplier. The functional is defined directly on the Lie group manifold. In order to solve the PDE numerically such that the Lie group field evolves on the group manifold we use the Lie group integrating methods introduced in [8,10]. Our main example is the $SO(N)$ group which is of relevant in DT-MRI yet the formalism is of general applicability.

The plan of the paper is as follows: In section 2 we will give some mathematical preliminaries that will be used in this work. In section 3 we present the generalized Principal Chiral Model (PCM) action and derive the gradient descent equations. The gradient descent equations for this action will define the PDE flow on the group manifold. In section 4 we describe how to implement the flow to evolve on group manifold in general and on $SO(N)$ in particular. We will present and use modern Lie-group numerical integration methods. Results are given in section 5 where we demonstrate regularization of noisy three-dimensional orthogonal tensor field. Finally, concluding remarks are given in section 6.

2 A Bit About Lie Groups

For our discussion it is essential to introduce some of the basic definitions considering Lie groups and Lie algebra.

Definition 1. *A* Lie group *is a group $\mathcal{G}$ which is a differentiable manifold equipped with smooth product $\mathcal{G} \times \mathcal{G} \mapsto \mathcal{G}$.*

Definition 2. *The* Lie algebra $\mathfrak{g}$ *of Lie group G is defined as the linear vector space of all tangent vectors to G at the identity. This tangent space is denoted $T_I\mathcal{G}$.*

Definition 3. *A real matrix Lie group is a smooth subset* $\mathcal{G} \subseteq \mathbb{R}^{N\times N}$ *closed under matrix product and matrix inversion. The identity matrix is denoted* $I \in \mathcal{G}$.

Definition 4. *A Lie algebra of a matrix Lie group is a linear subspace* $\mathfrak{g} \subseteq \mathbb{R}^{N\times N}$ *equipped with the operation* $\mathfrak{g} \times \mathfrak{g} \mapsto \mathfrak{g}$ *which is the Lie bracket (the commutator)* $[A, B] = AB - BA$. *This operation is bilinear, skew-symmetric* ($[A, B] = -[B, A]$), *and satisfies the Jacobi identity*

$$[A, [B, C]] + [C, [A, B]] + [B, [C, A]] = 0. \tag{1}$$

Definition 5. *The elements which span the Lie algebra space are called the generators of the Lie group or the infinitesimal operators of the group. Let* t_a, t_b *and* t_c *be the generators of the Lie group, then their algebra is close under the commutator operation*

$$[t_a, t_b] = f_{ab}^c t_c, \tag{2}$$

where f_{ab}^c *are the structure constants of the group and are antisymmetric in their lower indices* $f_{ab}^c = -f_{ba}^c$.

We will demonstrate our study on the special orthogonal matrix Lie group, $SO(N)$. Its elements are $N \times N$ orthogonal matrices with determinant one. This group is a subgroup of $O(N)$ which is the orthogonal group and its elements are $N \times N$ orthogonal matrices. The Lie algebra of $SO(N)$ and $O(N)$ is denoted $\mathfrak{so}(n)$ and consists of $N \times N$ skew-symmetric matrices. $O(N)$ and $SO(N)$ are special cases of quadratic Lie group which takes the form

$$\mathcal{G} = \{X | X^T P X = P\}, \tag{3}$$

where P is a constant matrix (for $O(N)$ and $SO(N)$, P is identity matrix). The corresponding Lie algebra is given by $\mathfrak{g} = \{A | PA + A^T P = 0\}$.

In order to map elements of the Lie algebra into the Lie group one may use the following maps

Definition 6. *The exponential mapping* $expm : \mathfrak{g} \mapsto \mathcal{G}$ *is defined as*

$$expm(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}, \tag{4}$$

where $expm(0) = I$. *Note that for* A *which is sufficiently near* $o \in \mathfrak{g}$ *the exponential mapping has a smooth inverse given by the matrix logarithm* $logm : \mathcal{G} \mapsto \mathfrak{g}$.

For quadratic groups one may also use the Cayley mapping

Definition 7. *The Cayley mapping* $Cay : \mathfrak{g} \mapsto \mathcal{G}$ *is defined as*

$$Cay_\rho(x) = (I - \rho x)^{-1}(I + \rho x), \tag{5}$$

where ρ is a non-negative constant. When $\rho = 1/2$ the Cayley map is a special case of the Padé approximant to the exponential, $Cay_{1/2}(x) = expm(x) + O(x^3)$.

Definition 8. *The inverse of the Cayley mapping invcay* : $\mathcal{G} \mapsto \mathfrak{g}$ *is defined as*

$$invcay_\rho(X) = \frac{1}{\rho}(I+X)^{-1}(X-I). \tag{6}$$

Note that if X has an eigenvalue -1, this transform is undefined.

Definition 9. *The adjoint representation, Ad, and its derivative, ad, are given by the formulae*

$$Ad_A(B) = BAB^{-1}, \tag{7}$$
$$ad_A(B) = [A,B] = AB - BA. \tag{8}$$

3 The Generalized Principal Chiral Model

The principal chiral models (PCMs) which are known also as the sigma models arise in many branches of physics (e.g., classical and quantum physics, condensed matter, high-energy physics, etc...). These models are known to be integrable [33,34,7]. We consider a variation of the sigma models which is the generalized principal chiral model (GPCM) and is given by the action [26,9]

$$\mathcal{L} = \int d^2x\, \eta^{\mu\nu} H_{ab}(g)(g^{-1}\partial_\mu g)^a (g^{-1}\partial_\nu g)^b, \tag{9}$$

where g takes values in the Lie group $\mathcal{G}$, η is the spatial metric and $H_{ab}(g)$ is invertible symmetric $dim\mathcal{G} \times dim\mathcal{G}$ matrix such that

$$H_{ab}(g) = H(g)K_{ab}, \tag{10}$$

where K_{ab} is the bilinear Killing form

$$K_{ab} = Tr(t_a t_b), \;\; t_a, t_b \in \mathfrak{g}. \tag{11}$$

The bilinear form is considered as the metric over the Lie group manifold.

Since we are interested in tensor fields which are attached to two-dimensional flat image space, we will take the metric $\eta^{\mu\nu}$ to be the Euclidean metric $\eta^{\mu\nu} = \delta^{\mu\nu}$. The integration is taken over the two-dimensional image space. The term $A_\mu = g^{-1}\partial_\mu g$ is known as the flat-connection and also as the Yang-Mills gauge field. The flat-connection is an element of the Lie algebra and therefore it may be represented in terms of the generators of the Lie algebra such that

$$A_\mu = g^{-1}\partial_\mu g = A^a_\mu t_a. \tag{12}$$

Also, it obeys the Bianchi identity

$$\partial_\mu A_\nu - \partial_\nu A_\mu + [A_\mu, A_\nu] = 0. \tag{13}$$

In order to obtain the equations of motion we vary the GPCM action with respect to $g^{-1}\delta g$ to obtain

$$-H^{ad}\frac{\delta \mathcal{L}}{\delta \rho^d} = \partial_\mu A^{\mu a} + \Gamma^a_{bc} A^b_\mu A^{\mu c} = 0, \tag{14}$$

where we have used the relation

$$\delta A^a_\mu = \partial_\nu \delta\rho^a - f^a_{bc} A^b_\nu \delta\rho^c, \tag{15}$$

and where $\delta\rho = g^{-1}\delta g$. The connection Γ^a_{bc} is a sum of two parts

$$\Gamma^a_{bc} = S^a_{bc} + \gamma^a_{bc}, \tag{16}$$

where S^a_{bc} is defined as

$$\begin{aligned} S^a_{bc} &= \frac{1}{2}(F^a_{bc} + F^a_{cb}), \\ F^a_{bc} &= (H^{-1})^{ap} f^q_{pb} H_{qc}. \end{aligned} \tag{17}$$

The second part are the Christoffel symbols for the metric H_{ab}

$$\gamma^a_{bc} = \frac{1}{2}(H^{-1})^{ad}(\partial_b H_{cd} + \partial_c H_{bd} - \partial_d H_{bc}). \tag{18}$$

Taking H_{ab} to be constant on the group manifold (i.e., $H_{ab} = K_{ab}$) we have $\gamma^a_{bc} = 0$. The bilinear form over the $SO(N)$ group manifold, for example, is negative definite and is given by $K_{ab} = Tr(t_a t_b) = -2\delta_{ab}$. Plugging H_{ab} into Eq. (17) we have,

$$F^a_{bc} = 2(f^a_{bc} + f^a_{cb}). \tag{19}$$

However, since the structure constants are antisymmetric in their indices (i.e., $f^a_{bc} = -f^a_{cb}$), $F^a_{bc} = 0$ and we are left with the equation of motion

$$\partial_\mu A^{\mu a} = 0. \tag{20}$$

Contracting this equation with the group generators t_a from the right we have

$$\partial_\mu A^{\mu a} t_a = \partial_\mu A^\mu = 0. \tag{21}$$

Since $A^\mu = g^{-1}\partial^\mu g$ we may write the equation of motion in the following form

$$\partial_\mu (g^{-1}\partial^\mu g) = 0. \tag{22}$$

In order to write the gradient descent equations we have to remember that the term $\partial_\mu(g^{-1}\partial^\mu g)$ is in the Lie algebra and therefore the left hand side (LHS) of the gradient descent equation should contain a term which is also in the Lie algebra. Therefore, we suggest the following expression

$$\begin{aligned} g^{-1}\frac{\partial g}{\partial t} &= \partial_\mu (g^{-1}\partial^\mu g) \\ &= \partial_x (g^{-1}\partial_x g) + \partial_y (g^{-1}\partial_y g). \end{aligned} \tag{23}$$

where $\left(g^{-1}\frac{\partial g}{\partial t}\right) \in \mathfrak{g}$.

Multiplying by g from the left of both sides we have

$$\frac{\partial g}{\partial t} = g\partial_\mu(g^{-1}\partial^\mu g). \tag{24}$$

This equation has the form of an orthogonal *ODE* flow

$$\frac{\partial g}{\partial t} = ga, \tag{25}$$

where $g \in \mathcal{G}$ and $a \in \mathfrak{g}$. The numerical solutions of this type of equation were discussed in [10,8]. However, since a and g in our case depend on the spatial coordinates as well as the time, our equation is an orthogonal *PDE* flow. Note that $\frac{\partial g}{\partial t} \in T_g\mathcal{G}$ and the RHS (right hand side) lies also in $T_g\mathcal{G}$ since it is a left-trivialization form of the tangent written as ga. In [10,8] the tangent is written in its right-trivialization form $ag \in T_g\mathcal{G}$. A *different* orthogonal PDE flow which has the same form was discussed in [30,5].

4 Implementation

The implementation of Eq. (24) is not straightforward. In order to get the desired results, the flow has to evolve on the group manifold. This means that the group element g has to preserve its properties (i.e., orthogonality and unit determinant) for every time t. Since the group manifold is not a linear space, we cannot use classical PDEs integration schemes since the group structure will not be preserved along the flow. For the same reason we cannot use finite-difference schemes in order to evaluate the spatial derivatives. Therefore, the first challenge is to find a scheme which enables to evaluate the spatial derivative such as $\partial_\mu g \in T_g\mathcal{G}$ and $g^{-1}\partial_\mu g \in \mathfrak{g}$ for any number of iterations.

This goal is achieved by using the exponential mapping in order to express the Lie group element in terms of the Lie algebra. Then, the spatial derivative of the group element reads $\partial_\mu exp(a)$ where now we have to evaluate the derivative of the exponent. For the scalar case $a \in \mathbb{R}$ and for Abelian groups (where $a, \tilde{a} \in \mathfrak{g}$ commute) the formula for the derivative of the exponent is $\frac{d}{dx}exp(a(x)) = a'(x)exp(a(x))$. However, this formula does not hold for non-Abelian groups such as $SO(N)$ with $N > 2$ since $[a, \tilde{a}] \neq 0$. Therefore, one should apply a different formula.

The correct formula may be written in terms of the *dexp* functionsuch that

$$\frac{\partial}{\partial x}exp(a(x,t)) = dexp_{a(x,t)}a'(x,t)exp(a(x,t)), \tag{26}$$

where a' is the derivative with respect to the spatial coordinate. The *dexp* function is defined as a power series as follows

$$\begin{aligned} dexp_A B &= B + \frac{1}{2!}[A,B] + \frac{1}{3!}[A,[A,B]] \\ &+ \frac{1}{4!}[A,[A,[A,B]]] + \ldots = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} ad_A^k B. \end{aligned} \tag{27}$$

As we have mentioned earlier, the derivative of the exponential mapping should lie in $T_{g(x,t)}\mathcal{G}$. In the right-trivialization form, the tangent may be written as $a(x,t)g(x,t)$ which is exactly the expression in the RHS of Eq. (26). However, since in Eq. (24) the tangent is written in its left-trivialized form $g(x,t)a(x,t) \in T_{g(x,t)}\mathcal{G}$, we should use the left-trivialized version of Eq. (26) which takes the form [10]

$$\frac{\partial}{\partial x} exp(a(x,t)) = exp(a(x,t)) dexp_{-a(x,t)} a'(x,t), \tag{28}$$

where the sign of the commutators in the *dexp* series has been changed by adding a minus sign. Finally, we multiply this equation from the left by $g^{-1} = exp(-a(x,t))$ to obtain

$$g^{-1}\partial_\mu g = dexp_{-a(x,t)} \partial_\mu a(x,t) \ . \tag{29}$$

Then, the flow reads

$$\frac{\partial g(x,t)}{\partial t} = g(x,t)\partial_\mu \left[dexp_{-a(x,t)}(\partial^\mu a(x,t)) \right] . \tag{30}$$

Since the Lie-algebra is a linear space, the partial derivative of a may be evaluated using e.g. the forward finite difference scheme

$$\begin{aligned} \frac{\partial a}{\partial x} &\approx \frac{a(x+h,y) - a(x,y)}{h}, \\ \frac{\partial a}{\partial y} &\approx \frac{a(x,y+h) - a(x,y)}{h}, \end{aligned} \tag{31}$$

where h is the grid size. The partial derivative of the *dexp* function will be evaluated using the backward finite difference scheme. The values of the Lie-algebra elements on the grid will be calculated using the *logm* operator such that

$$logm : g(x,y,t) \mapsto a(x,y,t). \tag{32}$$

In order that the proposed flow evolves on the group manifold we use methods of Lie group integration mainly due to Iserles et al. [10]. We apply the simplest time integration operator which is the Lie-group version of the forward Euler operator. It reads

$$g_{n+1} = \phi(dt\, a(g_n, t_n)) g_n, \tag{33}$$

where dt is the time step, a is the element of the algebra and $\phi : \mathfrak{g} \mapsto \mathcal{G}$. For our 'left-trivialized' flow we may use the following forward Euler operator

$$g_{n+1} = g_n \phi(dt\, a(g_n, t_n)). \tag{34}$$

Therefore, our time step operator reads

$$\begin{aligned} g_{n+1} &= g_n \, exp\left(\partial_\mu \, dexp_{-a(x,t)}(\partial^\mu a(x,t))\right) \\ &= g_n \, exp\left(\partial_\mu \, dexp_{-log\, g(x,t)}(\partial^\mu log\, g(x,t))\right) . \end{aligned} \tag{35}$$

Although on each iteration we have to calculate the *dexp* power series, this calculation is almost immediate since this power series converges very fast. The calculation of the first eight terms is accurate enough where the norm of the eighth element if already of order 10^{-6}.

These calculations may be also be done via the Cayley mapping where the *dcay* function will replace the *dexp* function. However, despite of the Cayley mapping advantages (fast calculations), we have found that it is not a suitable choice for our algorithm. The main reason is that we have to use the *invcay* function instead of the *logm*. As we have pointed out in definition 8, the *invcay* mapping is undefined when $X \in \mathcal{G}$ has an eigenvalue -1. Since some elements of $SO(N)$ do have an eigenvalue -1, this causes the algorithm to be unstable numerically and to diverge.

5 Experiments

We demonstrate in Fig. 1 the isotropic regularization of an orthogonal tensor field using our proposed orthogonal PDE flow. We have built a synthetic tensor field of $SO(3)$ matrices which represents 3D rotations (Fig. 1a). We have created a discontinuity such that the tensor field is divided into two homogenous regions where each region corresponds to a different 3D rotation. The orthogonal matrices are represented in terms of the three column vectors where for $SO(N)$ matrix, these vectors form an N-dimensional orthonormal vector basis. A Gaussian noise has been added to the original field as normally distributed random rotations around the axes. We have applied Eq. (24) to the noisy field (Fig. 1b) for 100 iterations and with a time step of $dt = 0.1$. As expected, the result of the regularization process is a smooth averaged tensor field where the discontinuity has not been preserved (Fig. 1c).

5.1 Anisotropic Regularization

It is clear that Eq. (24) has to be modified in order to obtain an anisotropic regularization of the tensor field. It is well known due to the work by Perona and Malik that this goal may be achieved by replacing the diffusion constant by a spatially dependent function which is a function of the image gradient. This function has to be smooth and monotonically decreasing with $c(0) = 1$ and $c(+\infty) = 0$ whereas it controls the amount of local regularization. We will adopt this attitude. Since in our proposed model the operator which measures the gradients over the tensor field is $g^{-1}\partial_\mu g$, a suitable choice for this function will be

$$c(x, y, t) = \exp(-||g^{-1}(x, y, t)\partial_\mu g(x, y, t)||^2/k^2), \tag{36}$$

where k is the threshold. The flow then takes the form

$$\frac{\partial g}{\partial t} = g\partial_\mu(c(x, y, t)g^{-1}\partial^\mu g). \tag{37}$$

We have tested the modified equation on the same noisy tensor field which is presented in Fig. 1b. The result of the anisotropic regularization is presented

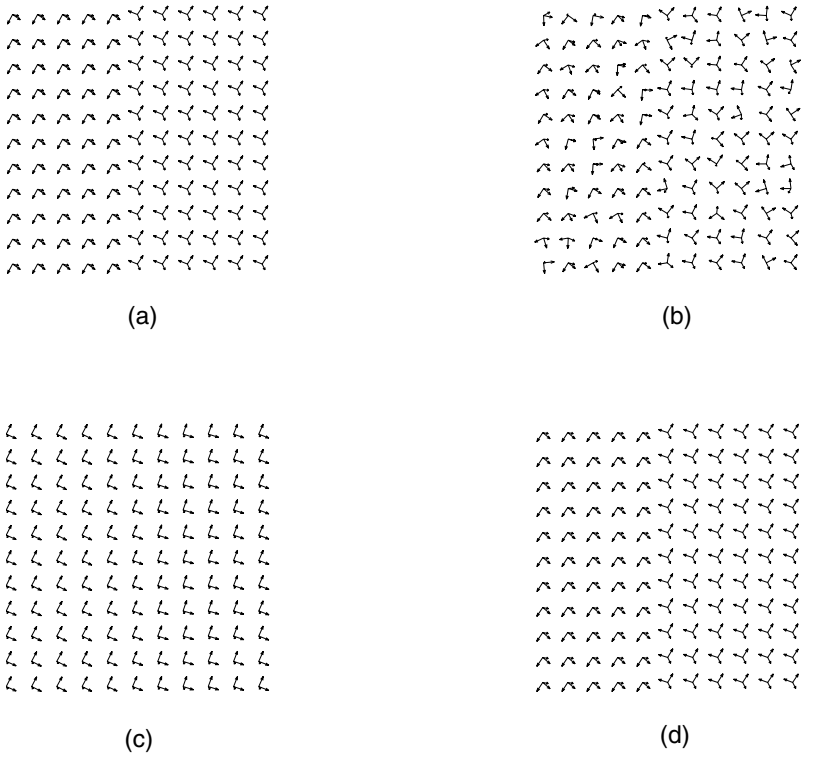


Fig. 1. (a) Original orthogonal tensor field. (b) Noisy field. (c) The result of applying the isotropic orthogonal PDE flow. (d) The results of applying the anisotropic orthogonal PDE flow for 50 iterations and time step $dt = 0.1$. The original tensor field has been recovered.

in Fig. 1d. One can see that at the end of the process the original tensor field has been recovered where the discontinuity has been preserved. In both cases, the isotropic and the anisotropic, the properties of the matrices (orthogonality and determinant one) has been preserved. The threshold k has been set by hand where we have found that its value has to be around one.

The distance between the regularized tensor field and the original tensor field was approximated using the MSE criterion. Let H be the original tensor field and $\hat{H}$ the regularized tensor field, then

$$MSE(\hat{H} - H) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |\hat{H}_{m,n} - H_{m,n}|^2, \tag{38}$$

where MN is the size of the grid. For the result which is presented in Fig. 1d we have $MSE = 0.0057$ which means that the regularized tensor field is very close to the original one. We have repeated the same experiment for the weighting function

$$c(x,y,t) = \frac{1}{1+\left(\frac{||g^{-1}\partial_\mu g||}{k}\right)^2}. \tag{39}$$

We have set the threshold to a value of $k = 0.4$ where the results in this case were as good as in the previous case with $MSE = 0.006$.

6 Summary

In this work we proposed a novel framework to tackle the problem of regularizing of Lie group tensor fields in general and the $SO(N)$ group in particular. This was obtained using a PDE flow which was derived directly from a minimization of the GPCM action. Since this action is defined over Lie-group manifold which is a constrained manifold, we arrived at the constrained flow without any additional operations. We have applied the proposed flow to a three-dimensional orthogonal tensor field in order to regularize it. Then, we have modified the flow à la Perona and Malik in order to obtain an anisotropic regularization of the tensor field. This framework is general where it can be applied to any dimension directly and without any additional complexities.

This work may be extended to many directions. We would like to apply this framework to the regularization problem of DT-MRI data sets. This framework may also be integrated with recent level-set frameworks [14,27,25] in order to consider regularization of tensor fields which are attached to non-flat manifolds. Also, other Lie-group manifolds rather than $SO(N)$ may be considered. All of these challenging problems as well as many others are under investigation.

References

1. Aubert, G. and Kornprobst, P.: Mathematical Problems in Image Processing. Springer-Verlag, New-York (2002).
2. Blomgren, P. and Chan, T. F.: Color TV: Total Variation Methods for Restoration of Vector Valued Images. IEEE Trans. on Image processing **7** (1998) 304–309.
3. Chan, T. and Shen, J.: Variational Restoration of nonflat Image Features: Models and Algorithms. SIAMJ Appl. Math. **61** (2000) 1338–1361.
4. Chen, Y., Guo, W., Zeng, Q., Yan, X., Huang, F., Zhang, H., He, G., Vemuri, B. C., Liu, Y.: Estimation, Smoothing, and Characterization of Apparent Diffusion Coefficient Profiles from High Angular Resolution DWI. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004) **1** (2004) 588–593.
5. Chefd'hotel, C., Tschumperlé, D., Deriche, R. and Faugeras, O.: Regularizing Flows for Constrained Matrix-Valued Images. Journal of Mathematical Imaging and Vision **20** (2004) 147–162.
6. Gallier, J. and Xu, D.: Computing Exponentials of Skew Symmetric Matrices and Logarithms of Orthogonal Matrices. Internation Journal of Robotics and Automation **18** (2003) 10–20.
7. Gürses, M. and Karasu, A.: Integrable Sigma Model. International Journal of Modern Physics A **6** (1990) 487–500.

8. Hairer, E., Lubich, C. and Wanner, G.: Geometric Numerical Integration. Springer-Verlag, Berlin Heidelberg (2002).
9. Hlavatý, L. and Šnobl, L.: Principal Chiral Models on non-Semisimple Group. J. Phys. A **34** (2001) 7795–7809.
10. Iserles, A., Munthe-Kaas, H. Z., Nørsett, S. P. and Zanna, A.: Lie-Group Methods. Acta Numerica **9** (2000) 215–365.
11. Kimmel, R.: Numerical Geometry of Images: Theory, Algorithms, and Applications, Springer-Verlag, New York (2004).
12. Kimmel, R. and Sochen, N.: Orientation Diffusion or How to Comb a Porcupine. Journal of Visual Communication and Image Representation **13** (2001) 238–248.
13. Lenglet, C., Rousson, M., Deriche, R. and Faugeras, O.: Statistics on Multivariate Normal Distributions: A Geometric Approach and its Application to Diffusion Tensor MRI. Technical Report 5242, INRIA, June (2004).
14. Memoli, F., Sapiro, G. and Osher, S.: Solving Variational Problems and Partial Differential Equations Mapping into General Target Manifolds. J. Comput. Phys. **195** (2004) 263–292.
15. Mumford, D. and Shah, J.: Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. Communications on Pure and Applied Mathematics **42** (1989) 577–684.
16. Osher, S. and Rudin, L. I.: Feature-Oriented Image Enhancement using Shock Filters. SIAM Journal of Numerical Analysis **27** (1990) 919–940.
17. Pennec, X., Fillard, P. and Ayache, N.: A Riemmannian Framework for Tensor Computing. International Journal of Computer Vision (2005), to appear.
18. Pennec, X.: Probabilities and Statistics on Riemannian Manifolds: A Geometric approach. Research Report 5093, INRIA, January 2004.
19. Perona, P. and Malik, J.: Scale-Space and Edge Detection using Anisotropic Diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990) 629–639.
20. Perona, P.: Orientation Diffusion. IEEE Trunsactions on Image Processing **7** (1998) 457–467.
21. Rudin, L., Osher S. and Fatemi, E.: Nonlinear Total Variation Based Noise Removal Algorithms. Physica D **60** (1992) 259–268.
22. Sochen, N., Kimmel, R. and Malladi, R.: From High Energy Physics to Low Level Vision. Scale-Space Theories in Computer Vision (1997) 236–247.
23. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis, Cambridge University Press, New York (2001).
24. Sapiro, G. and Ringach, D.: Anisotropic Diffusion of Multivalued Images with Application to Color Filtering. IEEE Trans. On Image Processing **5** (1996) 1582–1586.
25. Shafrir, D. Sochen, N., Deriche, R.: Regularization of Mappings between Implicit Manifolds of Arbitrary Dimension and Codimension, VLSM '05: Proceedings of the 3rd IEEE Workshop on Variational, Geometric and Level-set methods in Computer Vision 2005.
26. Sochen, N.: Integrable Generalized Principal Chiral Models. Phys. Lett. B **391** (1997) 374–380.
27. Sochen, N., Deriche, R. and Lopez-Perez, L.: The Beltrami Flow over Implicit Manifolds, ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision (2003) 832-839.
28. Tang, B., Sapiro, G. and Caselles, V.: Direction Diffusion. International Conference on Computer Vision (1999).

29. Tschumperlé, D.: PhD thesis: PDE's Based Regularization of Multivalued Images and Applications. Universite de Nice-Sophia, Antipolis (2002).
30. Tschumperlé, D. and Deriche, R.: Orthonormal Vector Sets Regularization with PDE's and Application. International Journal of Computer Vision **50** (2002) 237–252.
31. Weickert, J.: Anisotropic Diffusion in Image Processing, Teubner-Verlag, Stuttgart (1998).
32. Weickert, j., Feddern, C., Welk, M., Burgeth, B. and Brox, T.: PDEs for Tensor Image Processing. In: Weickert, J. and Hagen, H. (Eds.): Visualization and Processing of Tensor Fields. Springer, Berlin (2005), to appear.
33. Wiegmann, P. B.: On the Theory of nonabelian Goldstone Bosons in Two Dimesnions: Exact Solution of the $SU(N) \otimes SU(N)$ nonlinear σ Model". Physics Letters B **141** (1984) 217–222.
34. Wiegmann, P. B.: Exact Factorized S-Matrix of the Chiral Field in Two Dimension". Physics Letters B **142** (1984) 173–176.
35. Whitaker, R. and Gerig, G.: Vector-Valued Diffusion. In: B. M. ter Haar Romeny (ed.): Geometry Driven Diffusion in Computer Vision, Kluwer Academic Publishers, The Netherlands, (1994) 93–134.

Nonlinear Inverse Scale Space Methods for Image Restoration

Martin Burger[1], Stanley Osher[2], Jinjun Xu[2], and Guy Gilboa[2]

[1] Industrial Mathematics Institute, Johannes Kepler University, Altenbergerstr. 69, A 4040 Linz, Austria
martin.burger@jku.at
[2] Department of Mathematics, UCLA, Los Angeles, CA 90095, USA
{sjo, jjxu, gilboa}@math.ucla.edu

Abstract. In this paper we generalize the iterated refinement method, introduced by the authors in [8], to a time-continuous inverse scale-space formulation. The iterated refinement procedure yields a sequence of convex variational problems, evolving toward the noisy image.

The inverse scale space method arises as a limit for a penalization parameter tending to zero, while the number of iteration steps tends to infinity. For the limiting flow, similar properties as for the iterated refinement procedure hold. Specifically, when a discrepancy principle is used as the stopping criterion, the error between the reconstruction and the noise-free image decreases until termination, even if only the noisy image is available and a bound on the variance of the noise is known.

The inverse flow is computed directly for one-dimensional signals, yielding high quality restorations. In higher spatial dimensions, we introduce a relaxation technique using two evolution equations. These equations allow accurate, efficient and straightforward implementation.

1 Introduction

The processing of noisy images is a central task in mathematical imaging. Over the last decades, a variety of methods have been proposed ranging from filtering methods to variational approaches to techniques based on the solution of partial differential equations. Since the noise in images is usually expected to be a small scale feature, particular attention has been paid to methods separating scales, in particular those smoothing small scale features faster than large scale ones, so-called *scale space methods*.

Scale space methods are obtained for example by nonlinear diffusion filters [9] of the form

$$\frac{\partial u}{\partial t} = \operatorname{div}(\gamma(|\nabla u|^2)\nabla u), \tag{1}$$

in $\Omega \times \mathbb{R}_+$ with $u(x,0) = f(x)$, where $f : \Omega \to \mathbb{R}$ denotes the given image intensity (Ω being a bounded open subset in $\mathbb{R}^2$) and $u : \Omega \times \mathbb{R}_+ \to \mathbb{R}$ the flow of smoothed images. The diffusion coefficient involves a positive and monotone function γ. For such methods it can be shown that small scales are smoothed

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 25–36, 2005.

faster than large ones, so if the method is stopped at a suitable final time, we may expect that noise is smoothed while large-scale features are preserved to some extent. Diffusion filters can be related to regularization theory (cf. [13]) with certain regularization functionals, but foundations of choosing optimal stopping times are still missing.

Recently, *inverse scale space methods* have been introduced in [12], which are based on a different paradigm. Instead of starting with the noisy image and gradually smoothing it, inverse scale space methods start with the image $u(x,0) = 0$ and approach the noisy image f as time increases, with large scales converging faster than small ones. Thus, if the method is stopped at a suitable time, large scale features may already be incorporated into the reconstruction, while small scale features (including the "noise") are still missing. The inverse scale space method can also be related to regularization theory, in particular iterated Tikhonov regularization (cf. [6,12]) with the same regularization functionals as for diffusion filters. The construction of inverse scale space methods in [12] worked well for quadratic regularization functionals, but did not yield convincing results for other important functionals, in particular for the total variation. In this paper we present a different version of constructing inverse scale space methods as the limit of an iterated refinement procedure previously introduced by the authors (cf. [8]) and demonstrate its applicability to image restoration. With the new approach we are able to perform inverse scale space methods even for the total variation functional, and, in contrast to diffusion filters, we obtain a simple stopping criterion for the methods.

2 Iterated Refinement

In [8], an iterated refinement procedure for total variation restoration was introduced, motivated by the variational problem

$$u = \operatorname*{argmin}_{u \in BV(\Omega)} \left\{ |u|_{BV} + \frac{\lambda}{2} \|f - u\|_{L^2}^2 \right\} \tag{2}$$

for some scale parameter $\lambda > 0$, where $BV(\Omega)$ denotes the space of functions with bounded variation on Ω, equipped with BV seminorm which is formally given by

$$|u|_{BV} = \int_\Omega |\nabla u|,$$

also referred to as the *total variation* (TV) of u. This is the ROF model, introduced to the field of image restoration in [11].

In [8] the authors showed that an iterative procedure (which turned out to be equivalent to Bregman's relaxation method, cf. [1], and proximal point algorithms, cf. [3]) could be used to improve the quality of regularized solutions to inverse problems, based on regularization functionals as in (2). Given a convex functional $J(u)$, e.g., $J(u) = |u|_{BV}$, the *iterated refinement method* defines a sequence $\{u_k\}$ by:

– Set $u_0 = 0,\ p_0 = 0$;
– Given u_{k-1} and p_{k-1},

  • step 1: compute $u_k = \arg\min_u Q_k(u)$ with

$$Q_k : u \longmapsto J(u) - J(u_{k-1}) - \langle p_{k-1}, u - u_{k-1}\rangle + \frac{\lambda}{2}\|f - u\|^2_{L^2}, \qquad (3)$$

  where $\langle\cdot,\cdot\rangle$ denotes the usual duality product;
  • step 2: update the dual variable $p_k = p_{k-1} + \lambda(f - u_k)$.

– Increase k by 1 and continue.

The quantity p_k is identified with $\partial J(u_k)$, as discussed below. This procedure improves the quality of reconstruction for many problems with discontinuous solutions, e.g., deblurring and denoising of images (cf. [7,8]) when $\|f - u\|^2_{L^2}$ is replaced by an appropriate fitting term for individual examples.

Note that the regularization term used in the first step is a so-called generalized *Bregman distance* between u and u_{k-1}, defined as follows,

$$D(u, v) = J(u) - J(v) - \langle u - v, p\rangle,\ p \in \partial J(v),$$

where $\partial J(v)$ is the subgradient of the convex functional $J(v)$. Note that the subgradient may contain more than one element if the functional J is not continuously differentiable, so that the distance would depend on the specific choice of the subgradient. However, we shall suppress the dependence on the subgradient in the notation below. Note that for strictly convex functionals the subgradient contains at most one element and $D(u, v)$ is a scalar distance, that is strictly positive for $u \neq v$. We can then rewrite the functional Q_k minimized in each iteration step as: $Q_k(u) = D(u, u_{k-1}) + \frac{\lambda}{2}\|f - u\|^2_{L^2}$. The Bregman distance and the associated iteration were not used in this fashion previously, but they have been rather employed to minimize functions $H(u, f)$ where H is a (usually complicated) convex function of u having a unique minimum (cf. e.g., [3]).

It was shown in [8] that the iterated refinement method yields a well-defined sequence of minimizers u_k and subgradients $p_k \in \partial J(u_k)$. Moreover, it was proved that the sequence $\{u_k\}$ satisfies $\|u_k - f\|^2_{L^2} \leq \|u_{k-1} - f\|^2_{L^2}$ and if $f \in BV(\Omega)$, then $\|u_k - f\|^2_{L^2} \leq \frac{J(f)}{k}$, i.e., u_k converges monotonically to f in $L^2(\Omega)$ with a rate of $\frac{1}{\sqrt{k}}$. Of course, this convergence result does not give particular information on the behavior of the method as a denoising method, in particular for the typical case of a noisy image f.

The key denoising result obtained in [8] is as follows: for $g \in BV(\Omega)$ we have

$$D(g, u_k) < D(g, u_{k-1}) \quad \text{if } \|f - u_k\|_{L^2} \geq \tau\|g - f\|_{L^2} \qquad (4)$$

for any $\tau > 1$. Thus, the distance between a restored image u_k and a possible exact image g is decreasing until the L^2-distance of f and u_k is larger than the L^2-distance of f and g. This result can be used to construct a stopping rule for our iterative procedure. If we have an estimate of the variance of the noise, i.e.,

$f = g + n, \|n\|_{L^2} = \sigma$, where $g \in BV(\Omega)$ is the noise-free image and n is the noise, then we can stop the iteration at the first k for which $\|f - u_{k+1}\|_{L_2} < \tau\sigma$. The choice of τ allows some freedom to apply the stopping rule also in the case when we only know an upper bound for σ.

It is interesting to note that for TV-based denoising where $J(u) = |u|_{BV}$, the sequence $\{u_k\}$ has the following interpretation (cf. [8]):

- Let $u_1 = \arg\min J(u) + \frac{\lambda}{2}\|f - u\|^2_{L^2}$;
- Define $v_1 = f - u_1$;
- Then inductively for $k \geq 2$, let

$$u_k = \arg\min \left\{ J(u) + \frac{\lambda}{2}\|f + v_{k-1} - u\|^2_{L^2} \right\}$$

and $f + v_{k-1} = u_k + v_k$.

In other words, we *add* the "small scales" v_{k-1} back to f and perform ROF minimization with f replaced by $f + v_{k-1}$ and decompose this function into "large scales" (u_k) plus "small scales" (v_k). This interpretation already yields a multiscale interpretation of the method, since the "small scales" are somehow doubled in each step and so their larger parts can be incorporated into the large scale part after the next iteration. A related procedure involving the ROF model using Tikhonov-Morozov rather than Bregman iteration which multiplies λ by two in each step yields a multiscale method suggested in [6] and analyzed in [14].

3 Inverse Scale Space Methods

In the following we generalize the concept of inverse scale space theory introduced in [6,12] in the context of Tikhonov regularization for the case

$$J(u) = \frac{1}{2}\int_\Omega |\nabla u|^2. \tag{5}$$

We shall derive general inverse scale space methods as a limit of the iterated refinement procedure for $\lambda \to 0$, concentrating in particular on the functional

$$J(u) = \int_\Omega \sqrt{|\nabla u|^2 + \epsilon^2}. \tag{6}$$

Recall that for a special $\lambda > 0$ the iterative refinement procedure constructs sequences u_k^λ of primal and p_k^λ of dual variables such that $u_0^\lambda = p_0^\lambda = 0$,

$$u_k^\lambda = \arg\min_{u \in BV(\Omega)} \left\{ D(u, u_{k-1}^\lambda) + \frac{\lambda}{2}\|f - u\|^2_{L^2} \right\}$$
$$p_k^\lambda = \partial J(u_k^\lambda) = J'(u_k^\lambda).$$

From the Euler-Lagrange equation $J'(u_k^\lambda) - p_{k-1}^\lambda + \lambda(u_k^\lambda - f) = 0$ and $p_k^\lambda = J'(u_k^\lambda)$ we are led to the relation:

$$\frac{p_k^\lambda - p_{k-1}^\lambda}{\lambda} = f - u_k, \quad k = 1, 2, \ldots$$

for the updates. We now reinterpret $\lambda = \Delta t$ as a time step and the difference quotient on the left-hand side as an approximation of a time derivative. Setting $t_k = k\Delta t$, $p^{\Delta t}(t_k) = p_k^{\Delta t}$, and $u^{\Delta t}(t_k) = u_k^{\Delta t}$, we have $p_{k-1}^{\Delta t} = p^{\Delta t}(t_{k-1}) = p^{\Delta t}(t_k - \Delta t)$ and

$$\frac{p^{\Delta t}(t_k) - p^{\Delta t}(t_k - \Delta t)}{\Delta t} = f - u^{\Delta t}(t_k).$$

For $\Delta t \downarrow 0$, dropping subindex k we arrive at the differential equation

$$\frac{\partial p}{\partial t}(t) = f - u(t), \qquad p(t) = J'(u(t)), \tag{7}$$

with initial values given by $u(0) = p(0) = 0$. We assume $\int_\Omega f = 0$.

If the flow $u(t)$ according to (7) exists and is well behaved (which can be shown under reasonable assumptions on the functional J, in particular for total variation, cf. [2]), it is an inverse scale space method in the sense of [6]. This means that the flow starts at $u(0) = 0$ and incorporates finer and finer scales (with the concept of scale depending on the functional J) finally converging again to the image f as $t \to \infty$, i.e. $\lim_{t\to\infty} u(t) = f$. Through (7) the image $u(t)$ flows from the smoothest possible image ($u(0) = 0$) to the noisy image f. Our goal is to use the flow to denoise the image, and therefore we shall use a finite stopping time for the flow. As we shall see below, we can use a simple stopping criterion related to the fitting term $\|u(t) - f\|^2$ only.

3.1 Behaviour for Quadratic Regularization

We start by briefly reviewing the results obtained in [6] for (5). In this case we obtain from the variation of the functional J the equation $p = -\Delta u$ in Ω with boundary condition $\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$. Given p, $\int_\Omega p = 0$, there exists a unique solution $u = -\Delta^{-1}p$ (since $\int_\Omega u = \int_\Omega f = 0$).

A simple manipulation (and the fact that $\frac{\partial f}{\partial t} = 0$) leads us to the equation $\frac{\partial}{\partial t}(u-f) = \Delta^{-1}(u-f) = -A(u-f)$, with the notation $A := -(\Delta)^{-1}$. Thus, the function $v = u - f$ satisfies a simple linear ordinary differential equation in the function space, whose solution is given by $u(t) - f = v(t) = e^{-tA}v(0) = -e^{-tA}f$. It is well-known that A is a positive definite operator and thus, $e^{-tA}f$ decays to zero. As a consequence, the difference $u(t) - f = -e^{-tA}f$ decays exponentially as $t \to \infty$.

3.2 General Convex Regularization

We consider the case of general convex functionals on the digital image space, i.e., $J : \mathbb{R}^N \to \mathbb{R}$. If J is continuously differentiable, we can compute the implicitly defined $u = u(p)$ as the solution of $J'(u(p)) = p$. Note that if J is smooth and strictly convex, the Jacobian of J' given by J'' is positive definite, and hence, the existence of a solution is guaranteed under a standard condition like $J(0) = 0$ by the inverse function theorem.

A possibility to invert the equation for u is the use of the the dual functional (cf. [4]), defined by $J^*(p) := \sup_u \{\langle u, p\rangle - J(u)\}$. Then one can easily show that $p = \partial_u J(u)$ is equivalent to $u = \partial_p J^*(p)$ and we obtain an explicit relation for $u(p)$ provided we can compute the dual functional J^*.

Under the above conditions, we can obtain some important estimates for the inverse scale space flow (7) associated to J. We start by computing the time-derivative of the fitting functional and the (partial) time derivative of u:

$$\frac{1}{2}\frac{d}{dt}\|u(t) - f\|_{L^2}^2 = \langle u(t) - f, \partial_t u(t)\rangle$$

$$\partial_t u(t) = \frac{d}{dt}(\partial_p J^*(p(t))) = H^*(p(t))\partial_t p(t) = -H^*(p(t))(u(t) - f),$$

where we used the notation $H^* = \partial_{pp}^2 J^*$ for the Hessian of the dual functional. If J^* is strictly convex, then there exists a constant $a > 0$ such that $H^*(q) \geq a$ for all $q \in \mathbb{R}$. Hence, combining the above estimates we deduce

$$\frac{1}{2}\frac{d}{dt}\|u - f\|_{L^2}^2 \leq -\langle u(t) - f, H^*(p(t))(u(t) - f)\rangle \leq -a\|u - f\|_{L^2}^2$$

and from a standard ordinary differential equation argument we deduce

$$\|u(t) - f\|_{L^2} \leq e^{-a(t-s)}\|u(s) - f\|_{L^2} \leq e^{-at}\|f\|_{L^2}$$

if $t > s$. Thus, as $t \to \infty$ we obtain convergence $u(t) \to f$ with exponential decay of the error in the L^2-norm.

Note that for the above L^2-estimates, we do not need severe assumptions on f, so that the estimate holds for a clean image as well as for a noisy version used in the algorithm. If we assume that f is a clean image and $J(f) < \infty$, then we can also obtain a decay estimate on the error in the Bregman distance via

$$\begin{aligned}\frac{d}{dt}D(f, u(t)) &= \frac{d}{dt}\big[J(f) - J(u(t)) - \langle f - u(t), p(t)\rangle\big] \\ &= -\langle f - u(t), \partial_t p(t)\rangle = -\|u(t) - f\|_{L^2}^2 \leq -\|f\|^2 e^{-2at}.\end{aligned}$$

In fact

$$\begin{aligned}D(f, u(t)) - D(f, u(s)) &\leq \frac{1}{2a}\|f\|_{L^2}^2[e^{-2at} - e^{-2as}] < 0, \\ D(f, u(t)) &\leq \frac{1}{2a}\|f\|_{L^2}^2[e^{-2at} - 1] + J(f).\end{aligned}$$

All results so far give information about the convergence of u to the clean image f (with finite value $J(f)$) only. In a more practical situation, f is the noisy version of an image g to be restored, and we might even have $J(f) = \infty$, while $J(g) < \infty$. In this case we can derive a similar estimate as follows:

$$\begin{aligned}\frac{d}{dt}D(g, u) &= \langle -\partial_t p(t), g - u(t)\rangle = -\langle f - u(t), g - u(t)\rangle \\ &= -\|f - u(t)\|_{L^2}^2 - \langle f - u(t), g - f\rangle \leq -\frac{\|f - u(t)\|_{L^2}^2}{2} + \frac{\|f - g\|_{L^2}^2}{2}.\end{aligned}$$

The last term on the right-hand side is negative if $\|f-u(t)\|_{L^2} > \|f-g\|_{L^2}$. This means that $u(t)$ approaches any "noise free" image g in the sense of Bregman distance, as long as the residual (the L^2 difference between $u(t)$ and f) is larger than the difference between the noisy image f and g. The left-hand side, namely the residual $\|f-u(t)\|_{L^2}$ can be monitored during the iteration, it only involves the known noisy image f and the computed restoration $u(t)$. The right-hand side is not known for the "real" image g to be restored, since g itself is unknown. However, in typical imaging situations, an estimate for the noise variance is known, which yields a bound of the form $\|f-g\|_{L^2} \leq \sigma$. The above estimate guarantees that the distance $D(g,u)$ is decreasing at least as long as $\|f-u(t)\|_{L^2} > \sigma$, and one could terminate the inverse scale space flow for the minimal t_* such that $\|f-u(t_*)\|_{L^2} = \sigma$. This stopping criterion is well-known in the theory of iterative regularization of inverse problems as the so-called *discrepancy principle* (cf. [5,10] for a detailed discussion).

4 Direct Solution for Regularized Total Variation in 1D

In the following we discuss the numerical solution of (7) in 1D. We recall here that $p(t) \in \partial J(u(t))$ and $u \in \partial J^*(p)$. For the (nondifferentiable) total variation functional we only have (multivalued) subgradients instead of derivatives and therefore we shall instead consider the regularized total variation $J(u) = \int \sqrt{|\nabla u|^2 + \epsilon^2}$, with

$$\partial J(u(t)) = -\mathrm{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2+\epsilon^2}}\right) = p(t). \tag{8}$$

Note that since $\partial J(u+c) = \partial J(u)$, the solution of (8) is not unique if we take the standard assumption that u satisfies homogeneous Neumann boundary condition. In this case, the solvability condition is $\int p(x,t)\,dx = 0$ for all t and the conservation of mean value gives an additional property implying uniqueness, namely $\int u\ dx \equiv \int f\ dx = 0$.

For a fixed time t, we have to solve

$$-\left(\frac{u_x}{\sqrt{u_x^2+\epsilon^2}}\right)_x = p \ \text{ in } D=(a,b), \quad \int_a^b u\,dx = 0, \tag{9}$$

If we denote $q := \dfrac{u_x}{\sqrt{u_x^2+\epsilon^2}}$, then

$$q(x,t) = -\int_a^x p(s,t)\,ds = \int_x^b p(s,t)\,ds \tag{10}$$

and hence, $u_x = \epsilon \dfrac{q}{\sqrt{1-q^2}}$. Therefore,

$$u(x,t) = \epsilon \int_a^x \frac{q(y,t)}{\sqrt{1-q^2(y,t)}}\,dy + C \tag{11}$$

where C is a constant chosen to normalize $\int_a^b u(x)\,dx = 0$. We mention that the same formula for u can be obtained by duality arguments, since J^* can be explicitly calculated in spatial dimension one.

5 Relaxed Inverse Scale Space Flow

In order to implement the process in any dimension we resort to a new kind of approximation.

Consider the following coupled equations:

$$\begin{aligned} u_t &= -p(u) + \lambda(f + v - u), \\ v_t &= \alpha(f - u), \end{aligned} \tag{12}$$

where $u|_{t=0} = v|_{t=0} = 0$ and $\alpha > 0$ is a constant. These equations can be viewed as a time-continuous interpretation of the discrete iterated refinements procedure.

It is easy to see that the steady state of these equations ($u_t = 0, v_t = 0$) is: $u = f$, $v = \frac{p(f)}{\lambda}$. We still would like to show that for any $f \in BV$, the solutions converge to this steady state. We will do the analogue of this only for the simple linear case below. Our numerical evidence indicates that this is indeed true for the regularized TV flow in one and two dimensions.

By taking the time derivative of the first equation in (12) and substituting for v_t by using the second equation, we can view this process as a single, second order in time, evolution:

$$u_{tt} + (\lambda + p_u)u_t + \alpha\lambda u = \alpha\lambda f, \tag{13}$$

where $u|_{t=0} = 0, u_t|_{t=0} = \lambda f$. Here we assume $p(0) = 0$. We now analyze the linear case, where $-p(u) = \Delta u$. Rewriting the flow in the frequency domain ξ, by taking the Fourier transform, the characteristic equation is $r^2 + (\lambda + |\xi|^2)r + \alpha\lambda = 0$, with the solutions

$$r_\pm = \frac{-(\lambda + |\xi|^2) \pm \sqrt{(\lambda + |\xi|^2)^2 - 4\alpha\lambda}}{2}. \tag{14}$$

Using the Taylor approximation $\sqrt{1+x} \approx 1 + \frac{x}{2}$, $x \ll 1$, one can approximate (for frequencies for which $|\xi|^4 \gg \alpha\lambda$)

$$r_\pm \approx \frac{-(\lambda + |\xi|^2)(1 \pm (1 - \frac{2\alpha\lambda}{(\lambda+|\xi|^2)^2}))}{2}, \tag{15}$$

obtaining two roots with different characteristic behavior: $r_+ \approx -(\lambda + |\xi|^2)$, $r_- \approx \frac{-\alpha\lambda}{\lambda+|\xi|^2}$. The Fourier transform of the solution is

$$U(\xi) = (c_+ e^{r_+ t} + c_- e^{r_- t} + 1)F(\xi) \tag{16}$$

where $c_+ = \dfrac{\lambda + r_-}{r_+ - r_-}$, $c_- = \dfrac{\lambda + r_+}{r_- - r_+}$.

We observe that the first part, containing r_+, corresponds to a Gaussian convolution, which decays very quickly with time. The approximate second part, containing r_-, corresponds to the inverse scale-space solution (with some scaling in time) which we actually want to solve. Our numerical results indicate that this kind of behavior extends to the nonlinear process.

From (14) we see that for both parts to have decaying exponential solutions (real valued $r_\pm$) we should require $\alpha \leq \frac{\lambda}{4}$. In the numerical experiments below we set $\alpha = \frac{\lambda}{4}$.

The relaxed inverse scale space flow has about the same complexity as the standard gradient descent to steady state approach of ROF. Moreover, for the linear case, where $J(u) = \frac{1}{2}\int |\nabla u|^2$, as shown in [12], we obtain a step size estimation of $O(1)$ for the direct solution of inverse scale space flow.

6 Results

In this section we present some numerical examples. We solve the 1D problems by the direct approach discussed above and the relaxed inverse flow in order to test and compare their behavior. Motivated by the accordance of one-dimensional results, we only used the computationally cheaper relaxed flow for two-dimensional tests. In all experiments we use a uniform spatial grid of size $h = 1$, a standard assumption in imaging problems.

Example 1. We first consider a 1D denoising problem. Figure 1 shows the clean signal g, the noisy signal f, the noise n ($\sigma = \|n\|_{L^2} = 10 \approx 24\%\|g\|_2$) and the solutions recovered from ROF, direct inverse flow and relaxed inverse flow, respectively. The typical signal loss can be observed in the result of ROF, and as expected the loss is much smaller in the results of the inverse scale flows. This confirms the intuition that the inverse TV flows yields better restorations than the original ROF model.

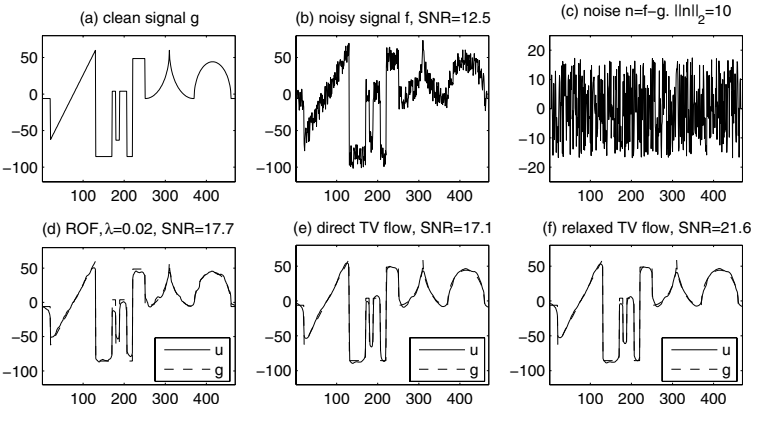


Fig. 1. 1D denoising. (a): clean g; (b): noisy f; (c): noise n; (d)-(f): u recovered from ROF, direct inverse flow and relaxed inverse flow

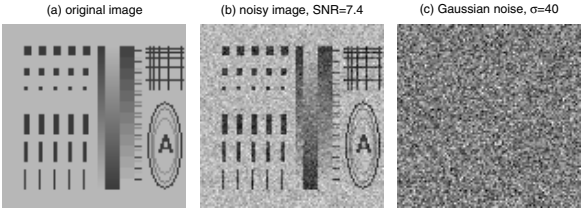


Fig. 2. 2D shape image. (a): original image; (b): noisy image; (c): Gaussian noise.

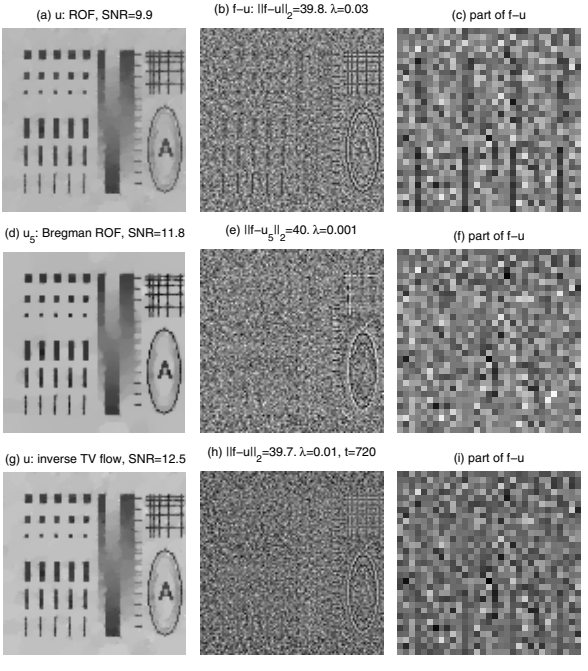


Fig. 3. 2D denoising on shape image. row-by-row: denoised u, residual $w = f - u$ and part of w from ROF, Bregman ROF and relaxed inverse TV flow.

The regularization parameter was chosen as $\epsilon = 1.5$ and for all three experiments, and all restored images u satisfy $\|f - u\|_{L^2} \approx \sigma$. The reason we use a relatively large ϵ here is that the direct solver method is sensitive to numerical error. Moreover, we used the parameter $\lambda = 0.02$ for ROF, time step $\Delta t = 5 \times 10^{-9}$ for the inverse TV flow, and $\lambda = 0.001$, $\Delta t = 0.1$ for the relaxed inverse TV flow. However, the difference of Δt and t in these two inverse flow experiments are only due to the different scaling, the number of time steps until the stopping time $t^* = \min\{t : \|f - u(t)\|_2 \leq \sigma\}$ was reached, is about the same.

Example 2. Now we turn to the denoising of 2D images. In this example we consider an image with different scales and shapes and corrupted by Gaussian

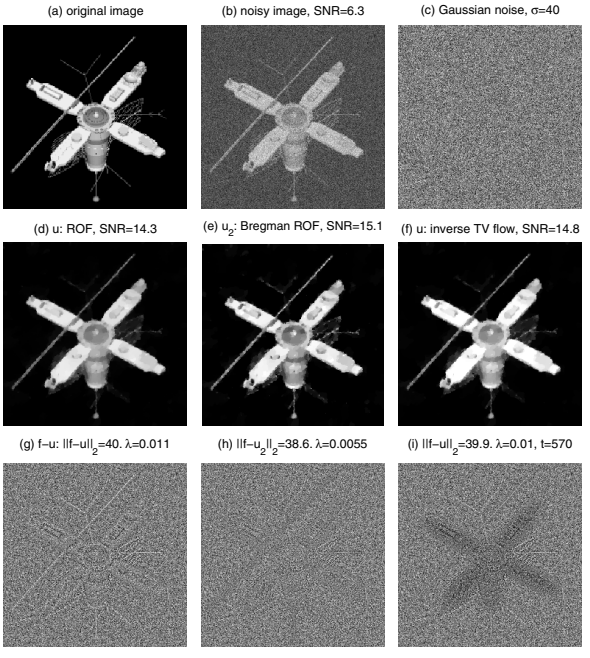


Fig. 4. 2D denoising on satellite image. first row: original, noisy image and noise; second row and third row: u and $w = f - u$ from ROF, Bregman ROF and relaxed inverse TV flow, column-by-column.

noise, which is shown in Figure 2. $SNR(f) = 7.4, \sigma = \|f - g\|_2 = 40$. Figure 3 shows the results from ROF, iterated TV refinement (Bregman ROF), and relaxed inverse TV flow, row-by-row respectively. In each row we display the resulting restoration u and its corresponding residual $w = f - u$ and part of w. One observes that for the ROF model, visible signal is still contained in w (e.g. the small blocks and grids) and it's much smaller in the other two models. This is also quantified by the signal-to-noise ratios $SNR(u) = 9.9, 11.8, 12.5$ obtained for these three experiments respectively.

Example 3. In this final example we denoise a real satellite image with the same methods as used in example 2. Figure 4 shows the data and results. Here we have $SNR(f) = 6.3$ and $SNR(u) = 14.3, 15.1, 14.8$ for ROF, Bregman ROF and inverse TV flow, respectively. Again one can see some visible signal in w (such as the long antenna) for the ROF model, but less signal for the other two experiments.

Acknowledgement

The work of M.B. has been supported by the Austrian National Science Foundation FWF through Project SFB F 013/08. The work of S.O., J.X. and G.G. has

been supported by the NIH through grant U54RR021813, and the NSF through grants DMS-0312222, ACI-0321917 and DMI-0327077. The authors would like to thank Mark Green and Barry Merriman (both UCLA) for useful and stimulating discussions.

References

1. L. M. Bregman. The relaxation method for finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comp. Math. and Math. Phys.*, 7:200–217, 1967.
2. M. Burger, D. Goldfarb, S. Osher, J. Xu, and W. Yin. Inverse total variation flow. in preparation.
3. G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM J. Optim.*, 3:538–543, 1993.
4. I. Ekeland and R. Temam. *Convex Analysis and Variational Problems.* North-Holland Publishers, Amsterdam, 1976.
5. H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
6. C. Groetsch and O. Scherzer. Nonstationary iterated Tikhonov-Morozov method and third order differential equations for the evaluation of unbounded operators. *Math. Methods Appl. Sci.*, 23:1287–1300, 2000.
7. L. He, A. Marquina, and S. Osher. Blind deconvolution using TV regularization and Bregman iteration. *Int. J. of Imaging Systems and Technology*, 5:74–83, 2005.
8. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation based image restoration. *Multiscale Model. and Simul.*, 4:460–489, 2005.
9. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. PAMI*, 12(7):629–639, 1990.
10. R. Plato. The discrepancy principle for iterative and parametric methods to solve linear ill-posed problems. *Numer. Math.*, 75:99–120, 1996.
11. L. I. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, 1992.
12. O. Scherzer and C. Groetsch. Inverse scale space theory for inverse problems. In M.Kerckhove, editor, *Scale-Space and Morphology in Computer Vision, Lecture Notes in Comput. Sci. 2106*, pages 317–325. Springer, New York, 2001.
13. O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *J. Math. Imaging and Vision*, 12:43–63, 2000.
14. E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (BV, L^2) decompositions. *Multiscale Model. Simul.*, 2:554–579, 2004.

Towards PDE-Based Image Compression

Irena Galić[1,2], Joachim Weickert[1], Martin Welk[1],
Andrés Bruhn[1], Alexander Belyaev[2], and Hans-Peter Seidel[2]

[1] Mathematical Image Analysis Group, Faculty of Math. and Computer Science,
Saarland University, Building 27, 66041 Saarbrücken, Germany
{galic, weickert, welk, bruhn}@mia.uni-saarland.de
[2] Max-Planck Institute for Informatics, Stuhlsatzenhausweg 85,
66123 Saarbrücken, Germany
{belyaev, hpseidel}@mpi-sb.mpg.de

Abstract. While methods based on partial differential equations (PDEs) and variational techniques are powerful tools for denoising and inpainting digital images, their use for image compression was mainly focussing on pre- or post-processing so far. In our paper we investigate their potential within the decoding step. We start with the observation that edge-enhancing diffusion (EED), an anisotropic nonlinear diffusion filter with a diffusion tensor, is well-suited for scattered data interpolation: Even when the interpolation data are very sparse, good results are obtained that respect discontinuities and satisfy a maximum–minimum principle. This property is exploited in our studies on PDE-based image compression. We use an adaptive triangulation method based on B-tree coding for removing less significant pixels from the image. The remaining points serve as scattered interpolation data for the EED process. They can be coded in a compact and elegant way that reflects the B-tree structure. Our experiments illustrate that for high compression rates and non-textured images, this PDE-based approach gives visually better results than the widely-used JPEG coding.

1 Introduction

In recent years several partial differential equations (PDEs) and variational techniques have shown their usefulness in so-called inpainting methods [19,2,4,13,27]. Here one aims at filling in missing informations in certain corrupted image areas by means of second or higher-order PDES. The basic idea is to regard the given image data as Dirichlet boundary conditions, and interpolate the data in the inpainting regions by solving appropriate boundary value problems. Related variational and PDE methods have also been investigated for more classical interpolation problems such as zooming into an image by increasing its resolution [3,18]. For such interpolation problems with data given on a regular grid, these techniques are in competition with cubic or quintic splines, radial basis functions and sinc-based interpolation techniques; see e.g. [17,20]. If the data are not available on a regular grid, scattered data interpolation techniques have been proposed [11,22], among which radial basis functions such as thin plate splines [9] are popular and well-performing.

Interestingly, not many of the variational and PDE-based interpolation and inpainting techniques have been used for scattered data interpolation. It seems that the sparsity

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 37–48, 2005.

of the scattered data constitutes a real challenge for these techniques: While second-order PDEs may satisfy a maximum–minimum principle, they often create singularities at isolated interpolation points in 2-D. Higher-order PDEs, on the other hand, may give smoother solutions, but the violation of an extremum principle can lead to undesirable over- and undershoots; see e.g [3].

The goal of the present paper is twofold: First we address the before mentioned problem by investigating a partial differential equation that has not been considered for interpolation problems before, namely *edge-enhancing anisotropic diffusion (EED)* [28]. It uses a diffusion tensor that allows smoothing along discontinuities while inhibiting smoothing across them. In our experiments we will see that this technique performs favourably in the context of scattered data interpolation. A second goal of our paper is to investigate if this property can be used for lossy image compression. While contemporary image compression methods are dominated by concepts that involve the discrete cosine transform (such as the widely-used JPEG standard [23]) or the discrete wavelet transform (in JPEG2000 [26]), our goal is to give a proof-of-concept that there are alternatives where PDEs may be beneficial. The basic idea is to reduce the image data to a well-adapted set of significant sparse points that can be coded in an efficient way. Decoding is accomplished by using these scattered data and interpolating them by means of edge-enhancing anisotropic diffusion. As a tool for creating a useful sparse point representation, we consider the *B-tree triangular coding (BTTC)* by Distasi et al. [8], since it is relatively simple and allows an efficient coding of the sparsified image data.

Our paper is organised as follows. In Section 2 we describe PDE-based interpolation techniques and show that scattered data interpolation with EED performs particularly well. In Section 3 we review the BTTC model for image coding and describe how it can be coupled with PDE-based interpolation. Experiments on EED-based image coding are presented in Section 4, and the paper is concluded with a summary in Section 5.

Related Work. In the context of image compression, PDEs and related variational techniques have mainly been used as a preprocessing step before coding [10,16] or as a post-processing tool for removing coding artifacts [1,12,21,29,30]. Our works differs from these papers by the fact that we use PDEs within the decoding step rather than as pre- or postprocessing tools. Chan and Zhou [5] used total variation regularisation in order to modify the coefficients in a wavelet decomposition such that oscillatory edge artifacts are reduced, while Solé et al. [25] investigate three PDEs for interpolating structures in digital elevation maps and report the most favourable results with the Laplacian operator. An interesting coding scheme that exploits scattered data interpolation has been proposed by Demaret et al. [7]. They construct an adaptive Delaunay triangulation that is used for decoding the image by linear interpolation. Their experiments show that it can be an alternative to JPEG 2000 coding for texture-free images.

2 PDE-Based Interpolation

We start by considering a PDE approach to image interpolation. First we discuss a general model, then we investigate four possibilities for smoothing operators, and finally we present an experiment that illustrates their properties.

2.1 General Model

Let $\Omega \subset \mathbb{R}^n$ denote an n-dimensional image domain. We want to recover some unknown scalar-valued function $v : \Omega \to \mathbb{R}$, from which we know its values on some subset $\Omega_1 \subset \Omega$. Our goal is to find an interpolating function $u : \Omega \to \mathbb{R}$ that is smooth and close to v in $\Omega \setminus \Omega_1$ and identical to v in Ω_1.

We may embed this problem in an evolution setting with some evolution parameter (the "time") $t \geq 0$. It solution $u(x,t)$ gives the desired interpolating function as its steady state ($t \to \infty$). We initialise the evolution with some function $f : \Omega \to \mathbb{R}$ that is identical to v on Ω_1 and that is set to some arbitrary value (e.g. to 0) on $\Omega \setminus \Omega_1$:

$$f(x) := \begin{cases} v(x) & \text{if } x \in \Omega_1 \\ 0 & \text{else.} \end{cases} \tag{1}$$

We consider the evolution

$$\partial_t u \;=\; (1-c(x))\, Lu \;-\; c(x)\,(u-f) \tag{2}$$

with f as initial value,

$$u(x,0) \;=\; f(x), \tag{3}$$

and reflecting (homogeneous Neumann) boundary conditions on the image boundary $\partial\Omega$. The function $c : \Omega \to \mathbb{R}$ is the characteristic function on Ω_1, i.e.

$$c(x) := \begin{cases} 1 & \text{if } x \in \Omega_1 \\ 0 & \text{else,} \end{cases} \tag{4}$$

and L is some elliptic differential operator. The idea is to solve the steady state equation

$$(1-c(x))\, Lu \;-\; c(x)\,(u-f) = 0 \tag{5}$$

with reflecting boundary conditions. In Ω_1 we have $c(x) = 1$ such that the interpolation condition $u(x) = f(x) = v(x)$ is fulfilled. In $\Omega \setminus \Omega_1$ it follows from $c(x) = 0$ that the solution has to satisfy $Lu = 0$. This elliptic PDE can be regarded as the steady state of the evolution equation

$$\partial_t u = Lu \tag{6}$$

with Dirichlet boundary conditions given by the interpolation data on Ω_1.

2.2 Specific Smoothing Operators

Regarding the elliptic differential operator L, many possibilities exist. The simplest one uses the Laplacian $Lu := \Delta u$ leading to *homogeneous diffusion* [15]:

$$\partial_t u = \Delta u. \tag{7}$$

A prototype for a higher order differential operator is the biharmonic operator $Lu := -\Delta^2 u$ giving the *biharmonic smoothing* evolution

$$\partial_t u = -\Delta^2 u. \tag{8}$$

Using it for interpolation comes down to thin plate spline interpolation [9], a rotationally invariant multidimensional generalisations of cubic spline interpolation. Note that only

the second-order differential operators allow a maximum–minimum principle, where the values of u stay within the range of the values of f in Ω_1.

Nonlinear isotropic diffusion processes are governed by $Lu := \operatorname{div}(g(|\nabla u|^2)\,\nabla u)$. This gives [24]

$$\partial_t u = \operatorname{div}(g(|\nabla u|^2)\,\nabla u) \tag{9}$$

where the diffusivity function g is decreasing in its argument, since the goal is to reduce smoothing at edges where $|\nabla u|$ is large. One may e.g. choose the *Charbonnier diffusivity* [6]

$$g(s^2) = \frac{1}{\sqrt{1 + s^2/\lambda^2}} \tag{10}$$

with some contrast parameter $\lambda > 0$. Since (9) uses a scalar-valued diffusivity we name this process *isotropic* (in contrast to the nomenclature in [24]).

Real anisotropic behaviour is possible when a diffusion tensor is used. As a prototype for nonlinear anisotropic diffusion filtering we consider *edge-enhancing diffusion (EED)* [28]. The idea is to reduce smoothing across edges while still permitting diffusion along them. The EED diffusion tensor has one eigenvector parallel to ∇u_σ, where u_σ is obtained from convolving u with a Gaussian with standard deviation σ. The corresponding eigenvalue is given by $g(|\nabla u_\sigma|^2)$ with a diffusivity function such as (10). The other eigenvectors are orthogonal to ∇u_σ and have corresponding eigenvalues 1. If we use the convention to extend a scalar-valued function $g(x)$ to a matrix-valued function $g(A)$ by applying g to the eigenvalues on A and leaving the eigenvectors unchanged, then EED can be formally linked to $Lu := \operatorname{div}(g(\nabla u_\sigma \nabla u_\sigma^\top)\,\nabla u)$. Hence, its evolution is governed by

$$\partial_t u = \operatorname{div}(g(\nabla u_\sigma \nabla u_\sigma^\top)\,\nabla u). \tag{11}$$

2.3 Experiments on Interpolation

In order to evaluate the potential of the preceding PDEs for scattered data interpolation, we have discretised them with central finite differences in space. For the diffusion equations, we performed a semi-implicit time discretisation with SOR as solver for the linear systems of equations, while for biharmonic smoothing an explicit scheme was used. Runtimes for a non-optimised C implementation on a 1.5 GHz Centrino laptop range between a few seconds and several minutes for a 256×256 image.

In Figure 1 we present an experiment that illustrates the use of the different smoothing operators for scattered data interpolation. It depicts a zoom into the famous *lena* image, where 2 percent of all pixels have been chosen randomly as scattered interpolation points. We observe that homogeneous diffusion is not very suitable for scattered data interpolation, since it creates singularities at the interpolation points. This can be avoided with interpolation using biharmonic smoothing. It gives fairly good results, but suffers from over- and undershoots near edges due to the violation of an extremum principle (see e.g. the shoulder). Interestingly, going from homogeneous diffusion to nonlinear isotropic diffusion does not give an improvement: Although nonlinear isotropic diffusion may allow discontinuities, its interpolant is too flat and tends to keep many interpolation points as isolated singularities. The fact that EED, on the other hand, gives the best results shows the importance of the anisotropic behaviour. Its ability to smooth

along edges seems to be very beneficial for avoiding singularities at interpolation points. Moreover, this second-order PDE respects a maximum–minimum priciple, such that the solution is within the greyscale bounds of the interpolation points.

Fig. 1. (a) Top left: Zoom into the test image *lena*, 256×256 pixels. **(b) Top middle:** Grey values of the scattered interpolation points (2 percent of all pixels, chosen randomly). **(c) Top right:** Interpolation by linear diffusion. **(d) Bottom left:** Biharmonic smoothing. **(e) Bottom middle:** Nonlinear isotropic diffusion ($\lambda = 0.1$). **(f) Bottom right:** EED ($\lambda = 0.1$, $\sigma = 1$).

Table 1. Average absolute errors (AAE) for the PDEs used for scattered data interpolation in Figure 1

PDE method	AAE
homogeneous diffusion (7)	16.977
biharmonic smoothing (8)	15.787
Charbonnier diffusion (9)	21.864
edge-enhancing diffusion (11)	14.584

Our visual impression is confirmed by Table 1, in which we have computed the average absolute error (AAE) between the interpolated image and the original image. For two images (u_{ij}) and (v_{ij}) with N pixels, the AAE is defined as

$$\mathrm{AAE}(u, v) = \frac{1}{N} \sum_{i,j} |u_{ij} - v_{ij}|. \tag{12}$$

Nonlinear isotropic diffusion performes worst, followed by homogeneous diffusion and biharmonic smoothing, while EED gives the smallest interpolation error. This shows that for scattered data interpolation, EED is a very promising PDE that has not been investigated in this context before.

3 Image Coding by Binary Trees

Now that we have seen that EED is useful for scattered data interpolation, we want to exploit this technique for image compression. To this end we have to couple it with a method that provides a useful sparse image representation with scattered data.

3.1 Creating Scattered Interpolation Points

Our image compression and decompression scheme relies on an adaptive sparsification of the image data by means of the triangulation from *B-tree triangular coding (BTTC)* [8]. In this coding, an image is decomposed into a number of triangular regions such that within each region it can be recovered in sufficient quality by interpolation from the vertices. In our case, all regions are isosceles rectangular triangles. The decomposition into triangles then is stored in a binary tree structure.

In order to describe the compression procedure, let us assume we have an image $v = (v_{ij})$ of size $(2^m + 1) \times (2^m + 1)$. Smaller images should be filled up to such a size in a suitable way. Initially, the image is split by one of its diagonals into two triangles. The four image corners $(1, 1)$, $(1, 2^m + 1)$, $(2^m + 1, 1)$ and $(2^m + 1, 2^m + 1)$ are vertices of these two triangles.

To refine this initial configuration, an approximation (u_{ij}) of the image (v_{ij}) is calculated by using only the grey values from the vertices and interpolating linearly within each triangle. If the error $e_{ij} := |u_{ij} - f_{ij}|$ satisfies $e_{ij} \leq \varepsilon$ for all pixels (i, j), with a given tolerance parameter $\varepsilon > 0$, the representation by triangles is considered sufficiently fine. Otherwise, for each pixel (i, j) for which $e_{ij} > \varepsilon$ holds, the triangle which contains (i, j) is split into two similar triangles by the height on its hypotenuse. The centre of the hypotenuse thereby becomes an additional vertex of the representation. By recalculating approximation errors within the new smaller triangles, it is determined whether to split these again etc. Since the approximation error is zero at vertices, triangles with legs of length 1 are not split further, which guarantees that the recursive splitting terminates. Moreover, vertices throughout the process have integer coordinates. Which pixels are vertices is indicated in a vertex mask of size equal to the image that is generated during the triangulation.

One point which needs additional consideration is the treatment of pixels located on the sides of triangles during the splitting process. If the error bound is violated in such a pixel, it is sufficient for our compression and decompression method to split one of the two adjacent triangles. This allows to reduce the number of triangles noticeably since in regions with fine details, a large number of small triangles occur, and many pixel positions then happen to be located on sides.

3.2 Coding

To efficiently store the triangulation, we notice that the hierarchical splitting of triangles gives rise to a binary tree structure. Each triangle occurring during the splitting process is represented by a node while leaves correspond to those triangles which are not divided further. When a triangle is split, its two subtriangles become the children of its representing node. To store the structure of the tree, one traverses the tree and stores one bit per node: a 1 for a node that has children, and a 0 for a leave. Preorder or level-order traversal are equally possible. Note that by the tree structure, the vertex mask is fully determined. Further space in storing the tree is saved by measuring globally the minimal and maximal depth of the tree. Only for nodes at intermediate levels, the corresponding bits are stored.

For coding the grey values in all vertices, we first zig zag traverse the sparse image created with the binary tree structure and store it in a sequence of grey values. This sequence is then compressed with Huffman coding [14], a lossless variable-length prefix code that assigns smaller codes to more frequent characters. It also uses a tree structure.

Our entire coded image format then reads as follows:

- image size (4 bytes)
- minimal and maximal depth of the binary tree (together 2 bytes)
- binary string encoding binary tree structure (1 bit for each node between minimal and maximal depth, filled up with zeros to the next byte boundary)
- first grey-value in a sequence of grey values (1 byte)
- minimal and maximal depth of the Huffman coded binary tree (2 bytes)
- binary string for Huffman-coded binary tree (1 bit for each node between minimal and maximal depth, filled up with zeros to the next byte boundary)
- Huffman dictionary (less than 256 bytes)
- sequence of Huffman-coded grey values

We further enhanced this coding by a (lossy) requantisation step that reduced the number of grey values in the initial image from 256 to 64.

3.3 Decoding

Decompression takes place in two steps. In the first step, the vertex mask is recovered from the binary tree representation, and the stored grey values are placed at the appropriate pixel positions to give the sparse image. To recover the vertex mask, the tree is generated in the same order as it was stored. Along with generating nodes, vertex positions are calculated and marked in the vertex mask.

The second step consists in the interpolation of the image, where the vertex mask becomes the interpolation mask. In the BTTC scheme of Distasi et al. [8], linear interpolation within each traingle is used. In the sequel we will denote this technique by BTTC-L.

Since we have already seen that EED performs favourably as a scattered data interpolant, it is natural to renounce the linear interpolation step in the BTTC-L method and apply EED to the interpolation mask that has been created by the BTTC method.

We abbreviate this novel method by BTTC-EED. Note that in contrast to BTTC-L, BTTC-EED does not rely on the triangulation, only on its vertices as interpolation points.

4 Experiments on Compression

Let us now investigate the effects of our EED-based interpolation in the context of image coding. Figures 2 and 3 show two test images and their compressed versions using BTTC-EED, respectively. We have chosen the threshold parameter ϵ such that compressions of 0.8, 0.4 and 0.2 bits per pixel (bpp) are achieved. Since the usual coding uses 1 byte per pixel, this comes down to compression ratios of 1:10, 1:20 and 1:40. In Fig. 3, we display both the coded pixels with their corresponding grey values, and the result after scattered data interpolation with EED. We observe that even at high compression rates, fairly realistic results are possible.

Fig. 2. Test images, 257×257 pixels. **(a) Left:** *trui*. **(b) Right:** *peppers*.

In order to illustrate the differences between BTTC-EED and BTTC-L as well as JPEG, we depict the corresponding results in Fig. 4. We perform this comparison at the high compression rate of 1:40 (or equivalently 0.2 bpp) where the visual differences are well visible. We observe that JPEG coding suffers from severe block artifacts that result from the fact that the discrete cosine transform is computed within blocks of 8×8 pixels. The BTTC-L method, on the other hand, creates artifacts where the underlying triangulation becomes visible. Since BTTC-EED only uses the interpolation points from the BTTC method, but not the corresponding triangulation, it is clear that this method cannot suffer from such a shortcoming. If not enough data are available, its interpolation tends to be on the smoother side. It gives the visually most convincing results among the three methods.

This visual impression is also confirmed by the quantitative measurements in Table 2, where the average absolute error is listed. We see that at the compression rate 1:40, JPEG performs worst, BTTC-L is in the midfield, and BTTC-EED gives the best results.

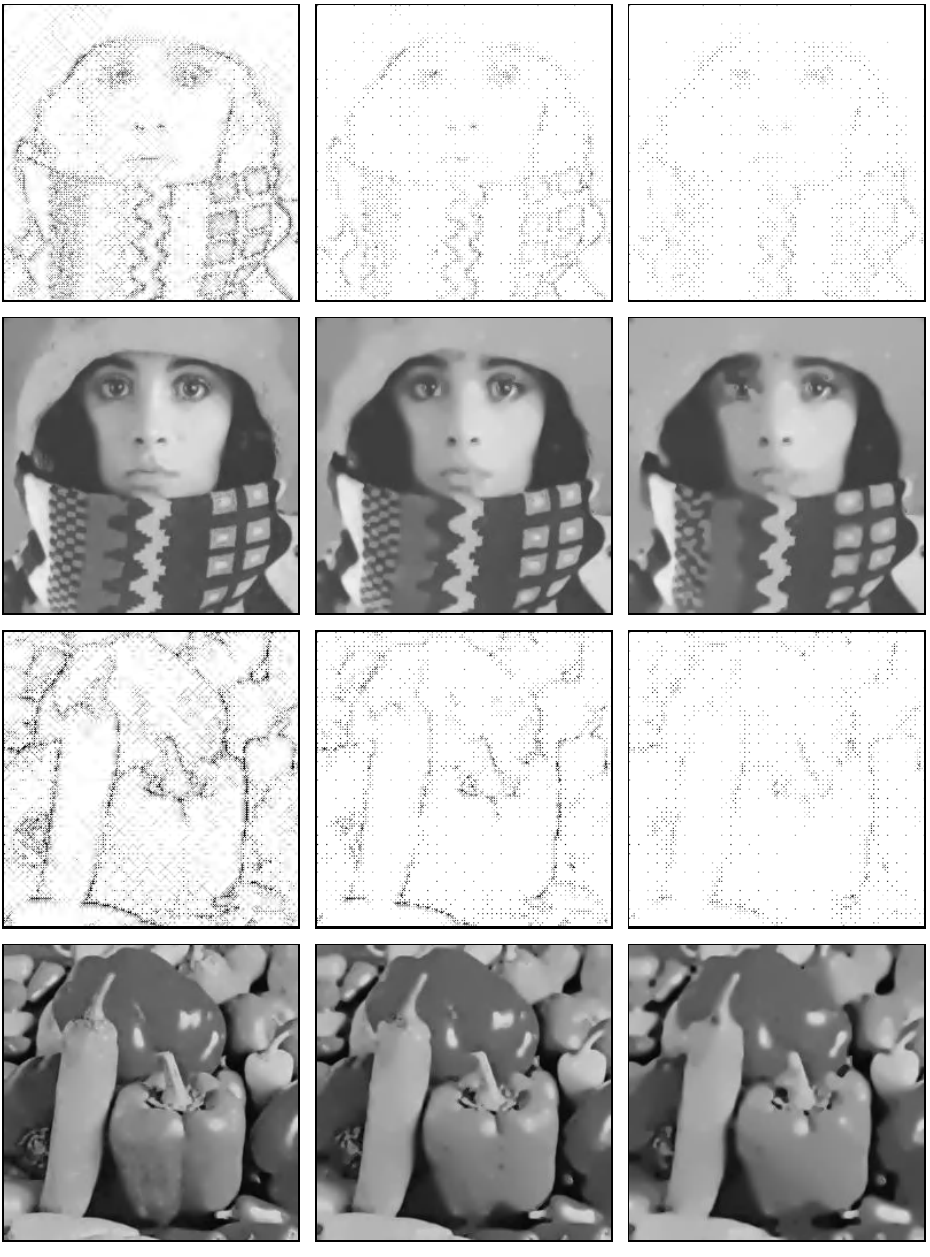


Fig. 3. First row, left to right: Adaptive sparsification of *trui*, using BTTC with compression to 0.8 bpp, 0.4 bpp, 0.2 bpp. **Second row, left to right:** Corresponding EED-based interpolation. **Third row, left to right:** Adaptive sparsification of *peppers*, using BTTC with compression to 0.8 bpp, 0.4 bpp, 0.2 bpp. **Fourth row, left to right:** Corresponding EED-based interpolation.

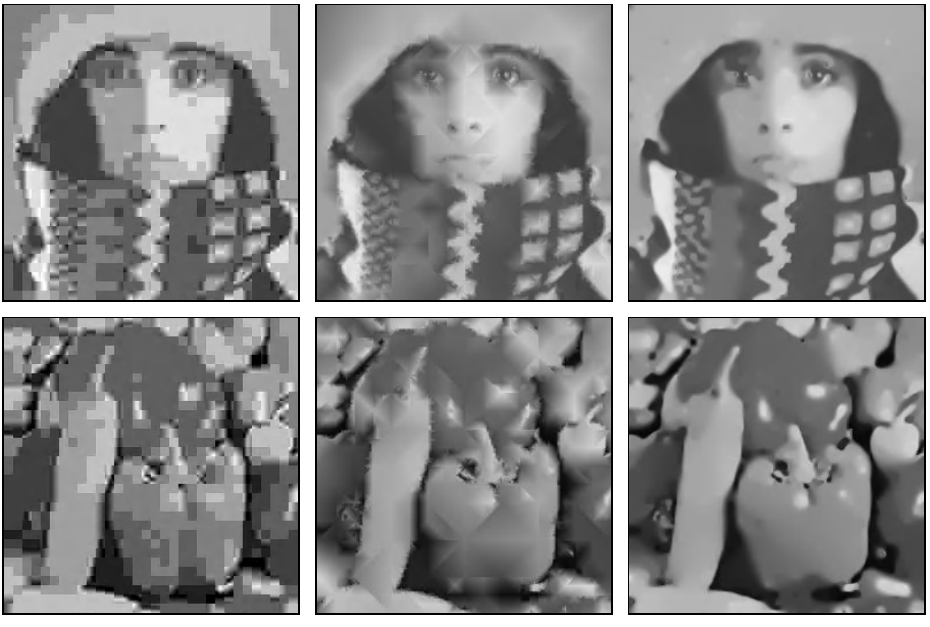


Fig. 4. Comparison at high compression rates (0.2 bpp) for the test images *trui* and *peppers*. **Left column:** JPEG. **Middle column:** BTTC-L. **Right column:** BTTC-EED.

Table 2. Comparison of the average absolute error for the different images and compression methods at 0.2 bpp

compression	trui	peppers
JPEG	11.25	12.99
BTTC-L	8.63	11.22
BTTC-EED	8.45	9.99

5 Conclusions

In this paper we presented a proof-of-concept that edge-enhancing anisotropic diffusion (EED), a PDE originally designed for denoising, can also be useful for scattered data approximation and for image compression. In the latter case we sparsified the image data by B-tree triangular coding, and used EED as a scattered data interpolant for decoding. Our experiments indicate that this strategy is particularly useful when high compression rates become necessary.

Since modern image compression methods have reached a high degree of sophistication, it is evident that our paper can only serve as a first step towards PDE-based image compression. In our ongoing work we are investigating different options for further performance improvements and a more detailed performance evaluation including also upcoming standards such as JPEG 2000.

Acknowledgements. Our research is partly funded by the *International Max-Planck Research School (IMPRS)*. This is gratefully acknowledged. Joachim Weickert also thanks Vicent Caselles for interesting discussions on EED-based interpolation during a stay at the University Pompeu Fabra, Barcelona.

References

1. F. Alter, S. Durand, and J. Froment. Adapted total variation for artifact free decompression of JPEG images. *Journal of Mathematical Imaging and Vision*, 23(2):199–211, September 2005.
2. M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. SIGGRAPH 2000*, pages 417–424, New Orleans, LI, July 2000.
3. V. Caselles, J.-M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7(3):376–386, March 1998.
4. T. F. Chan and J. Shen. Non-texture inpainting by curvature-driven diffusions (CDD). *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.
5. T. F. Chan and H. M. Zhou. Total variation improved wavelet thresholding in image compression. In *Proc. Seventh International Conference on Image Processing*, volume II, pages 391–394, Vancouver, Canada, September 2000.
6. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.
7. L. Demaret, N. Dyn, and A. Iske. Image compression by linear splines over adaptive triangulations. Technical report, Dept. of Mathematics, University of Leicester, UK, January 2005.
8. R. Distasi, M. Nappi, and S. Vitulano. Image compression by B-tree triangular coding. *IEEE Transactions on Communications*, 45(9):1095–1100, 1997.
9. J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Mathematical Models and Numerical Analysis*, 10:5–12, 1976.
10. G. E. Ford, R. R. Estes, and H. Chen. Scale-space analysis for image sampling and interpolation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 165–168, San Francisco, CA, March 1992.
11. R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38:181–200, 1982.
12. A. Gothandaraman, R. Whitaker, and J. Gregor. Total variation for the removal of blocking effects in DCT based encoding. In *Proc. 2001 IEEE International Conference on Image Processing*, volume 2, pages 455–458, Thessaloniki, Greece, October 2001.
13. H. Grossauer and O. Scherzer. Using the complex Ginzburg–Landau equation for digital impainting in 2D and 3D. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 225–236, Berlin, 2003. Springer.
14. D. A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the IRE*, 40:1098–1101, 1952.
15. T. Iijima. Basic theory on normalization of pattern (in case of typical one-dimensional pattern). *Bulletin of the Electrotechnical Laboratory*, 26:368–388, 1962. In Japanese.
16. I. Kopilovic and T. Szirányi. Artifact reduction with diffusion preprocessing for image compression. *Optical Engineering*, 44(2):1–14, February 2005.
17. T. Lehmann, C. Gönner, and K. Spitzer. Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075, November 1999.

18. F. Malgouyres and F. Guichard. Edge direction preserving image zooming: A mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, 39(1):1–37, 2001.
19. S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Proc. 1998 IEEE International Conference on Image Processing*, volume 3, pages 259–263, Chicago, IL, October 1998.
20. E. Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, March 2002.
21. P. Mrázek. *Nonlinear Diffusion for Image Filtering and Monotonicity Enhancement*. PhD thesis, Czech Technical University, Prague, Czech Republic, June 2001.
22. G. M. Nielson and J. Tvedt. Comparing methods of interpolation for scattered volumetric data. In D. F. Rogers and R. A. Earnshaw, editors, *State of the Art in Computer Graphics: Aspects of Visualization*, pages 67–86. Springer, New York, 1994.
23. W. B. Pennebaker and J. L. Mitchell. *JPEG: Still Image Data Compression Standard*. Springer, New York, 1992.
24. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.
25. A. Solé, V. Caselles, G. Sapiro, and F. Arandiga. Morse description and geometric encoding of digital elevation maps. *IEEE Transactions on Image Processing*, 13(9):1245–1262, September 2004.
26. D. S. Taubman and M. W. Marcellin, editors. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer, Boston, 2002.
27. D. Tschumperlé and R. Deriche. Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):506–516, April 2005.
28. J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing Supplement*, 11:221–236, 1996.
29. S. Yang and Y.-H. Hu. Coding artifact removal using biased anisotropic diffusion. In *Proc. 1997 IEEE International Conference on Image Processing*, volume 2, pages 346–349, Santa Barbara, CA, October 1997.
30. S. Yao, W. Lin, Z. Lu, E. P. Ong, and X. Yang. Adaptive nonlinear diffusion processes for ringing artifacts removal on JPEG 2000 images. In *Proc. 2004 IEEE International Conference on Multimedia and Expo*, pages 691–694, Taipei, Taiwan, June 2004.

Color Image Deblurring with Impulsive Noise

Leah Bar[1], Alexander Brook[2], Nir Sochen[3], and Nahum Kiryati[1]

[1] School of Electrical Engineering,
Tel-Aviv University, Israel
[2] Dept. of Mathematics, Technion, Israel
[3] Dept. of Applied Mathematics, Tel-Aviv University, Israel

Abstract. We propose a variational approach for deblurring and impulsive noise removal in multi-channel images. A robust data fidelity measure and edge preserving regularization are employed. We consider several regularization approaches, such as Beltrami flow, Mumford-Shah and Total-Variation Mumford-Shah. The latter two methods are extended to multi-channel images and reformulated using the Γ-convergence approximation. Our main contribution is in the unification of image deblurring and impulse noise removal in a multi-channel variational framework. Theoretical and experimental results show that the Mumford-Shah and Total Variation Mumford Shah regularization methods are superior to other color image restoration regularizers. In addition, these two methods yield a denoised edge map of the image.

1 Introduction

Image deblurring and denoising are classical problems that have been extensively studied. Consider the standard linear and space invariant blur model with additive noise. Let z denote the blurred and noisy image, h the blur kernel, u the original image and n the noise. Thus, $z = h * u + n$. In the multi-channel case,

$$z^\sigma = h^\sigma * u^\sigma + n^\sigma,$$

where σ indicates the channel. In the processing of standard color images, $\sigma \in \{r, g, b\}$. We assume that $h^\sigma = h$ for all σ. This approximation holds for cameras with lenses of reasonable quality.

Consider first the single-channel case. A variational approach to the recovery of u from z (given h) is based on the minimization of a cost functional that includes fidelity and regularization terms. The fidelity term quantifies the discrepancy between the observed image z and the blurred version $h * u$ of the recovered image. The regularization term is necessary since the image deconvolution problem is ill-posed.

The fidelity terms in variational image deblurring methods are commonly designed for Gaussian noise and are thus based on the L^2 norm

$$\int (h * u - z)^2 \, dA \, .$$

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 49–60, 2005.

To better deal with outliers and impulsive noise in image deconvolution, fidelity measurement using the L^1 norm

$$\int |h * u - z| \, dA$$

can be considered. In practice, the modified L^1 norm

$$\int \sqrt{(h * u - z)^2 + \eta} \, dA$$

is an approximation that offers numerical advantages ($0 < \eta \ll 1$).

Regularization in variational image deblurring can be accomplished using various stabilizers, such as Tikhonov (L^2) [1], Total Variation (L^1) [2,3,4] and ϕ-formalism [5,6]. Recently, elements of the Mumford-Shah functional have been used for regularization in the image deblurring problem [7].

Deblurring of color images in the variational framework has received surprisingly little attention. Blomgren and Chan [8] extended the Total Variation norm to vector-valued images in the context of image denoising. Barash [9] restored color images by combining an L^2 fidelity term with Perona-Malik regularization [10], but processed each channel separately. Two recent studies on color image deblurring were presented by Welk *et al* [6] and Kaftory *et al* [11]. The former study used L^2 fidelity term and a coupled multi-channel extension of the Perona-Malik regularizer. The latter presented a color image deblurring method employing an L^2 fidelity term and Beltrami flow regularization. Other than in the variational framework, multispectral Wiener-based restoration was suggested in [12]; see also [13,14].

This research concentrates on the deblurring of multi-channel images contaminated by impulsive noise. Addressing the problem in the variational setting, we extend the L^1 fidelity measure to multi-channel images. We study several regularization terms, and propose the generalization of two efficient stabilizers to the multi-channel case. The novel cost functionals, algorithms and theoretical discussions are supported by comparative experiments. Successful color image deblurring at high levels of impulse noise is demonstrated.

2 Cost Functionals

Image deblurring is an ill-posed inverse problem, that has to be regularized. In the variational framework, the recovered image is the minimizer of a cost functional. In the multi-channel case, the functional is of the general form

$$\mathcal{F} = \int_{\Omega} \sum_{\sigma} \Phi(h * u^{\sigma} - z^{\sigma}) \, dA + \mathcal{J}(u),$$

where $\Phi(\cdot)$ is a potential function and $\mathcal{J}(u)$ is a regularization operator that depends on all the channels. Ω is the image domain and $u : \Omega \to [0,1]^3$. In the case of Gaussian noise, a quadratic form of the data-fidelity is appropriate,

$$\Phi(h * u^{\sigma} - z^{\sigma}) = (h * u^{\sigma} - z^{\sigma})^2.$$

The quadratic form is inadequate for impulsive noise. To effectively suppress outliers, a (modified) robust L^1 norm can be used [4,3,15]:

$$\Phi(h * u^\sigma - z^\sigma) = \sqrt{(h * u^\sigma - z^\sigma)^2 + \eta},$$

where $\eta \ll 1$ is a positive constant.

This study focuses on color image deblurring with impulsive noise. We therefore base the fidelity term, in all the methods we consider, on the multi-channel modified L^1 norm. The methods differ in the regularization used; we present and discuss several possibilities, and evaluate their resulting restoration performance.

Total Variation (TV) regularization, first introduced by Rudin *et al* [2], has been widely used in image processing. A straightforward but naive extension to color images is via channel-by-channel TV regularization. For each channel,

$$\mathcal{J}^{TV}(u^\sigma) = \beta \int_\Omega |\nabla u^\sigma| \, dA.$$

Channel-by-channel image deblurring, with multiple decoupled functionals, is simple but may lead to artifacts. Specifically, since the channels are decoupled, color edges in different channels may not coincide, resulting in thin false stripes [16]. Channel coupling in TV regularization (Color-TV) was presented by Blomgren and Chan [8].

An alternative approach is the Beltrami flow introduced by Sochen *et al* [17]. Its superiority with respect to Color-TV regularization has been shown by Tschumperle [18]. In the Beltrami framework, a color image (u^r, u^g, u^b) is regarded as a two-dimensional surface embedded in $\mathbb{R}^5$-space spanned by (x, y, u^r, u^g, u^b). The area of this surface is given by

$$\int_\Omega \sqrt{\det G} \, dA,$$

where the metric G takes the form

$$G = \begin{pmatrix} 1 + \gamma^2 \sum_\sigma (u_x^\sigma)^2 & \gamma^2 \sum_\sigma u_x^\sigma u_y^\sigma \\ \gamma^2 \sum_\sigma u_x^\sigma u_y^\sigma & 1 + \gamma^2 \sum_\sigma (u_y^\sigma)^2 \end{pmatrix}.$$

This area is a measure of image smoothness, and has the important advantage of gradient alignment between channels. The Beltrami functional is given by

$$\mathcal{J}^{BEL} = \alpha \int_\Omega \sqrt{\det(G)} \, dA.$$

The advantages of this regularizer are more obvious if we rewrite it as

$$\int_\Omega \sqrt{1 + \gamma^2 \sum_\sigma (|\nabla u^\sigma|^2) + \frac{1}{2}\gamma^4 \sum_{\sigma_1, \sigma_2} |\nabla u^{\sigma_1} \times \nabla u^{\sigma_2}|^2} dA,$$

where $\times$ denotes cross-product. The cross-product term enforces channel alignment, making the gradients (u_x^σ, u_y^σ), $\sigma \in \{r, g, b\}$ parallel to each other, and

producing crisp color edges [19]. In the late stages of minimization, when the cross-product term is nearly zero, J^{BEL} approaches the robust Total Variation norm.

Reflecting the preference for piecewise smooth images, parts of the Mumford-Shah segmentation functional [20] can be used for regularization in image restoration as well [7]. In this stabilizer, the energy assigned to a gray level image $u : \Omega \to [0, 1]$ with an edge set $K \subset \Omega$ is

$$\mathcal{J}^{MS}(u, K) = \beta \int_{\Omega \setminus K} |\nabla u|^2 dA + \alpha \int_K d\mathcal{H}^1.$$

The first term forces the smoothness of u everywhere except on the discontinuity set K. The second term minimizes the one-dimensional Hausdorff measure (length) of the discontinuity set. Using the Γ-convergence framework, Ambrosio and Tortorelli [21] approximated this irregular functional by a sequence of regular functionals

$$\mathcal{J}_\epsilon^{MS}(u, v) = \beta \int_\Omega v^2 |\nabla u|^2 \, dA + \alpha \int_\Omega \left(\epsilon |\nabla v|^2 + \frac{(v-1)^2}{4\epsilon} \right) \, dA \, . \tag{1}$$

The auxiliary function $v(x)$ represents the edges. The minimizers of $\mathcal{J}_\epsilon^{MS}$ approach the minimizer of $\mathcal{J}^{MS}$ as $\epsilon \to 0$. In the color version of this functional, suggested by Brook *et al* [16],

$$|\nabla u| = \sqrt{\sum_\sigma \left[(u_x^\sigma)^2 + (u_y^\sigma)^2 \right]} \, . \tag{2}$$

This term is referred to as the Frobenius norm of ∇u. Note that in this regularizer the edge map v is common for the three channels.

A modified version of the Mumford-Shah functional, in its Γ-convergence approximation, was suggested by Shah [22] for gray-level images:

$$\mathcal{J}_\epsilon^{MSTV}(u, v) = \beta \int_\Omega v^2 |\nabla u| \, dA + \alpha \int_\Omega \left(\epsilon |\nabla v|^2 + \frac{(v-1)^2}{4\epsilon} \right) \, dA. \tag{3}$$

In this version the L^2 norm of ∇u was replaced by the L^1 norm in the first term. Alicandro *et al* [23] proved the Γ-convergence of this functional to

$$\mathcal{J}^{MSTV} = \beta \int_{\Omega \setminus K} |\nabla u| \, dA + \alpha \int_K \frac{|u^+ - u^-|}{1 + |u^+ - u^-|} d\mathcal{H}^1 \; + \; |D^c u|(\Omega),$$

where u^+ and u^- denote the image values on two sides of the edge set K, $\mathcal{H}^1$ is the one-dimensional Hausdorff measure and $D^c u$ is the Cantor part of the distributional derivative Du. This functional was generalized by Brook *et al* [16] for color images where the Frobenius norm (2) was used in this case as well.

In this research, we consider color image deblurring functionals with a multichannel L^1 fidelity term, and one of the regularization terms $\mathcal{J}^{BEL}$, $\mathcal{J}^{MS}$, and $\mathcal{J}^{MSTV}$. As a baseline, the channel-by-channel $\mathcal{J}^{TV}$ term will also be discussed.

3 Minimization

Minimization of the cost functionals is carried out using the Euler-Lagrange (E-L) equations with homogeneous Neumann boundary conditions $\partial u^\sigma/\partial N = 0$, $\partial v/\partial N = 0$, where N is the normal to the image boundary. We present the E-L equations of the four functionals that we consider. For channel-by-channel TV regularization, the E-L equation is

$$\frac{\delta \mathcal{F}^{TV}}{\delta u^\sigma} = \frac{h * u^\sigma - z^\sigma}{\sqrt{(h * u^\sigma - z^\sigma)^2 + \eta}} * h(-x,-y) - 2\beta \, \nabla \cdot \left(\frac{\nabla u^\sigma}{|\nabla u^\sigma|} \right) \; = \; 0 \qquad (4)$$

The Beltrami E-L equation takes the form

$$\frac{\delta \mathcal{F}^{BEL}}{\delta u^\sigma} = \frac{h * u^\sigma - z^\sigma}{\sqrt{(h * u^\sigma - z^\sigma)^2 + \eta}} * h(-x,-y) - \alpha \nabla \cdot \left(\sqrt{\det(G)} G^{-1} \nabla u^\sigma \right) = 0 \quad (5)$$

The parameter α can be made adaptive [24]. Here, α is replaced by $\alpha/\sqrt{\det(G)}$ in order to convert the regularizer to the Laplace-Beltrami geometric operator [11].

The objective functionals with the MS and MSTV regularization terms (1,3) depend on the recovered image u and on the edge map v. With MS regularization (1), the E-L equations are

$$\frac{\delta \mathcal{F}_\epsilon^{MS}}{\delta u^\sigma} = \frac{h * u^\sigma - z^\sigma}{\sqrt{(h * u^\sigma - z^\sigma)^2 + \eta}} * h(-x,-y) - 2\beta \nabla \cdot (v^2 \nabla u^\sigma) \; = \; 0 \qquad (6)$$

$$\frac{\delta \mathcal{F}_\epsilon^{MS}}{\delta v} \; = \; 2\beta v |\nabla u|^2 \; + \; \alpha \left(\frac{v-1}{2\epsilon} \right) \; - \; 2\epsilon \, \alpha \nabla^2 v \; = \; 0 \qquad (7)$$

For MSTV regularization (3), the L^1 norm $|\nabla u|$ is replaced by the modified L^1 norm $\sqrt{\gamma + |\nabla u|^2}$. Thus,

$$\frac{\delta \mathcal{F}_\epsilon^{MSTV}}{\delta u^\sigma} = \frac{h * u^\sigma - z^\sigma}{\sqrt{(h * u^\sigma - z^\sigma)^2 + \eta}} * h(-x,-y) - 2\beta \nabla \cdot \left(\frac{v^2 \nabla u^\sigma}{\sqrt{\gamma + |\nabla u|^2}} \right) = 0 \quad (8)$$

$$\frac{\delta \mathcal{F}_\epsilon^{MSTV}}{\delta v} = 2\beta v \sqrt{\gamma + |\nabla u|^2} + \alpha \left(\frac{v-1}{2\epsilon} \right) \; - \; 2\epsilon \, \alpha \nabla^2 v \; = \; 0 \qquad (9)$$

It can be easily seen that Eqs. (4,5,6,8) are non-linear integro-differential equations. Following Vogel and Oman [25], linearization of the E-L equations for the color channels u is performed using the fixed point iteration scheme, where their denominators are lagged by one iteration with respect to the numerators. The linearized functions are then solved by the conjugate gradients method. Eqs. (7,9) are linear with respect to v and are solved using the Minimal Residual algorithm. Note that with MS and MSTV regularization, there are four equations to solve. The minimization procedure alternates between u^σ, $\sigma \in \{r, g, b\}$ and v until convergence.

The discretization of $|\nabla u|$ in Eqs. (7) and (9) was carried out using the central difference scheme. Terms of the form $\nabla \cdot (C \; \nabla u^\sigma)$ and $\nabla^2 v$ were discretized using forward difference for the gradient and backward difference for the divergence.

4 Robust Statistics Interpretation

In this section we provide a robust-statistics interpretation of the MS (1) and MSTV (3) regularizers. Consider half-quadratic regularization [26]. In this approach, the regularizer is a non-decreasing potential function $\rho(t)$, where in the context of image restoration $t = |\nabla u|$. A central element in half-quadratic regularization is the representation of ρ as an infimum of a quadratic function with an auxiliary variable b. Explicitly, if $\rho(\sqrt{t})$ is concave and non-decreasing, we can write

$$\rho(t) = \inf_b (bt^2 + \Psi(b))$$

such that $\Psi(b)$ is convex and decreasing. This representation is quadratic with respect to t when b is fixed, and therefore leads to easier optimization. In the case of edge-preserving image restoration, the auxiliary function b represents the edges. For example, the Geman and McClure [27] potential function corresponds to the half-quadratic form

$$\frac{|\nabla u|^2}{1 + |\nabla u|^2/\delta} = b|\nabla u|^2 + \delta(\sqrt{b} - 1)^2,$$

where δ is a positive constant. Teboul *et al* [28] noticed that in the presence of noise, image restoration requires well-behaved edges. Therefore an edge regularization term $\phi_b(b)$ was added:

$$\mathcal{J} = \lambda_1 \int_\Omega \left[b|\nabla u|^2 + \Psi(b) \right] dA + \lambda_2 \int_\Omega \phi_b(b) dA.$$

In the case that $\Psi(b) = \delta(\sqrt{b} - 1)^2$ and $\phi_b = |\nabla(\sqrt{b})|^2$,

$$\mathcal{J} = \lambda_1 \int_\Omega \left[b|\nabla u|^2 + \delta(\sqrt{b} - 1)^2 \right] dA + \lambda_2 \int_\Omega \left| \nabla(\sqrt{b}) \right|^2 dA \ . \tag{10}$$

Substituting $b = v^2$, $\lambda_1 = \beta$, $\delta = \alpha/4\epsilon\beta$ and $\lambda_2 = \alpha\epsilon$ yields equivalence between Eq. (10) and the MS regularizer (1) in its Γ-convergence approximation. This regularizer is therefore the robust Geman-McClure ρ function with an additional spatial edge organization constraint, where

$$\rho^{MS} = \frac{|\nabla u|^2}{1 + |\nabla u|^2/\delta} \ . \tag{11}$$

In the same manner, the robust ρ function that corresponds to the MSTV regularizer (3) with the modified L^1 norm is ($\delta = \alpha/4\epsilon\beta$)

$$\rho^{MSTV} = \frac{\sqrt{\gamma + |\nabla u|^2}}{1 + \sqrt{\gamma + |\nabla u|^2/\delta}} \ , \tag{12}$$

with $\delta = \alpha/4\epsilon\beta$.

Fig. 1. Color image deblurring in the presence of impulsive noise, using a multi-channel modified L^1 fidelity term, with four different regularization methods. *Top-left:* Blurred image. *Top-middle:* Blurred image contaminated by 10% salt and pepper noise. *Top-right:* Deblurring with channel-by-channel TV regularization. *Bottom-left:* Beltrami flow. *Bottom-middle:* Color Mumford-Shah (MS). *Bottom-right:* Color TV Mumford-Shah (MSTV).

5 Results

The original 256×256 *Lena* image (not shown) was blurred by a pill-box kernel of radius 3 (7×7 kernel), see Fig. 1 top-left. Each color channel was then contaminated by salt-and-pepper noise of 10% density (top-middle). Image restoration results, with a fidelity term based on the color version of the modified L^1 norm and the various regularization terms are compared in Fig. 1. The modified L^1 norm parameter was set to $\eta = 10^{-4}$ in all experiments. The top-right image is the outcome of the channel-by-channel TV regularization (Eq. 4) with $\beta = 0.1$. Recovery with the Beltrami stabilizer (Eq. 5) is shown bottom left. In this case, the parameters were set to $\alpha = 0.8$ and $\gamma = 0.1$. The images recovered using the MS (Eqs. 6 and 7) and MSTV (Eqs. 8 and 9) regularizers are shown bottom-middle and right respectively. For both methods $\alpha = 0.5$ and $\epsilon = 0.1$, while $\beta = 0.7$ for MS and $\beta = 0.5$ for MSTV.

As can be observed, channel-by-channel TV regularization is inferior with respect to the other three methods. This is not surprising, since channel-by-channel TV regularization does not exploit the inter-channel redundancy. The images recovered using the Beltrami, MS and MSTV stabilizers are of high quality, and are quite similar to each other at this moderate noise level. Table 1

Table 1. PSNR values using several regularization methods

Image	10% noise [dB]	30% noise [dB]
Observed	14.96	10.35
TV	21.14	14.8
Beltrami	23.57	21.93
MS	23.87	23.28
MSTV	23.76	23.42

(the 10% noise column) provides quantitative comparative evaluation of these deblurring results using PSNR values:

$$\mathrm{PSNR}(I) = 20 \log \sqrt{\frac{3MN}{\sum_\sigma \sum_{i,j} (I_{ij}^\sigma - \hat{I}_{ij}^\sigma)^2}}.$$

Here M and N are the image dimensions, I and $\hat{I}$ are the original and recovered images respectively.

The differences between the Beltrami, MS and MSTV regularizers can be better observed at a higher noise level, see Fig. 2. The blurred *Lena* image, with 30% noise is shown top-left. Shown top-right is the outcome of Beltrami flow regularization with $\alpha = 1.2$ and $\gamma = 0.1$. The images recovered using the MS and MSTV stabilizers are shown bottom-left and bottom-right respectively. In these cases the parameters were $\alpha = 0.5$ and $\epsilon = 0.1$, with $\beta = 2.2$ for MS and $\beta = 1.5$ for MSTV.

The images recovered with MS and MSTV regularization are quite similar and visibly cleaner than the image obtained using the Beltrami flow stabilizer. Fig. 3 shows magnifications of the images obtained with Beltrami (left) and MSTV (right) regularization. The quantitative PSNR results are shown in the right column of Table 1, and are compatible with the perceived visual quality.

The superiority of the MS and MSTV regularizers with respect to the Beltrami flow in dealing with impulse noise can be well understood via the robust statistics interpretation presented in section 4. While all three stabilizers support inter-channel coupling, only the MS and MSTV methods impose an edge organization constraint. Edge organization is at the heart of the difference between an impulse noise point and a point that is part of a structured image edge.

The robust statistics perspective can be also used to compare the MS and MSTV regularizers. Fig. 4 presents the robust functions corresponding to the MS regularizer (solid, red) and the MSTV stabilizer at two values of γ. The dotted (green) curve corresponds to $\gamma = 0.1$ that we used in our experiments, and the dashed (blue) curve is for $\gamma = 10^{-10}$. The solid and dotted curves nearly coincide near the origin, but elsewhere the MSTV curve is lower than the solid curve. This implies better edge preservation with the MSTV regularizer. As γ is decreased, the MSTV robust function becomes narrower, driving the recovered

Fig. 2. Color image restoration using several regularization methods. *Top-left:* Blurred image with 30% impulse noise. *Top-right:* Beltrami flow regularization. *Bottom-left:* Color Mumford-Shah (MS). *Bottom-right:* Color TV Mumford-Shah (MSTV).

image towards the cartoon (piecewise constant) limit. Additional insight about the importance of color-channel coupling in image restoration can be gained by comparing restoration using multi-channel MS regularization with channel-by-channel MS restoration, i.e., by applying the method of [15] to each channel separately. As seen in Fig. 5, the multi-channel approach yields excellent results even at a high (40%) level of impulse noise (left), where channel-by-channel processing is clearly inadequate.

Finally, a useful byproduct of the MS and MSTV regularization methods is the auxiliary function v, that can serve as an edge map. Fig 6 (left) is a blurred and noisy *Lena* image; Fig 6 (right) is the v function (edge map) obtained with the MS regularizer.

6 Conclusion

We presented variational methods for color image deblurring with impulse noise. The methods share an L^1 color fidelity term, but differ in the regularization used. Our results verify the importance of channel coupling in the regularization terms. Furthermore, with impulsive noise, regularization methods based on the Mumford-Shah functional and its variants are superior to other methods,

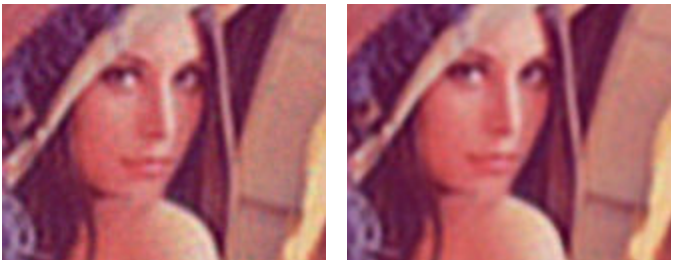


Fig. 3. Magnification of the images shown in the right column of Fig. 2. *Left:* Beltrami flow regularization. *Right:* MSTV regularization.

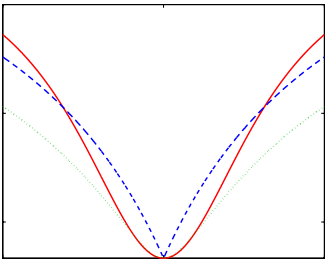


Fig. 4. Robust functions. *Red solid:* ρ^{MS}. *Blue dashed:* ρ^{MSTV} with $\gamma = 10^{-10}$. *Green dotted:* ρ^{MSTV} with $\gamma = 0.1$.

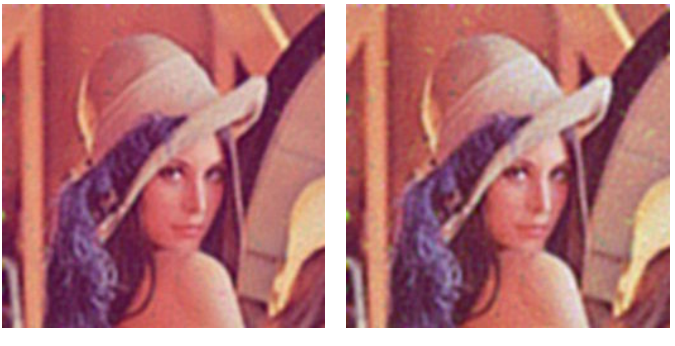


Fig. 5. Multi-channel vs. chanel-by-channel deblurring at a high level of impulse noise (40%). *Left:* Multi-channel MS regularization. *Right:* Channel-by-channel MS processing.

including the Beltrami flow. This is seen in the experiments, and theoretically explained from the robust statistics point of view. Promising deconvolution results are obtained even at high levels of impulsive noise.

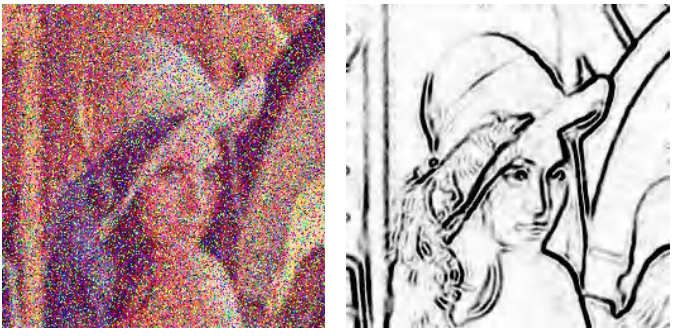


Fig. 6. *Left:* Blurred image contaminated by 30% impulsive noise. *Right:* Edge map obtained as a by-product of restoration using MS regularization.

Acknowledgment

This research was supported by MUSCLE: Multimedia Understanding through Semantics, Computation and Learning, a European Network of Excellence funded by the EC 6th Framework IST Programme. It was supported also by the Israel Science Foundation.

References

1. Tikhonov, A.N., Arsenin, V.I.: Solutions of ill-posed problems. Winston (1977)
2. Rudin, L., Osher, S., Fetami, E.: Non linear total variatrion based noise removal algorithms. Physica D **60** (1992) 259–268
3. Nikolova, M.: A variational approach to remove outliers and impulse noise. J. Math. Imaging Vision **20** (2004) 99–120
4. Nikolova, M.: Minimizers of cost-functions involving nonsmooth data-fidelity terms: Application to the processing of outliers. SIAM J. Numer. Anal. **40** (2002) 965–994
5. Deriche, R., Faugeras, O.: Les EDP en traitement des images et vision par ordinateur. Traitement du Signal **13** (1996)
6. Welk, M., Theis, D., Brox, T., Weickert, J.: PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. In: Proceeding of 5th International Conference on Scale Space and PDE Methods in Computer Vision. Volume 3439 of LNCS. (2005) 585–597
7. Bar, L., Sochen, N., Kiryati, N.: Variational pairing of image segmentation and blind restoration. In Pajdla, T., Matas, J., eds.: Proceedings of 8th European Conference on Computer Vision. Volume 3022 of LNCS. (2004) 166–177
8. Blomgren, P., Chan, T.F.: Color TV: total variation methods for restoration of vector-valued images. IEEE Trans. Image Processing **7** (1998) 304–309
9. Barash, D.: One-step deblurring and denoising color images using partial differential equations. Technical Report HPL-2000-102R1, HP Laboratories (2000)
10. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12** (1990) 629–639
11. Kaftory, R., Sochen, N., Zeevi, Y.Y.: Color image denoising and blind deconvolution using the Beltrami operator. In: Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis. Volume 1. (2003) 1–4

12. Hunt, B., Kübler, O.: Karhunun-Loeve multispectral image restoration, part I: Theory. IEEE Trans. Acoustics, Speech and Signal Proc. **32** (1984) 592–600
13. Banham, M., Katsaggelos, A.: Digital image restoration. IEEE Signal Processing Mag. **14** (1997) 24–41
14. Molina, R., Mateos, J., Katsaggelos, A.K., Vega, M.: Bayesian multichannel image restoration using compound Gauss-Markov random fields. IEEE Trans. Image Proc. **12** (2003) 1642–1654
15. Bar, L., Sochen, N., Kiryati, N.: Image deblurring in the presence of salt-and-pepper noise. In: Proceeding of 5th International Conference on Scale Space and PDE Methods in Computer Vision. Volume 3439 of LNCS. (2005) 107–118
16. Brook, A., Kimmel, R., Sochen, N.: Variational restoration and edge detection for color images. J. Math. Imaging Vision **18** (2003) 247–268
17. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Trans. Image Proc. **7** (1998) 310–318
18. Tschumperlé, D.: PDE's based regularization of multivalued images and applications. PhD thesis, University of Nice–Sophia Antipolis (2002)
19. Kimmel, R., Malladi, R., Sochen, N.: Images as embedded maps and minimal surfaces: Movies, color, texture, and volumetric medical images. Int. J. Computer Vision **39** (2000) 111–129
20. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. **42** (1989) 577–685
21. Ambrosio, L., Tortorelli, V.M.: Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. Comm. Pure Appl. Math. **43** (1990) 999–1036
22. Shah, J.: A common framework for curve evolution, segmentation and anisotropic diffusion. In: IEEE Conference on Computer Vision and Pattern Recognition. (1996) 136–142
23. Alicandro, R., Braides, A., Shah, J.: Free-discontinuity problems via functionals involving the L^1-norm of the gradient and their approximation. Interfaces and Free Boundaries **1** (1999) 17–37
24. Strong, D., Chan, T.: Edge-preserving and scale dependent properties of total variation regularization. CAM Report 00–38, UCLA Math department (2000)
25. Vogel, C.R., Oman, M.E.: Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans. Image Proc. **7** (1998) 813–824
26. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Trans. Image Proc. **6** (1997) 298–311
27. Geman, S., McClure, D.E.: Bayesian image analysis: An application to single photon emission tomography. Proc. Amer. Statist. Assoc. Statistical Computing Section (1985) 12–18
28. Teboul, S., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Variational approach for edge-preserving regularization using coupled PDE's. IEEE Trans. Image Proc. **7** (1998) 387–397

Using an Oriented PDE to Repair Image Textures

Yan Niu[1] and Tim Poston[2]

[1] College of Computer Science and Technology,
JiLin University, QianWei Rd. 10, ChangChun 130012, P.R. China
niuyan@email.jlu.edu.cn
[2] National Institute of Advanced Studies,
Indian Institute of Science Campus, Bangalore 560012, India

Abstract. PDE-based image inpainting efficiently recovers structured features. We expand this to textures. We adjust the coordinates to proper directions, and embed in anisotropy terms the brightness correlation between pixels adjoining on the new grid. A simple elliptic equation then repairs both oriented textures and edges by one uniform, automated algorithm. Extensive experimental results on a variety of standard natural images show the technique's generality and stability.

1 Introduction

This paper offers a new partial differential equation (PDE) repair for loss or occlusion in digital images.

Broken edge repair is well solved by anisotropic PDE's, beginning with the third-order diffusion equation by Bertalmio *et al.* [1]; total variation (TV) minimization by Chan and Shen [2]; and the level-line method by Masnou and Morel [3]. Their novelty is that these PDE's utilize the image gradient vector as an edge detector, allowing the simultaneous inpainting of smoothness and sharp edges.

However, any edge detector is sensitive to textures. As a result, to use these PDE's for image inpainting, additional texture synthesis (*e.g.*, Efros-Leung's [4]) and feature discrimination are required. For example, Ref. [5] pre-classifies the lost blocks into edgy or textured type, and Ref. [6] extracts textures from structures. The results are satisfactory, but performance hinges on the efficiency of the feature discrimination. Efros-Leung's texture synthesis (a pixel-by-pixel matching and replicating method) performs well but is known as time-consuming. Acton, Mukherjee, Havlicek and Bovik [7] propose an impressive reaction-diffusion (RD) model for fingerprint completion, using a Gabor filter-based reaction term for texture recreation. The literature also holds a number of PDE methods for texture processing, such as the texture enhancement of Weickert and Scharr [8], texture extraction of Vese and Osher [9], and texture generation of Witkin and Kass [10]. Nevertheless, PDE inpainting has been limited to non-texture regions.

In this paper, a simple anisotropic elliptic PDE discretized along proper directions, with brightness correlation embedded in its anisotropy terms, repairs broken

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 61–72, 2005.

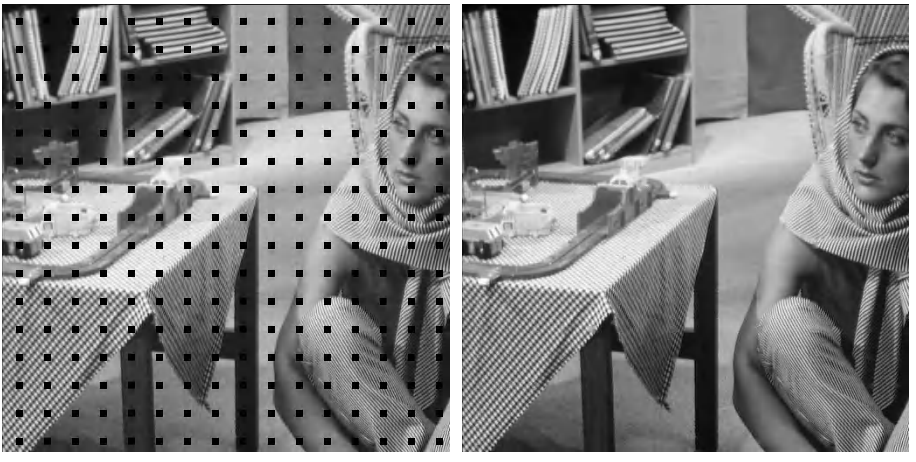


Fig. 1. An example of our image repair. Left: the 512×512 image with a decoding failure caused by a bit error in one ERC-SPIHT coded bit stream (out of 16). Right: the image recovered by our approach.

textures as well as edges. This recovers all content by a fully automated, uniform algorithm, with no discriminative preprocess of classification or decomposition.

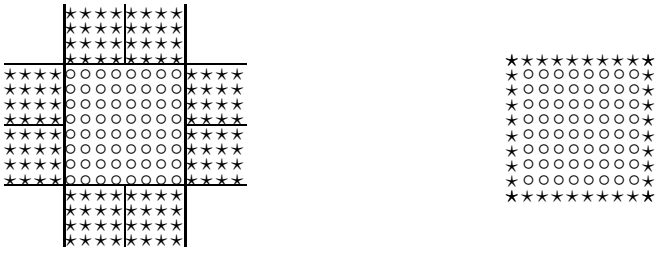
Traditionally, a continuous PDE model discretized along horizontal and vertical axes estimates a missing pixel by low pass filtering its nearest neighbors. We use a differently angled subgrid. A pixel in an oriented feature differs significantly from its next neighbors, but with near-shared values typically displaced by a repetition vector, depending on either the direction or the repetitiveness of a feature (two dimensions for texture perception [11]). More favorably, pixels on edges also conform to such repetition regularity. This observation led us to compute the repetition vector with a simple optical flow algorithm from the computer vision literature, building our model on this direction and its normal.

The PDE's anisotropy term is another crucial factor. In our new coordinate grid, a lost pixel and its repetition vector direction neighbors have similar intensity; normal to the direction, values tend to change gradually. We treat separately the terms for anisotropy along each axis, consequently, the coefficient field of our model takes the form of a diagonal matrix. As we estimate their values from the valid pixels, our model is an elliptic PDE rather than a diffusion equation. This saves the computational cost on updating the diffusion coefficients in each diffusion step, and gives more accurate modelling of brightness contrast.

2 New PDE Recovery Approach

2.1 Repetition Vector

Image recovery infers a lost region from regularity in ambient valid regions: here regularity in repetitiveness of oriented features, either edges or textures. Denote



(a) (b)

Fig. 2. (a) an example lost block and the reference blocks used for RV search, (b) the one-pixel-wide strip for RV selection. Circles stand for lost pixels, stars for valid ones.

the original value at pixel (i, j) by $I(i, j)$, the $N \times N$ block of missing pixels by B, surrounded by a set S of valid pixels. An integer vector $\rho = (\rho_x, \rho_y) \in \mathrm{Z}^2$ is a *repetition vector* (RV) of $\bar{B} = B \cup S$, if for any $(i, j) \in \bar{B}$,

$$I\,(i,j) \approx I\,(i + \rho_x, j + \rho_y)\,, \tag{1}$$

where we interpret "$\approx$" below. The symmetry of, e.g., a chessboard may give multiple RVs, but commonly (up to sign) a unique ρ best satisfies (1). For edges, ρ is fixed by edge direction or isophotes; for textures, ρ depends on the direction or repetitiveness. Evidently, we must seek the ρ for $\bar{B}$ in S. We narrow the range to R, the union of the eight $N/2 \times N/2$ *references blocks* that abut B (Fig. 2(a)). Finding ρ is analogous to finding optical flow in an image sequence: both essentially matching problems. We borrow from the computer vision literature the classical optical flow searching method of corner extraction and matching [12].

We extract from each reference block as corners the pixels with intensities near local maximum and minimum, if the maximum and minimum differ larger than a threshold (*e.g.*, 30 apart in 8-bit monochrome images); and sequentially match these corners. Specifically, for two corners (i_1, j_1), (i_2, j_2) in R with values $I(i, j)$ close (*e.g.*, less than 10 apart), fix a matching 3×3 window centered at each, and a set V of valid neighbors $V = \{(k, h) \in [-1, 1] \times [-1, 1] | (i_1 + k, j_1 + h), (i_2 + k, j_2 + h) \in S\}$. $(i_2 - i_1, j_2 - j_1)$ is determined to be a candidate for ρ, if the mean absolute difference (MAD)

$$MAD = \frac{1}{|V|}\{\sum_{(k,h)\in V} |I(i_1 + k, j_1 + h) - I(i_2 + k, j_2 + h)|\}.$$

is below a threshold (*e.g.*, 15). From all candidates, one of two tests (AAD or SSE) picks as optimal RV the vector that minimizes its cost function on the surrounding one-pixel-wide strip τ of neighbors (Fig. 2(b)).

Average of Absolute Difference (AAD) Selection: For a candidate $c = (c_x, c_y)$, an arbitrary pixel (u, v) in τ has two correspondents $(u + c_x, v + c_y)$ and $(u - c_x, v - c_y)$ with respect to c. As (u, v) is at the boundary of a reference block, if $|c_x|$ and $|c_y|$ are not so large as to reach the next missing region, one correspondent (u', v') has a correctly decoded value. Select as ρ the c that minimizes

$$\rho = \arg\min_{(c_x,c_y)} \frac{1}{|\tau|} \sum_{(u,v)\in\tau} \text{abs}(I(u,v) - I(u',v')).$$

Sum of Squared Error (SSE) Selection: For each candidate RV, test on τ the recovery algorithm (13) that will be introduced in Section 2.3, which at each pixel $(u,v) \in \tau$ gives the recovered value $I^*_{c_x,c_y}(u,v)$. Select as ρ the c that minimizes

$$\rho = \arg\min_{(c_x,c_y)} \sum_{(u,v)\in\tau} (I(u,v) - I^*_{c_x,c_y}(u,v))^2.$$

We compare these methods in Section 3. It is possible that the search finds no match, for instance, if the missing region and its neighborhood are in a smooth area, then no corner is extracted. In this case, we assign $[0,1]$ as the RV. Later we will find, our system based on $\rho = [0,1]$ degenerates to the traditional one.

Normal to the RV $\rho = \pm(\rho_x, \rho_y)$ are $\rho^\perp = \pm(-\rho_y, \rho_x)$, two vectors of the same size. From these four vectors, $\overrightarrow{V_\zeta} = (n,m)$ with $m,n \geq 0$ and its normal $\overrightarrow{V_\eta} = (-m,n)$ in the second quadrant are our unit vectors for ζ and η axes respectively, giving the frame for our PDE model. Fig. 3(a) shows a lost block in the standard image *Barbara*, for which the output ρ of the RV search algorithm is (-3,-2). Fig. 3(b) illustrates the oriented basis formed by $\overrightarrow{V_\zeta} = (3,2)$ and $\overrightarrow{V_\eta} = (-2,3)$.

The new coordinate differs fundamentally from the Gauge coordinate, although both are dictated by the feature orientation. The Gauge coordinate directions $\omega = \nabla I/\|\nabla I\|$ and $\vartheta = \omega^\perp$, the maximum intensity change direction and its normal, still depend on the estimation of horizontal and vertical partial differentials $\partial I/\partial x$ and $\partial I/\partial y$. As the oriented texture pixel is singular either horizontally or vertically—that is why texture can be perceived —a traditional estimation of $\partial I/\partial x$ and $\partial I/\partial y$ by the forward (backward) differences could be problematic. In our system, the integer repetition vector $\overrightarrow{V_\zeta}$ is found as in optical flow searching. The role of $\pm\overrightarrow{V_\zeta}$ is to point out directly the pixels that are supposed to have the similar intensity value, without estimating the derivative (or difference) along any direction.

2.2 Oriented Anisotropic Brightness Equation

The PDE's anisotropy terms control the brightness contrast of adjacent pixels. They are classified into three categories [14], which we illustrate with the following examples.

- $\partial I/\partial t = \text{div}(g(\|\nabla I\|)\nabla I)$ (*e.g*, [2], [7], [13]). The coefficient field $g()$ is a scalar decreasing function of $\|\nabla I\|$.
- $\partial I/\partial t = \text{div}(D \cdot \nabla I)$ (*e.g.*, [8]). The *diffusion tensor* matrix D is symmetric. The three entries of D is determined by the structure tensor matrix.
- $\partial I/\partial t = A \cdot H$ (*e.g.*, [10]). H is the *Hessian* matrix, and A is a 2×2 symmetric matrix determined by the principle direction.

Our elliptic equation model, the *oriented anisotropic brightness equation* (OABE), is in-between, with the anisotropy terms $p(\zeta,\eta) = 1/|\frac{\partial I}{\partial \zeta}|$, $q(\zeta,\eta) = 1/|\frac{\partial I}{\partial \eta}|$,

$$\begin{cases} \frac{\partial}{\partial \zeta}(p(\zeta,\eta)\frac{\partial I}{\partial \zeta}) + \frac{\partial}{\partial \eta}(q(\zeta,\eta)\frac{\partial I}{\partial \eta}) = 0 & (\zeta,\eta) \in B, \\ I(\zeta,\eta) = I_0(\zeta,\eta) & (\zeta,\eta) \in S. \end{cases} \tag{2}$$

where I_0 indicates the values of valid pixels.

A conservative 5-point differencing scheme for (2) at pixel (i,j),

$$\begin{aligned} 0 = {} & p((i,j) - V_\zeta/2)(I((i,j) - V_\zeta) - I(i,j)) \\ & + p((i,j) + V_\zeta/2)(I((i,j) + V_\zeta) - I(i,j)) \\ & + q((i,j) - V_\eta/2)(I((i,j) - V_\eta) - I(i,j)) \\ & + q((i,j) + V_\eta/2)(I((i,j) + V_\eta) - I(i,j)). \end{aligned} \tag{3}$$

gives our estimation scheme. With the lost block of Fig. 3(a) as an example, Fig. 3(c) shows the pixels used to estimate the value at bottom-left, $I(240, 393)$, the true value of which is 167. Apparently, its valid neighbors in the new coordinate grid (with values 169, 175, and 174) are more suitable for the recovery than its horizontal and vertical neighbors (152 and 209).

Previous diffusion inpainting generally updates the anisotropy terms during the iteration process, using the intensity values generated in the previous iteration step. We estimate the terms from the valid pixels directly, which saves the computational cost of parameter fitting in each iteration step. Moreover, the brightness contrast is modelled more accurately from the valid pixels. We first derive the restricted $p((u,v) \pm V_\zeta/2)|S$ from the known $I(u,v)|S$. At $(u,v) \in S$, replace partials by central differences:

$$p((u,v) \pm V_\zeta/2) = \frac{1}{|I((u,v) \pm V_\zeta) - I(u,v)|}. \tag{4}$$

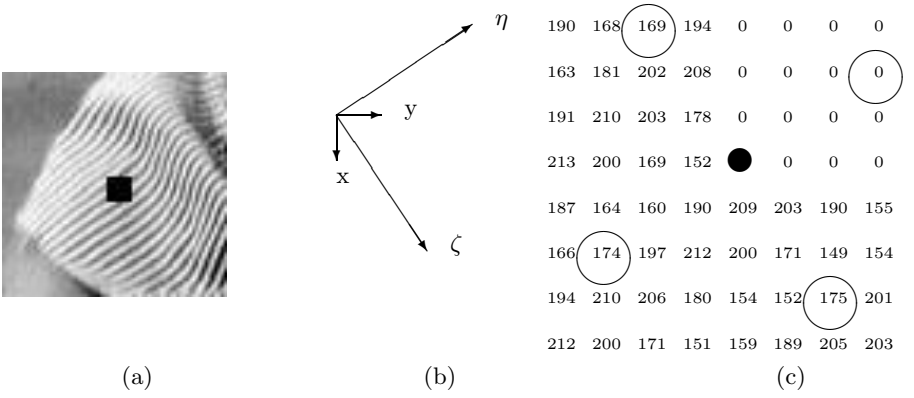
replacing by 1 any zero denominator(s). Next we linearly interpolate $p((i,j) \pm V_\zeta/2)|B$ from $p((i,j) \pm V_\zeta/2 + k_\eta V_\eta)$ and $p((i,j) \pm V_\zeta/2 - k_{-\eta} V_\eta)$, where

$$\begin{aligned} k_{-\eta} &= \min\{k > 0 | \{((i,j) \pm V_\zeta - kV_\eta), ((i,j) - kV_\eta))\} \subset S\} \\ k_\eta &= \min\{k > 0 | \{((i,j) \pm V_\zeta + kV_\eta), ((i,j) + kV_\eta))\} \subset S\}, \end{aligned}$$

by

$$\begin{aligned} p((i,j) \pm V_\zeta/2) = {} & \frac{k_{-\eta}}{k_\eta + k_{-\eta}} p((i,j) \pm V_\zeta/2 + k_\eta V_\eta) \\ & \frac{k_\eta}{k_\eta + k_{-\eta}} p((i,j) \pm V_\zeta/2 - k_{-\eta} V_\eta). \end{aligned} \tag{5}$$

similarly for $q()$, with exchange of V_ζ and V_η. Fig. 4 illustrates (5) for the bottom-left pixel of the lost block in Fig. 3, where $k_\eta = 3$ and $k_{-\eta} = 1$.



(a) (b) (c)

Fig. 3. (a) an example of an 8×8 lost block, (b) its new basis vectors fixed by the RV search algorithms, (c) pixels (in circles) set by the new basis, estimate the bottom-left corner (black dot) of the lost block

2.3 Solution of the OABE

With $p((i,j) \pm V_\zeta/2)|B$, $q((i,j) \pm V_\eta/2)|B$ given by (5), we solve the OABE. Vectorize the lost block entries $I(i,j)|B$ in raster order:

$$\begin{aligned} X &= (x_1, x_2, \cdots, x_{N\times N}) \\ &= (I(i_0,j_0), I(i_0,j_0+1), \cdots, I(i_0+N-1, j_0+N-1)), \end{aligned}$$

where (i_0, j_0) is the upper-left pixel of the lost block B. Clearly, if $I(i,j)$ corresponds to the lth entry x_l of X, then $I((i,j) \pm V_\zeta)$ and $I((i,j) \pm V_\eta)$ correspond to $x_{l\pm nN\pm m}$ and $x_{l\mp mN\pm n}$. Thus (3) can be written as

$$\begin{aligned} 0 = {} & [p((i,j) - V_\zeta/2) + p((i,j) + V_\zeta/2) + q((i,j) - V_\eta/2) + q((i,j) + V_\eta/2)]x_l \\ & -p((i,j) - V_\zeta/2)x_{l-nN-m} - p((i,j) + V_\zeta/2)x_{l+nN+m} \\ & -q((i,j) - V_\eta/2)x_{l+mN-n} - q((i,j) + V_\eta/2)x_{l-mN+n}. \end{aligned} \tag{6}$$

Setting

$$\begin{aligned} T(i,j) = {} & p((i,j) - V_\zeta/2) + p((i,j) + V_\zeta/2) \\ & +q((i,j) - V_\eta/2) + q((i,j) + V_\eta/2) \end{aligned} \tag{7}$$

$$\begin{aligned} \omega_{l,l-nN-m} &= \frac{p((i,j) - V_\zeta/2)}{T(i,j)} \quad \omega_{l,l+nN+m} = \frac{p((i,j) + V_\zeta/2)}{T(i,j)} \\ \omega_{l,l+mN-n} &= \frac{q((i,j) - V_\eta/2)}{T(i,j)} \quad \omega_{l,l-mN+n} = \frac{q((i,j) + V_\eta/2)}{T(i,j)} \end{aligned} \tag{8}$$

we have

$$\begin{aligned} & x_l - \omega_{l,l-nN-m}x_{l-nN-m} - \omega_{l,l+nN+m}x_{l+nN+m} \\ & - \omega_{l,l+mN-n}x_{l+mN-n} - \omega_{l,l-mN+n}x_{l-mN+n} = f_l = 0. \end{aligned} \tag{9}$$

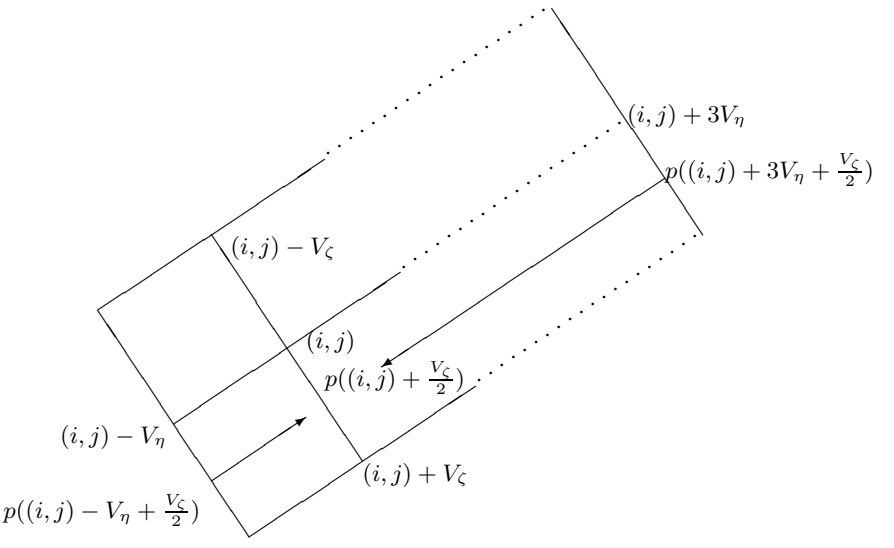


Fig. 4. A graphical illustration on interpolation equation (5), through the example of the bottom-left pixel in the lost block shown in Fig. 3

If any term among $x_{l\pm nN\pm m}$ and $x_{l\pm mN\mp n}$ is in S, shift it to the right of (9) and add to f_l. Eqs. (9) for each $l \in [1, N^2]$ form

$$WX^T = f, \tag{10}$$

a linear system with diagonal 1s, and lth row of W

$$(-\omega_{l,1}, -\omega_{l,2}, \cdots, -\omega_{l,l-1}, 1, -\omega_{l,l+1}, \cdots, -\omega_{l,N\times N}),$$

has at most five nonzero entries. Furthermore, equality (8) gives

$$\sum_{h\in[1,N^2]\setminus\{l\}} |-\omega_{l,h}| \le \omega_{l,l} = 1. \tag{11}$$

In practice, the non-0 and non-1 entries have finite precision. Discard the least significant digit of each, and the inequality

$$\sum_{h\in[1,N^2]\setminus\{l\}} |-\omega_{l,h}| < \omega_{l,l} \tag{12}$$

always holds: the matrix W is *strictly diagonally dominant*. This property implies the following theorem regarding the solution of (10).

1. W is non-singular. See Strang [15] (pp. 289-290) for a concise proof. Thus the solution of (10) exists and is unique.

2. The Guass-Seidel iteration over k,

$$x_l^{k+1} = \sum_{h=1}^{l-1} \omega_{l,h} x_h^{(k+1)} + \sum_{h=l+1}^{N^2} \omega_{l,h} x_h^{(k)} + f_l, \tag{13}$$

converges to the theoretical solution [16] (pp. 230-233).
3. The solution $I^*(i,j)|_{(i,j)\in B}$ satisfies

$$I^*(i,j) \geq 0 \quad (i,j) \in B \qquad \max_{(i,j)\in B} I^*(i,j) \leq \max_{(u,v)\in S} I(u,v).$$

This property can be verified straightforwardly, and guarantees a regenerated value in the reasonable range; *e.g.*, in 8-bit monochrome, it ensures values in [0,255].

3 Experimental Results

We evaluate this on standard 512×512, 8-bit monochrome raw images, with different loss patterns. In real-world use, with no access to original source images, the primary reconstruction evaluation criterion is visual quality: repaired defects should be as imperceptible as possible. To quantify generality and robustness of the OABE on different images, we follow tradition in previous work and compute, over the recreated pixels (since clean pixels are untouched during reconstruction), the mean square error (MSE) through peak-signal-to-noise ratio (PSNR)

$$\text{PSNR} = 10 \log_{10}^{\frac{255^2}{\text{MSE}}}. \tag{14}$$

with

$$\text{MSE} = \frac{1}{|\mathscr{B}| \cdot N^2} \sum_{B\in\mathscr{B}} \sum_{(i,j)\in B} \left(I_0(i,j) - I^*(i,j)\right)^2,$$

where B is a lost-recovered block of size $N \times N$, $\mathscr{B}$ is the collection of all these blocks, and I_0, I^* indicate the true and recovered values respectively. It should be of note that some publications (*e.g.*, [17]) compute the MSE over the entire image, which is higher than (14) by a constant set by the loss rate, given the same sum of squared errors. When comparing the OABE with previous methods, we shall specify the PSNR's by both schemes.

We first test the OABE on images coded by Error Resilient and Concealment-Set Partitioning in Hierarchical Trees (ERC-SPIHT), currently one of the most prevalent video coders. On the test images, we simulated 3-level, Haar-wavelet-based ERC-SPIHT, and the 16-codestream structure used in [18]. An early bit error in one bitstream over the binary symmetric channel incurs regular blank 8×8 squares. To see OABE's behavior in finest detail, assume the other 15 bit-streams are fully transmitted and decoded. Our setup is identical for every image. In solving the OABE, the bound on iterations in (13) is set to 200. On using SSE selection, the iteration bound is 80.

OABE performance is strongly influenced by RV choice. The ground-truth RV gives high PSNR and invisible concealment artifacts. In our study, after corner extraction and matching, it is generally in the candidate list, but on a few blocks the optimizer might select other candidates. On image *Barbara*, SSE is both objectively and subjectively superior to AAD; Fig. 1 shows its result, with a PSNR of 27.33dB. On the other test images, less textured than *Barbara*, SSE and AAD yield parallel results, and both suffice for visual and numerical quality. Moreover, AAD selection is faster. Fig. 5 shows the results by AAD selection.

We also test the OABE on recovering 16×16 blocks with high oscillation patterns from *Barbara*. Fig. 6 provides the zoom-in views of the recovery with AAD selection, good both visually and by PSNR measurement.

Finally, we compare the OABE with an up-to-date block-loss concealment method, RIBMAP (recovery of image blocks using the method of alternating projection) of [17] (by Park *et al.*, 2005). As in [17], we simulate on images *Masquerade* and *Lena* the loss pattern with lost block size 8×8, and block loss rate (BLR) 23.46%. With AAD selection, the OABE yields PSNR's of 25.38dB and 28.44dB respectively (Fig. 7). The PSNR comparison (Table 1) shows that we have achieved 1.8dB improvement on the *Masquerade* image. This comparison (with RIBMAP and the methods compared in [17]) shows the OABE superior to previous methods on recovering images rich in textures, and the satisfactory performance is maintained on structural images.

The OABE can be readily extended to recovering missing regions of general shape. Given any finite set of missing points, with cleanly received pixels surrounding the missing region, there is a discrete version of the equation that can be satisfied uniquely. The OABE has its limitation, though. It is unable to recover the missing regions that contain "Y"-shape junctions.

Table 1. PSNR comparison (in dB) to the RIBMAP in [17]

BLR 23.46%	Lena		Masquerade	
	PSNR over recovered blocks	PSNR over entire image	PSNR over recovered blocks	PSNR over entire image
RIBMAP	—	34.65	—	29.87
OABE	28.44	34.73	25.38	31.68

4 Conclusion

This paper presents a novel anisotropic elliptic PDE model. In contrast to the previous methods that treat *edge completion* and *texture synthesis* mutually exclusively, our approach repairs missing structures and textures by one uniformed, fully automated algorithm. Both textures and edges exhibit repetitiveness, with near-shared values typically displaced by the repetition vector (RV), which we compute via a classical optical flow vector computation method. We construct our anisotropic PDE model on the basis formed by the RV and its normal. We have mathematically shown the existence and uniqueness of the solution for the

Fig. 5. 512×512 standard test images. The left column: from top to bottom, original images referred to as *Lena*, *Couple*, *Goldhill* and *Boat*. The middle column: received corrupted images without error concealment, with a decoding failure in the 11th ERC-SPIHT bitstream; The right column: the recovered images by the proposed OABE model. From top to bottom, the PSNR results (in dB) are 28.44, 27.88, 26.93 and 25.82.

OABE, and presented the iteration scheme yielding the solution. Extensive experiments verify the high fidelity of the OABE on both structures and textures.

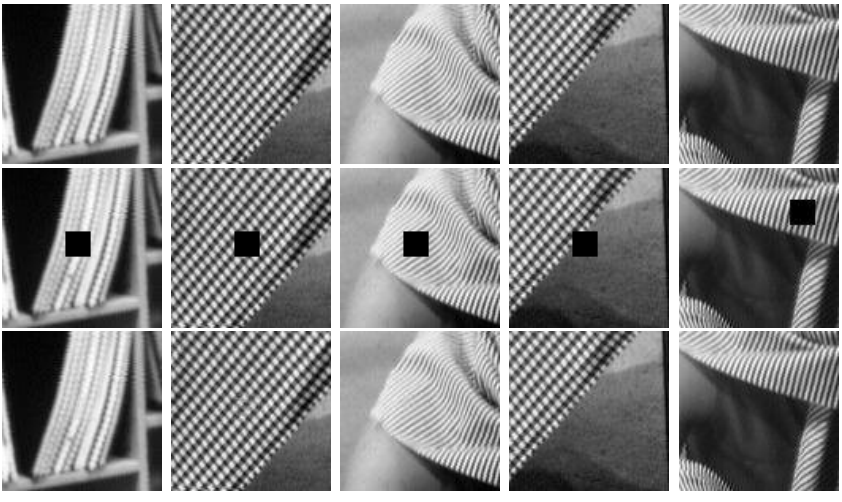


Fig. 6. Zoom-in view if the OABE recovery on lost blocks of size 16×16. The first line: segments from the original *Barbara* image; The middle line: corruption with simulated loss; The third line: the recovery results of the proposed OABE model. From left to right, the PSNR scores (in dB) are 28.80, 27.61, 31.39, 30.50 and 30.87.

Fig. 7. The images *Lena* and *Masquerade* with simulated corruption at block loss rate 23.46%, and the images recovered by the OABE

References

1. M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester: Image inpainting. Computer Graphics, SIGGRAPH 2000 July (2000) 417–424
2. T. Chan, J. Shen: Mathematical models for local non-texture inpaintings. SIAM J. of App. Math. **62(3)** (2001) 1019-1043
3. S. Masnou, J. Morel: Level lines based disocclusion. Proceedings of IEEE Conf. on Image Processing (1998) 259-262
4. A. A. Efros, T. K. Leung: Texture synthesis by nonparametric sampling. Proc. IEEE Conf. Comp. Vision (1999) 1033-1038
5. S. D. Rane, G. Sapiro, M. Bertalmio: Structure and texture filling-in of missing image blocks in wireless transmission and compression applications. IEEE Trans. on Image Processing **12(3)** (2003) 296-303
6. M. Bertalmio, L. Vese, G. Sapiro, S. Osher: Simultaneous structure and texture image inpainting. IEEE Trans. on Image Processing, **12(8)** (2003) 882-889
7. S. T. Acton, D. P. Mukherjee, J. P. Havlicek, A. C. Bovik: Oriented texture completion by AM-FM reaction-diffusion. IEEE Trans. Image Processing **10(6)** (2001) 885-896
8. J. Weickert, H. Scharr, A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance: J. Visual Communication and Image Representation, **13(1/2)** (2002) 103-118
9. L. A. Vese, S. J. Osher: Modeling textures with total variation minimization and oscillating patterns in image processing. J. of Scientific Computing **19(1-3)** (2003) 553-572
10. A. Witkin, M. Kass: Reaction-diffusion textures. SIGGRAPH (1991) 299-308
11. A. R. Rao, G. L. Lohse: Towards a texture naming system: Identifying relevant dimensions of texture. Proc. of IEEE Conf. on Visualization. (1993) 220-227
12. E. Trucco, A. Verri: Introductory Techniques for 3-D Computer Vision. Prentice Hall (1998)
13. P. Perona, J. Malik: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. PAMI. **12(7)** (1990) 629-639
14. D. Tschumperle, R. Deriche: Vector-valued image regularization with PDE's: A common framework for different applications. IEEE Trans. PAMI, **27(4)** (2005) 506-517
15. G. Strang, Linear Algebra and Its Applications. New York, Academic Press (1976)
16. O. Axelsson, Iterative Solution Methods. Cambridge Univ. Press (1994)
17. J. Park, D.-C. Park, R. J. Marks, M. A. El-Sharkawi: Recovery of image blocks using the method of alternating projections. IEEE Tran. Image Processing. **14(4)** April (2005) 461-474
18. S. Cho, W. A. Pearlman: Error resilient video coding with improved 3-D SPIHT and error concealment. SPIE/IS&T Elec. Imaging, Proc. SPIE **5022**, (2003). Also available at http://www.cipr.rpi.edu/ pearlman/

Image Cartoon-Texture Decomposition and Feature Selection Using the Total Variation Regularized L^1 Functional

Wotao Yin[1], Donald Goldfarb[1], and Stanley Osher[2]

[1] Department of Industrial Engineering and Operations Research,
Columbia University, New York, NY, USA
{wy2002, goldfarb}@columbia.edu
[2] Department of Mathematics, University of California at Los Angeles,
Los Angeles, CA, USA
sjo@math.ucla.edu

Abstract. This paper studies the model of minimizing total variation with an L^1-norm fidelity term for decomposing a real image into the sum of cartoon and texture. This model is also analyzed and shown to be able to select features of an image according to their scales.

1 Introduction

Let f be an observed image which contains texture and/or noise. Texture is characterized as repeated and meaningful structure of small patterns. Noise is characterized as uncorrelated random patterns. The rest of an image, which is called *cartoon*, contains object hues and sharp edges (boundaries). Thus an image f can be decomposed as $f = u + v$, where u represents image cartoon and v is texture and/or noise. A general way to obtain this decomposition using the variational approach is to solve the problem $\min \{\int |Du| \ | \ \|u-f\|_B \leq \sigma\}$, where Du denotes the generalized derivative of u and $\|\cdot\|_B$ is a norm (or semi-norm). The total variation of u, which is $\int |Du|$, is minimized to regularize u while keep edges like object boundaries of f in u (i.e. allow discontinuities in u). The fidelity term $\|t(u,f)\|_B \leq \sigma$ forces u to be close to f. Among the recent total variation-based cartoon-texture decomposition models, Meyer [15] and Haddad & Meyer [12] proposed to use the G-norm, Vese & Osher [23] approximated the G-norm by the $\mathrm{div}(L^p)$-norm, Osher & Sole & Vese [20] proposed to use the H^{-1}-norm, Lieu & Vese [14] proposed to use the more general H^{-s}-norm, and Le & Vese [13] proposed to use the $\mathrm{div}(BMO)$-norm. In addition, Alliney [2,3,4], Nikolova [16,17,18], and Chan & Esedoglu [8] used the L^1-norm together with total variation. In this paper, we study the TV-L^1 model.

The rest of the paper is organized as follows. In Section 2 we define certain fundamental function spaces and norms. In Section 3 we present and analyze the TV-L^1 model. In particular, we relate the level sets of the input to the solution of the TV-L^1 model using a geometric argument and discuss the scale-selection and morphologically invariant properties of this model. The proofs of the lemmas, theorems, and corollaries are given in the technical report [25]. In Section 4 we briefly give the second-order

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 73–84, 2005.

cone programming (SOCP) formulation of this model. Numerical results illustrating the properties of the model are given in Section 5.

2 Preliminaries

Let $u \in L^1$, and define the total variation of u as

$$\|Du\| := \sup \left\{ \int u \operatorname{div}(\boldsymbol{g})\, dx : \begin{array}{l} \boldsymbol{g} \in C_0^1(\mathbb{R}^n; \mathbb{R}^n), \\ |\boldsymbol{g}(x)|_{l^2} \le 1 \ \forall x \in \mathbb{R}^n \end{array} \right\},$$

and the BV-norm of u as $\|u\|_{BV} := \|u\|_{L^1} + \|Du\|$, where $C_0^1(\mathbb{R}^n; \mathbb{R}^n)$ denotes the set of continuously differentiable vector-valued functions that vanish at infinity. The Banach space of functions with bounded variation is defined as$BV := \left\{u \in L^1 : \|u\|_{BV} < \infty\right\}$ and is equipped with the $\|\cdot\|_{BV}$-norm. $\|Du\|$ is often written in a less mathematically strict form $\int |\nabla u|$.

$\|Du\|$ and $BV(\Omega)$ limited to Ω are defined analagously using $\boldsymbol{g} \in C_0^1(\Omega; \mathbb{R}^n)$.

Sets in $\mathbb{R}^n$ with finite perimeter are often referred to as BV sets. The perimeter of a set S is defined by $\operatorname{Per}(S) := \|D\mathbf{1}_S\|$, where $\mathbf{1}_S$ is the indicator function of S.

Next, we define the space G [15]. Let G denote the Banach space consisting of all *generalized functions* $v(x)$ defined on $\mathbb{R}^n$ that can be written as

$$v = \operatorname{div}(\boldsymbol{g}), \quad \boldsymbol{g} = [g_i]_{i=1,\ldots,n} \in L^\infty(\mathbb{R}^n; \mathbb{R}^n), \tag{1}$$

and equipped with the norm $\|v\|_G$ defined as the infimum of all L^∞ norms of the functions $|\boldsymbol{g}(x)|_{l^2}$ over all decompositions (1) of v. In short, $\|v\|_G := \inf\{\| \, |\boldsymbol{g}(x)|_{l^2} \, \|_{L^\infty} : v = \operatorname{div}(\boldsymbol{g})\}$.

G is the dual of the closed subspace $\mathcal{BV}$ of BV, where $\mathcal{BV} := \{u \in BV : \nabla u \in L^1\}$ [15]. We note that finite difference approximations to functions in BV and in $\mathcal{BV}$ are the same. For the definition and properties of $G(\Omega)$, where $\Omega \subset \mathbb{R}^n$, see [6].

It follows from the definitions of the BV and G spaces that

$$\int u\, v = \int u \nabla \cdot \boldsymbol{g} = -\int Du \cdot \boldsymbol{g} \le \|Du\| \|v\|_G, \tag{2}$$

holds for any $u \in \mathcal{BV}$ with a compact support and $v \in G$. We say (u, v) is an *extremal pair* if (2) holds with equality.

3 The TV-L^1 Model

The TV-L^1 model is define as a variational problem

$$\min_{u \in BV} TVL1_\lambda(u) = \min_{u \in BV} \int_\Omega |\nabla u| + \lambda \int |f - u|. \tag{3}$$

Although this model appears to be simple, it is very different to the ROF model [21]: it has the important property of being able to separate out features of a certain scale in an image as we shall show in the next section.

In the rest of this section we first relate the parameter λ to the G-norm of the texture output v, then we focus on the TV-L^1 geometry and discuss the properties of the TV-L^1 model for scale-based feature selection in subsection 3.1.

Meyer [15] recently showed that the G space, which is equipped with the G-norm, contains functions with high oscillations. He characterized the solution u of the ROF model using the G-norm: given any input f defined on $\mathbb{R}^n$, u satisfies $\|f-u\|_G = \frac{1}{2\lambda}$ if $\lambda > (2\|f\|_G)^{-1}$, and u vanishes (i.e., $u \equiv 0$) if $0 \leq \lambda \leq (2\|f\|_G)^{-1}$. We can interpret this result as follows. First, no matter how regular f is, u is always different to f as long as $f \not\equiv 0$. This is a major limitation of the ROF model, but it can be relaxed by applying the ROF model iteratively [19] or by using the inverse TV flow [7]. Second, the texture/noise output v has its G-norm given by $\min\{\frac{1}{2\lambda}, \|f\|_G\}$. Therefore, the oscillating signal with G-norm less than $\frac{1}{2\lambda}$ is removed by the ROF model. A similar characterization is given below for the TV-L^1 model in Theorems 1 and 2.

In order to use the G-norm, we first consider the approximate TV-L^1 model in which a perturbation ϵ has been added to the fidelity term $\|f-u\|_{L^1}$ to make it differentiable:

$$\min_{u\in BV(\Omega)} \int_\Omega |\nabla u| + \lambda \int_\Omega \sqrt{(f-u)^2+\epsilon}, \tag{4}$$

where the image support Ω is assumed to be compact. Since $TVL1_{\lambda,\epsilon}(u)$ is strictly convex, problem (4) has a unique solution $u_{\lambda,\epsilon}$.

Theorem 1. *The solution $u_{\lambda,\epsilon}(= f - v_{\lambda,\epsilon}) \in BV(\Omega)$ of the approximate TV-L^1 model satisfies*

$$\|\mathrm{sign}_\epsilon(v_{\lambda,\epsilon})\|_G \leq 1/\lambda,$$

where $\mathrm{sign}_\epsilon(\cdot)$ is defined point-wise by $\mathrm{sign}_\epsilon(g)(x) := g(x)/\sqrt{|g(x)|^2+\epsilon}$ for any function g.

Moreover, if $\|\mathrm{sign}_\epsilon(f)\|_G \leq 1/\lambda$, $u_{\lambda,\epsilon} \equiv 0$ is the solution of the approximate TV-L^1 model.

If $\|\mathrm{sign}_\epsilon(f)\|_G > 1/\lambda$, then there exists an optimal solution $u_{\lambda,\epsilon}$ satisfying

- $\|\mathrm{sign}_\epsilon(v_{\lambda,\epsilon})\|_G = 1/\lambda$;
- $\int u_{\lambda,\epsilon}\,\mathrm{sign}_\epsilon(v_{\lambda,\epsilon}) = \|Du_{\lambda,\epsilon}\|/\lambda$, *i.e., $u_{\lambda,\epsilon}$ and $\mathrm{sign}_\epsilon(v_{\lambda,\epsilon})$ form an extremal pair.*

Next, we relate the solution of the perturbed TV-L^1 model to the solution of the (unperturbed) TV-L^1 model.

Theorem 2. *Assuming the TV-L^1 model (3) using parameter λ has a unique solution u_λ, then the solution of approximate TV-L^1 model (4) using the same parameter λ satisfies*

$$\lim_{\epsilon\downarrow 0+} \|u_{\lambda,\epsilon} - u_\lambda\|_{L^1} = 0, \quad \lim_{\epsilon\downarrow 0+} \|v_{\lambda,\epsilon} - v_\lambda\|_{L^1} = 0.$$

We note that Chan and Esedoglu [8] proved that (4) has a unique solution for almost all λ's with respect to the Lebesgue measure.

In the above two theorems, for ϵ small enough, the value of $\mathrm{sign}_\epsilon(v)(x)$ can be close to $\mathrm{sign}(v)(x)$ even for small $v(x)$. In contrast to $\|v\|_G = \min\{\frac{1}{2\lambda}, \|f\|_G\}$ for the solution v of the ROF model, Theorems 1 and 2 suggest that the solution v of the TV-L^1

model can be much smaller. In other words, the TV-L^1 may not always remove some oscillating signal from f and erode the structure. This is supported by the following analytic example from [8]: if f is equal to the disk signal B_r, which has radius r and unit height, then the solution u_λ of the TV-L^1 model is 0 if $0 < \lambda < 2/r$, f if $\lambda > 2/r$, and cf for any $c \in [0,1]$ if $\lambda = 2/r$. Clearly, depending on λ, either 0 or the input f minimizes the TV-L^1 functional. This example also demonstrates the ability of the model to select the disk feature by its "scale" $r/2$. The next subsection focuses on this scale-based selection.

3.1 TV-L^1 Geometry

To use the TV-L^1 model to separate large-scale and small-scale features, we are often interested in an appropriate λ that will allow us to extract geometric features of a given scale. For general input, the TV-L^1 model, which has only one scalar parameter λ, returns images combining many features. Therefore, we are interested in determining a λ that gives the whole targeted features with the least unwanted features in the output.

For simplicity, we assume $\Omega = \mathbb{R}^2$ in this section. Our analysis starts with the decomposition of f using level sets and relies on the co-area formula (5) [11] and "layer cake" formula (6) [8], below. Then, we derive a TV-L^1 solution formula (8), in which u^* is built slice by slice. Each slice is then characterized by feature scales using the G-value, which extends the G-norm, and the *slopes* in Theorem 3, below. Last, we relate the developed properties to real-world applications. In the following we let $U(g,\mu) := \{x \in \mathrm{Dom}(g) : g(x) > \mu\}$ denote the (upper) level set of a function g at level μ.

The co-area formula [11] for functions of bounded variation is

$$\int |Du| = \int_{-\infty}^{\infty} \mathrm{Per}(U(u,\mu))\, d\mu. \tag{5}$$

Using (5), Chan and Esedoglu [8] showed that the $TVL1_\lambda$ functional can be represented as an integral over the perimeter and weighted areas of certain level sets by the following "layer cake" formula:

$$\begin{array}{l} TVL1_\lambda(u) = \int_{-\infty}^{\infty} (\mathrm{Per}(U(u,\mu)) \\ +\lambda\,|U(u,\mu)\backslash U(f,\mu)| + \lambda\,|U(f,\mu)\backslash U(u,\mu)|)\mathrm{d}\mu, \end{array} \tag{6}$$

where $|S|$ for a set S returns the area of S. Therefore, an optimal solution u_λ to the TV-L^1 model can be obtained by minimizing the right-hand side of (6). We are interested in finding a u^* such that $U(u^*,\mu)$ minimizes the integrant for almost all μ.

Let us fix λ and focus on the integrand of the above functional and introduce the problem

$$\min_{\Sigma} C(\Gamma,\Sigma) \tag{7}$$

where $C(\Gamma,\Sigma) := \mathrm{Per}(\Sigma) + \lambda|\Sigma\backslash\Gamma| + \lambda|\Gamma\backslash\Sigma|$, and Γ and Σ are sets with bounded perimeters in $\mathbb{R}^2$. Let $\Sigma_{f,\mu}$ denote a solution of (7) for $\Gamma = U(f,\mu)$. From the definition of the upper level set, for the existence of a u satisfying $U(u,\mu) = \Sigma_{f,\mu}$ for all μ, we need $\Sigma_{f,\mu_1} \supseteq \Sigma_{f,\mu_2}$ for any $\mu_1 < \mu_2$. This result is given in the following lemma:

Lemma 1. *Let the sets Σ_1 and Σ_2 be the solutions of (6) for $\Gamma = \Gamma_1$ and $\Gamma = \Gamma_2$, respectively, where Γ_1 and Γ_2 are two sets satisfying$\Gamma_1 \supset \Gamma_2$.*

If either one or both of Σ_1 and Σ_2 are unique minimizers, then $\Sigma_1 \supseteq \Sigma_2$; otherwise, i.e., both are not unique minimizers, $\Sigma_1 \supseteq \Sigma_2$ may not hold, but in this case, $\Sigma_1 \cup \Sigma_2$ is a minimizer of (7) for $\Gamma = \Gamma_1$.

Therefore, there always exists a solution of (7) for $\Gamma = \Gamma_1$ that is a superset of any minimizer of (7) for $\Gamma = \Gamma_2$.

Using the above lemma, we get the following geometric solution characterization for the TV-L^1 model:

Theorem 3. *Suppose that $f \in BV$ has essential infimum μ_0. Let function u^* be defined point-wise by*

$$u^*(x) := \mu_0 + \int_{\mu_0}^{\infty} \mathbf{1}_{\Sigma_{f,\mu}}(x) d\mu, \tag{8}$$

where $\Sigma_{f,\mu}$ is the solution of (7) for $\Gamma = U(f,\mu)$ that satisfies $\Sigma_{f,\mu_1} \supseteq \Sigma_{f,\mu_2}$ for any $\mu_1 < \mu_2$, i.e., $\Sigma_{f,\mu}$ is monotonically decreasing with respect to μ. Then u^* is an optimal solution of the TV-L^1 model (3).

Next, we illustrate the implications of the above theorem by applying the results in [22] to (7). In [22], the authors introduced the G-value, which is an extension of Meyer's G-norm, and obtained a characterization to the solution of the TV-L^1 model based on the G-value and the *Slope* [5]. These results are presented in the definition and the theorem below.

Definition 1. *Let $\Psi : \mathbb{R}^2 \to 2^{\mathbb{R}}$ be a set-valued function that is measurable in the sense that $\Psi^{-1}(S)$ is Lebesgue measurable for every open set $S \subset \mathbb{R}$. We do not distinguish Ψ between a set-valued function and a set of measurable (single-valued) functions, and let*

$$\Psi := \{\textit{measurable function } \psi \textit{ satisfying } \psi(x) \in \Psi(x), \forall x\}.$$

The G-value of Ψ is defined as follows:

$$G(\Psi) := \sup_{h \in C_0^\infty : \int |\nabla h| = 1} - \sup_{\psi \in \Psi} \int \psi(x) h(x) dx. \tag{9}$$

Theorem 4. *Let $\partial |f|$ denote the set-valued sub-derivative of $|f|$, i.e., $\partial |f|(x)$ equals $\operatorname{sign}(f(x))$ if $f(x) \neq 0$ and equals the interval $[-1,1]$ if $f(x) = 0$. Then, for the TV-L^1 model (3),*

1. *$u_\lambda = 0$ is an optimal solution if and only if $\lambda \leq \frac{1}{G(\partial |f|)}$;*
2. *$u_\lambda = f$ is an optimal solution if and only if $\lambda \geq \sup_{h \in BV} \frac{\|Df\| - \|Dh\|}{\int |f-h|}$,*

where $\frac{1}{G(\partial |f|)} \leq \sup_{h \in BV} \frac{\|Df\| - \|Dh\|}{\int |f-h|}$, $\forall f \in BV$.

It follows from the "layer cake" formula (6) that solving the geometric problem (7) is equivalent to solving the TV-L^1 model with input $f = \mathbf{1}_\Gamma$. Therefore, by applying Theorem 4 to $f = \mathbf{1}_\Gamma$, we can characterize the solution of (6) as follows:

Corollary 1. *For the geometric problem (7) with a given λ,*

1. *$\Sigma_\lambda = \emptyset$ is an optimal solution if and only if $\lambda \leq \frac{1}{G(\partial|\mathbf{1}_\Gamma|)}$;*
2. *$\Sigma_\lambda = \Gamma$ is an optimal solution if and only if $\lambda \geq \sup_{h\in BV} \frac{\|D\mathbf{1}_\Gamma\| - \|Dh\|}{\int|\mathbf{1}_\Gamma - h|}$.*

Corollary 1, together with Theorem 3, implies the followings. Suppose that the mask set S of a geometric feature F coincides with $U(f,\mu)$ for $\mu \in [\mu_0, \mu_1)$. Then, for any $\lambda < 1/G(\partial|\mathbf{1}_S|)$, $\mathbf{1}_{\Sigma_{f,\mu}} = \emptyset$ for $\mu \in [\mu_0, \mu_1)$; hence, the geometric feature F is not observable in u_λ. In the example where $F = f = cB_r$ (recall that B_r is the disk function with radius r and unit height), S and $U(f,\mu)$ are the circle $\bar{B}_r$ with radius r for $\mu \in [0, c)$, and $G(\partial|\mathbf{1}_S|) = G(\partial|B_r|) = r/2$. Therefore, if $\lambda < 1/G(\partial|\mathbf{1}_S|) = 2/r$, $\mathbf{1}_{\Sigma_{f,\mu}} = \emptyset$ for $\mu \in [0, c)$. Also because $\mu_0 = 0$ and $\mathbf{1}_{\Sigma_{f,\mu}} = \emptyset$ for $\mu \geq c$ in (8), $u_\lambda \equiv 0$, which means the feature $F = cB_r$ is not included in u_λ.

If $\lambda > 1/G(\partial|\mathbf{1}_S|)$, $\Sigma_{f,\mu} \neq \emptyset$ for $\mu \in [\mu_0, \mu_1)$, which implies at least some part of the feature F can be observed in u_λ. Furthermore, if $\lambda \geq \sup_{h\in BV}(\|D\mathbf{1}_\Gamma\| - \|Dh\|)/\int|\mathbf{1}_\Gamma - h|$, we get $\Sigma_{f,\mu} = U(f,\mu) = S$ for $\mu \in [\mu_0, \mu_1)$ and therefore, the feature F is fully contained in u_λ. In the above example where $F = f = cB_r$ and $S = \bar{B}_r$, it turns out $2/r = 1/G(\partial|\mathbf{1}_S|) = \sup_{h\in BV}(\|D\mathbf{1}_\Gamma\| - \|Dh\|)/\int|\mathbf{1}_\Gamma - h|$. Therefore, if $\lambda > 2/r$, $\Sigma_{f,\mu} = S$ for $\mu \in [0, c)$, and $u_\lambda = cB_r = f$.

In general, although a feature is often different from its vicinity in intensity, it cannot monopolize a level set of the input f, i.e., it is represented by an isolated set in $U(f,\mu)$, for some μ, which also contains isolated sets representing other features. Consequently, u_λ that contains a targeted feature may also contain many other features. However, from Theorem 3 and Corollary 1, we can easily see that the arguments for the case $S = U(f,\mu)$ still hold for the case $S \subset U(f,\mu)$.

Proposition 1. *Suppose there are a sequences of features in f that are represented by sets $S_1, S_2, \ldots, S_l$ and have distinct intensity values. Let*

$$\lambda_i^{\min} := \frac{1}{G(\partial|\mathbf{1}_{S_i}|)},\ \lambda_i^{\max} := \sup_{h\in BV} \frac{\|D\mathbf{1}_{S_i}\| - \|Dh\|}{\int|\mathbf{1}_{S_i} - h|}, \tag{10}$$

for $i = 1, \ldots, l$. If the features have decreasing scales and, in addition, the following holds

$$\lambda_1^{\min} \leq \lambda_1^{\max} < \lambda_2^{\min} \leq \lambda_2^{\max} < \ldots < \lambda_l^{\min} \leq \lambda_l^{\max}, \tag{11}$$

then feature i, for $i = 1, \ldots, l$, can be precisely retrieved as $u_{\lambda_i^{\max}+\epsilon} - u_{\lambda_i^{\min}-\epsilon}$ (here ϵ is a small scalar that forces unique solutions because $\lambda_i^{\min} = \lambda_i^{\max}$ is allowed).

This proposition holds since for $\lambda = \lambda_i^{\min} - \epsilon$, feature i completely vanishes in u_λ, but for $\lambda = \lambda_i^{\max} - \epsilon$, feature i is fully contained in u_λ while there is no change to any other features.

To extract a feature represented by set S in real-world applications, one can compute $G(\partial|\mathbf{1}_S|)$ off-line and use a λ slightly greater than $1/G(\partial|\mathbf{1}_S|)$. The intensity and the position of the feature in f are not required as priors.

Next, we present a corollary of Theorem 3 to finish this section.

Corollary 2. ***[Morphological invariance]*** *For any strictly increasing function $g : \mathbb{R} \to \mathbb{R}$, $u_\lambda(g \circ f) = g \circ u_\lambda(f)$.*

4 Second-Order Cone Programming Formulations

In this section, we briefly show how to formulate the discrete version of the TV-L^1 model (3) as a second-order program (SOCP) so that it can be solved in polynomial time.

In an SOCP the vector of variables $\mathbf{x} \in \mathbb{R}^n$ is composed of subvectors $\mathbf{x_i} \in \mathbb{R}^{n_i}$ – i.e., $\mathbf{x} \equiv (\mathbf{x_1}; \mathbf{x_2}; \ldots; \mathbf{x_r})$ – where $n = n_1 + n_2 + \ldots + n_r$ and each subvector $\mathbf{x_i}$ must lie either in an elementary *second-order cone* of dimension n_i

$$\mathcal{K}^{n_i} \equiv \{\mathbf{x_i} = (x_i^0; \bar{\mathbf{x}}_\mathbf{i}) \in \mathbb{R} \times \mathbb{R}^{n_i - 1} \mid \|\bar{\mathbf{x}}_\mathbf{i}\| \leq x_i^0\},$$

or an n_i-dimensional *rotated second-order cone*

$$\mathcal{Q}^{n_i} \equiv \{\mathbf{x_i} \in \mathbb{R}^{n_i} \mid \mathbf{x_i} = \bar{\mathbf{x}}, 2\bar{x}_1\bar{x}_2 \geq \sum_{i=3}^{n_i} \bar{x}_i^2,\ \bar{x}_1, \bar{x}_2 \geq 0\},$$

which is an elementary second-order cone under a linear transformation.

With these definitions an SOCP can be written in the following form [1]:

$$\begin{array}{ll} \min & \mathbf{c_1}^\top \mathbf{x_1} + \cdots + \mathbf{c_r}^\top \mathbf{x_r} \\ \text{s.t.} & A_1\mathbf{x_1} + \cdots + A_r\mathbf{x_r} = \mathbf{b} \\ & \mathbf{x_i} \in \mathcal{K}^{n_i} \text{ or } \mathcal{Q}^{n_i}, \quad \text{for } i = 1, \ldots, r, \end{array} \tag{12}$$

where $\mathbf{c_i} \in \mathbb{R}^{n_i}$ and $A_i \in \mathbb{R}^{m \times n_i}$, for any i, and $\mathbf{b} \in \mathbb{R}^m$. As is the case for linear programs, SOCPs can be solved in polynomial time by interior point methods.

We assume that images are represented as 2-dimensional $n \times n$ matrices, whose elements give the "grey" values of corresponding pixels, i.e., $f_{i,j} = u_{i,j} + v_{i,j}$, for $i, j = 1, \ldots, n$.

First, as the total variation of u is defined discretely by forward finite differences as $\int |\nabla u| := \sum_{i,j}[((\partial_x^+ u)_{i,j})^2 + ((\partial_y^+ u)_{i,j})^2]^{1/2}$, by introducing new variables $t_{i,j}$, we can express $\min\{\int |\nabla u|\}$ as $\min\{\sum_{i,j} t_{i,j}\}$ subject to the 3-dimensional second-order cones $(t_{i,j}\ ;\ (\partial_x^+ u)_{i,j}, (\partial_y^+ u)_{i,j}) \in \mathcal{K}^3$. Second, minimizing the fidelity term $\int |f - u|$ is equivalent to minimizing s subject to $\sum_{i,j}(f_{i,j} - u_{i,j}) \leq s$ and $\sum_{i,j}(u_{i,j} - f_{i,j}) \leq s$. Therefore, the SOCP formulation of the TV-L^1 model is

$$\begin{array}{lll} \min_{s,t,u,\partial_x^+ u, \partial_y^+ u} & \sum_{1 \leq i,j \leq n} t_{i,j} + \lambda s & \\ \text{s.t.} & (\partial_x^+ u)_{i,j} = u_{i+1,j} - u_{i,j} & \forall i, j = 1, \ldots, n, \\ & (\partial_y^+ u)_{i,j} = u_{i,j+1} - u_{i,j} & \forall i, j = 1, \ldots, n, \\ & \sum_{1 \leq i,j \leq n}(f_{i,j} - u_{i,j}) \leq s, & \\ & \sum_{1 \leq i,j \leq n}(u_{i,j} - f_{i,j}) \leq s, & \\ & (t_{i,j}\ ;\ (\partial_x^+ u)_{i,j}, (\partial_y^+ u)_{i,j}) \in \mathcal{K}^3 & \forall i, j = 1, \ldots, n. \end{array} \tag{13}$$

Finally, we note that both $G(\partial|f|)$ and $\sup_{h \in BV} \frac{\|Df\| - \|Dh\|}{\int |f-h|}$, after homogenizing the objective function of the latter, can be easily developed based on the SOCP formulation of the total variation term $\int |Dh|$.

5 Numerical Results

5.1 Comparison Among Three Decomposition Models

In this subsection, we present numerical results of the TV-L^1 model and compare them with the results of the Meyer [15] and the Vese-Osher (VO) [23] models, below.

$$\text{The Meyer model: } \min_{u\in BV} \{ \int |\nabla u| \ : \ \|v\|_G \leq \sigma, \ f = u + v\}.$$

$$\text{The Vese-Osher model: } \min_{u\in BV} \int |\nabla u| + \lambda \int |f - u - \text{div}(\boldsymbol{g})|^2 + \mu \int |\boldsymbol{g}|.$$

We also formulated these two models as SOCPs, in which no regularization or approximation is used (refer to [10] for details). We used the commercial package Mosek as our SOCP solver. In the first set of results, we applied the models to relatively noise-free images.

We tested textile texture decomposition by applying the three models to a part (Fig. 1 (b)) of the image "Barbara" (Fig. 1 (a)). Ideally, only the table texture and the strips on Barbara's clothes should be extracted. Surprisingly, Meyer's model did not give good results in this test as the texture v output clearly contains inhomogeneous background. To illustrate this effect, we used a very conservative parameter - namely, a small σ - in Meyer's model. The outputs are depicted in Fig. 1 (d). As σ is small, some table cloth and clothes textures remain in the cartoon u part. One can imagine that by increasing σ we can get a result with less texture left in the u part, but with more inhomogeneous background left in the v part. While Meyer's model gave unsatisfactory results, the other two models gave very good results in this test as little background is shown in Figures 1 (e) and (f). The Vese-Osher model was originally proposed as an approximation of Meyer's model in which the L^∞-norm of $|\boldsymbol{g}|$ is approximated by the L^1-norm of $|\boldsymbol{g}|$. We guess that the use of the L^1-norm allows $\boldsymbol{g}$ to capture more texture signal while the original L^∞-norm in Meyer's model makes $\boldsymbol{g}$ to capture only the oscillatory pattern of the texture signal. Whether the texture or only the oscillatory pattern is more preferable depends on the applications. For example, the latter is more desirable in analyzing fingerprint images. Compared to the Vese-Osher model, the TV-L^1 model generated a little sharper cartoon in this test. The biggest difference, however, is that the TV-L^1 model kept most brightness changes in the texture part while the other two kept them in the cartoon part. In the top right regions of the output images, the wrinkles of Barbara's clothes are shown in the u part of Fig. 1 (e) but in the v part of (f). This shows that the texture extracted by TV-L1 has a wider dynamic range.

In the second set of results, we applied the three models to the image "Barbara" after adding a substantial amount of Gaussian noise (standard deviation equal to 20). The resulting noisy image is depicted in Fig. 1 (c). All the three models removed the noise together with the texture from f, but noticeably, the cartoon parts u in these results (Fig. 1 (g)-(l)) exhibit a staircase effect to different extents. We tested different parameters and conclude that none of the three decomposition models is able to separate image texture and noise.

Table 1

Component	$\bar{S}_1$	$\bar{S}_2$	$\bar{S}_3$	$\bar{S}_4$	$\bar{S}_5$
G-value	19.39390	13.39629	7.958856	4.570322	2.345214
$\lambda^{\min}$	0.0515626	0.0746475	0.125646	0.218803	0.426400
$\lambda_1 =$	$\lambda_2 =$	$\lambda_3 =$	$\lambda_4 =$	$\lambda_5 =$	$\lambda_6 =$
0.0515	0.0746	0.1256	0.2188	0.4263	0.6000

(a) 512×512 "Barbara"

(b) a 256×256 part of (a)

(c) noisy "Barbara" (std.=20)

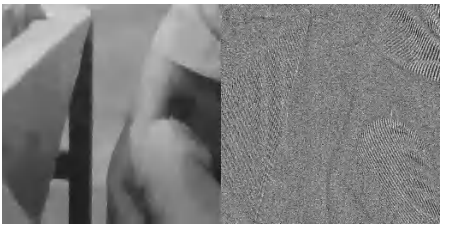
(d) Meyer ($\sigma = 15$) applied to (b)

(e) Vese-Osher ($\lambda = 0.1$, $\mu = 0.5$) applied to (b)

(f) TV-L^1 ($\lambda = 0.8$) applied to (b)

(g) Meyer ($\sigma = 20$) applied to (c)

(h) Vese-Osher ($\lambda = 0.1$, $\mu = 0.5$) applied to (c)



(l) TV-L^1 ($\lambda = 0.8$) applied to (c)

Fig. 1. Cartoon-texture decomposition and denoising results by the three models

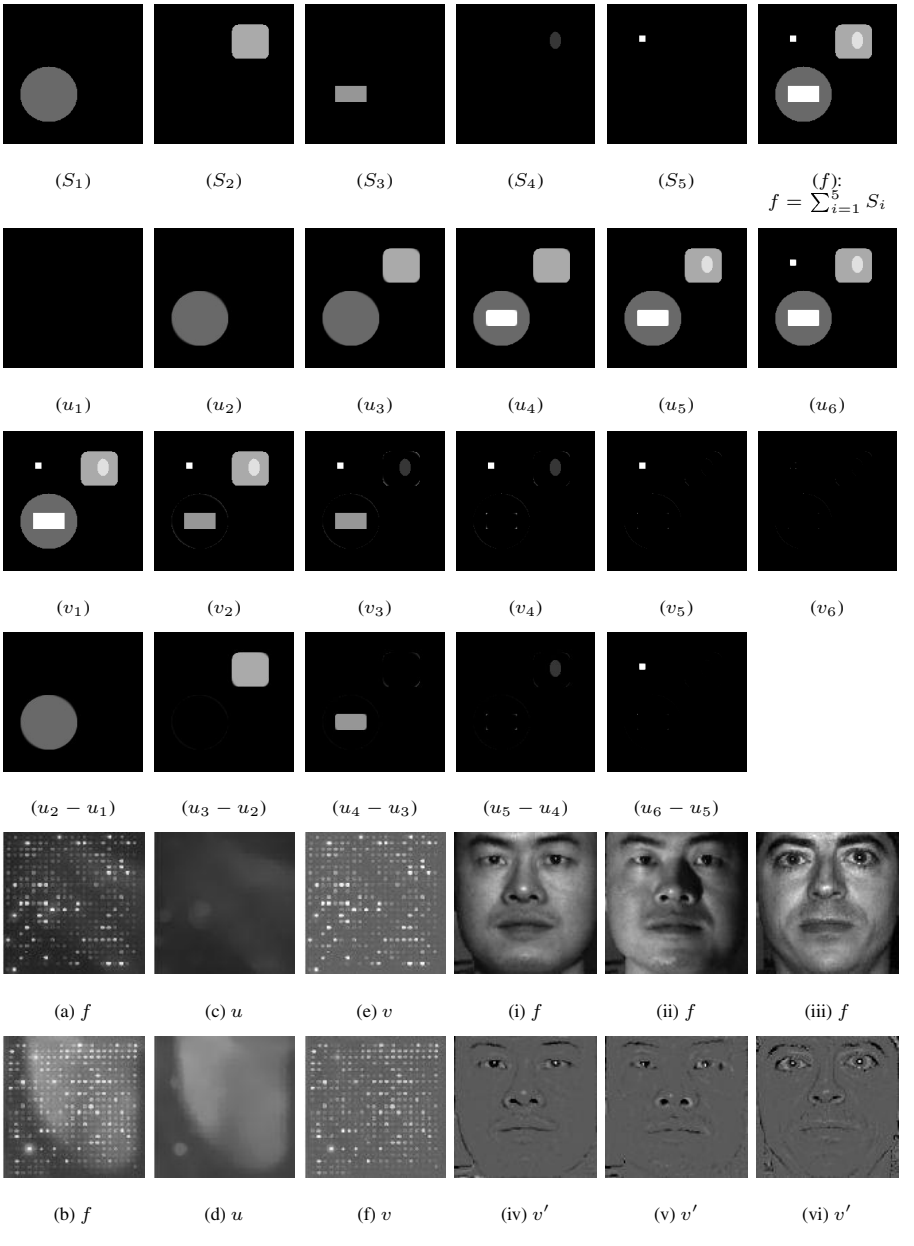


Fig. 2. Feature selection using the TV-L^1 model

5.2 Feature Selection Using the TV-L^1 Model

We applied the TV-L^1 model with different λ's to the composite input image (Fig. 2 (f)). Each of the five components in this composite image is depicted in Fig. 2 (S_1)-(S_5). We name the components by $S_1, \ldots, S_5$ in the order they are depicted in Fig. 2.

They are decreasing in scale. This is further shown by the decreasing G-values of their mask sets $\bar{S}_1, \ldots, \bar{S}_5$, and hence, their increasing $\lambda^{\min}$ values (see (10)), which are given in Table 1. We note that $\lambda_1^{\max}, \ldots, \lambda_6^{\max}$ are large since the components do not possess smooth edges in the pixelized images. This means that property (11) does not hold for these components, so using the lambda values $\lambda_1, \ldots, \lambda_6$ given in Table 1 does not necessarily give entire feature signal in the output u. We can see from the numerical results depicted in Fig. 2 that we are able to produce output u that contains only those features with scales larger that $1/\lambda_i$ and that leaves, in v, only a small amount of the signal of these features near non-smooth edges. For example, we can see the white boundary of S_2 in v_3 and four white pixels corresponding to the four corners of S_3 in v_4 and v_5. This is due to the nonsmoothness of the boundary and the use of finite differences. However, the numerical results closely match the analytic results given in Subsection 4.1. By forming differences between the outputs $u_1, \ldots, u_6$, we extracted individual features $S_1, \ldots, S_5$ from input f. These results are depicted in the fourth row of images in Fig. 2.

We further illustrate the feature selection capacity of the TV-L^1 model by presenting two real-world applications. The first application [24] is background correction for cDNA microarray images, in which the mRNA-cDNA gene spots are often plagued with the inhomogeneous background that should be removed. Since the gene spots have similar small scales, an appropriate λ can be easied derived from Proposition 1. The results are depicted in Fig. 2 (c)-(f). The second application [9] is illumination removal for face recognition. Fig. 2 (i)-(iii) depicts three face images in which the first two images belong to the same face but were taken under different lighting conditions, and the third image belongs to another face. We decomposed their logarithm using the TV-L^1 model (i.e., $f \overset{\log}{\rightarrow} f' \overset{\text{TV}-L^1}{\longrightarrow} u' + v'$) with $\lambda = 0.8$ and obtained the images (v') depicted in Fig. 2 (iv)-(vi). Clearly, the first two images (Fig. 2 (iv) and (v)) are more correlated than their originals while they are very less correlated to the third. The role of the TV-L^1 model in this application is to extract the small-scale facial objects like the mouth edges, eyes, and eyebrows that are nearly illumination invariant. The processed images should make the subsequent computerized face comparison and recognition easier.

References

1. F. ALIZADEH AND D. GOLDFARB, *Second-order cone programming*, Mathematical Programming, Series B, 95(1), 3–51, 2003.
2. S. ALLINEY, *Digital filters as absolute norm regularizers*, IEEE Trans. on Signal Processing, 40:6, 1548–1562, 1992.
3. S. ALLINEY, *Recursive median filters of increasing order: a variational approach*, IEEE Trans. on Signal Processing, 44:6, 1346–1354, 1996.
4. S. ALLINEY, *A property of the minimum vectors of a regularizing functional defined by means of the absolute norm*, IEEE Trans. on Signal Processing, 45:4, 913–917, 1997.
5. L. AMBROSIO, N. GIGLI, AND G. SAVARÉ, *Gradient flows, in metric spaces and in the space of probability measures*, Birkhäuser, 2005.
6. G. AUBERT AND J.F. AUJOL, *Modeling very oscillating signals. Application to image processing*, Applied Mathematics and Optimization, 51(2), March 2005.

7. M. BURGER, S. OSHER, J. XU, AND G. GILBOA, *Nonlinear inverse scale space methods for image restoration*, UCLA CAM Report, 05-34, 2005.
8. T.F. CHAN AND S. ESEDOGLU, *Aspects of total variation regularized L^1 functions approximation*, UCLA CAM Report 04-07, to appear in SIAM J. Appl. Math.
9. T. CHEN, W. YIN, X.S. ZHOU, D. COMANICIU, AND T. HUANG, *Illumination normalization for face recognition and uneven background correction using total variation based image models*, CVPR'05, 2005.
10. D. GOLDFARB AND W. YIN, *Second-order cone programming methods for total variation-based image restoration*, Columbia University CORC Report TR-2004-05.
11. E. GIUSTI, *Minimal surfaces and functions of bounded variation*, Birkhäuser, 1984.
12. A. HADDAD AND Y. MEYER, *Variantional methods in image processing*, UCLA CAM Report 04-52.
13. T. LE AND L. VESE, *Image decomposition using the total variation and div(BMO)*, UCLA CAM Report 04-36.
14. L. LIEU AND L. VESE, *Image restoration and decomposition via bounded total variation and negative Hilbert-Sobolev spaces*, UCLA CAM Report 05-33.
15. Y. MEYER, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, University Lecture Series Volume 22, AMS, 2002.
16. M. NIKOLOVA, *Minimizers of cost-functions involving nonsmooth data-fidelity terms*, SIAM J. Numer. Anal., 40:3, 965–994, 2002.
17. M. NIKOLOVA, *A variational approach to remove outliers and impulse noise*, Journal of Mathematical Imaging and Vision, 20:1-2, 99–120, 2004.
18. M. NIKOLOVA, *Weakly constrained minimization. Application to the estimation of images and signals involving constant regions*, Journal of Mathematical Imaging and Vision 21:2, 155–175, 2004.
19. S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method for total variation-based image restoration*, SIAM J. on Multiscale Modeling and Simulation 4(2), 460–489, 2005.
20. S. OSHER, A. SOLE, AND L.A. VESE, *Image decomposition and restoration using total variation minimization and the H^{-1} norm*, UCLA C.A.M. Report 02-57, (Oct. 2002).
21. L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60, 259–268, 1992.
22. O. SCHERZER, W. YIN, AND S. OSHER, *Slope and G-set characterization of set-Valued functions and applications to non-Differentiable optimization problems*, UCLA CAM Report 05-35.
23. L. VESE AND S. OSHER, *Modelling textures with total variation minimization and oscillating patterns in image processing*, UCLA CAM Report 02-19, (May 2002).
24. W. YIN, T. CHEN, X.S. ZHOU, AND A. CHAKRABORTY, *Background correction for cDNA microarray images using the TV+L^1 model*, Bioinformatics 21(10), 2410–2416, 2005.
25. W. YIN, D. GOLDFARB, AND S. OSHER, *Total variation-based image cartoon-texture decomposition*, Columbia University CORC Report TR-2005-01, UCLA CAM Report 05-27, 2005.

Structure-Texture Decomposition by a TV-Gabor Model

Jean-François Aujol*, Guy Gilboa**, Tony Chan*, and Stanley Osher***

Department of Mathematics, UCLA, Los Angeles, CA 90095, USA

Abstract. This paper explores new aspects of the image decomposition problem using modern variational techniques. We aim at splitting an original image f into two components u and v, where u holds the geometrical information and v holds the textural information. Our aim is to provide the necessary variational tools and suggest the suitable functional spaces to extract specific types of textures.

Our modeling uses the total-variation semi-norm for extracting the structural part and a new tunable norm, presented here for the first time, based on Gabor functions, for the textural part. A way to select the splitting parameter based on the orthogonality of structure and texture is also suggested.

Keywords: Image decomposition, BV, Hilbert space, projection, total-variation, Gabor functions.

1 Introduction

1.1 Motivation

Decomposing an image into meaningful components is an important and challenging inverse problem in image processing. A first range of models are denoising models: in such models, the image is assumed to have been corrupted by noise, and the processing purpose is to remove the noise. This task can be regarded as a decomposition of the image into signal parts and noise parts. Certain assumptions are taken with respect to the signal and noise, such as the piecewise smooth nature of the image, which enables good approximations of the clean original image.

In modern image-processing, two main successful approaches are usually considered to solve the denoising problem. The first one is based on manipulating the wavelet coefficients of the image [12,22,10,21,23]. The second one is based on solving nonlinear partial-differential equations (PDE's) associated with the minimization of an energy composed of some norm of the gradient [30,9,3,23,26,27].

* Jean-François Aujol and Tony Chan acknowledge supports by grants from the NSF under contracts DMS-9973341, ACI-0072112, INT-0072863, the ONR under contract N00014-03-1-0888, the NIH under contract P20 MH65166, and the NIH Roadmap Initiative for Bioinformatics and Computational Biology U54 RR021813 funded by the NCRR, NCBC, and NIGMS.

** Guy Gilboa acknowledges support by the following grants: NIH U54 RR021813, NSF IIS-0326388 (Prime award), NYU F5552-01.

*** Stanley Osher acknowledges support by the following grants: NIH U54 RR021813, NSF IIS-0326388 (Prime award), NYU F5552-01, NSF DMS-0312222, and NSF ACI-0321917.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 85–96, 2005.

A related but different problem, which is the main topic of this paper, is the decomposition of an image into its structural and textural parts [33,28,11,34,4,5,32,17]. The aim of this type of decomposition is harder to formulate explicitly. The general concept is that an image can be regarded as composed of a structural part, corresponding to the main large objects in the image, and a textural part, containing fine scale-details, usually with some periodicity and oscillatory nature.

We aim at splitting an original image f into two components u and v, u containing the geometrical information and v the textural information. Our modeling is based on TV regularization approaches: we minimize a functional with two terms, a first one based on the total variation and a second one on a different norm adapted to the texture component.

Gabor functions, proposed by [14], have been found to be very useful in texture processing applications, e.g. [13,20,35], and to have close relations with the human-visual system [29]. We design a family of Hilbert spaces based on Gabor functions. This provides us with a new TV-Gabor model in which one can take advantage of *a-priori* knowledge of both the frequency and the direction of the textures of interest.

We also attempt to provide a mechanism to select the regularization parameter for decomposition. Following ideas on diffusion stopping time for denoising [25], we suggest to use a selection criterion based on the correlation of the structure and the texture parts.

The paper is organized as follows: in Section 2 the general TV-Hilbert regularization model is explained, supplying the necessary theoretical foundations for the proposed method. In Section 3 we explain the motivation for texture specific kernels and introduce the TV-Gabor model. We address more specific implementation details in Section 4. In Section 5 we propose a way to select the splitting parameter. Numerical examples are shown in Section 6. We conclude with the main contributions of this study in Section 7. Notice that this paper is an abridged version of [7] with only a selected subset of the content of [7].

2 TV-Hilbert Regularization Model

2.1 Discretization

Our discretization assumes that the image is a two dimension vector of size $N \times N$. We denote by X the Euclidean space $\mathbb{R}^{N\times N}$, and $Y = X \times X$. The space X will be endowed with the L^2 inner product $(u,v)_{L^2} = \sum_{1\leq i,j\leq N} u_{i,j}v_{i,j}$ and the norm $\|u\|_{L^2} = \sqrt{(u,u)_{L^2}}$. To define a discrete total variation, we introduce a discrete version of the gradient operator. If $u \in X$, the gradient ∇u is a vector in Y given by: $(\nabla u)_{i,j} = ((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j})$, with

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

and

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}.$$

The discrete total variation of u is then defined by: $J(u) = \sum_{1\leq i,j\leq N} |(\nabla u)_{i,j}|$.

2.2 $\mathcal{H}$ Hilbert Space

In [6], the authors have considered other spaces to model oscillating patterns. They propose to use a general family of Hilbert spaces that we will consider in this paper. These Hilbert spaces are defined thanks to an operator K with the following properties: K is a linear symmetric positive operator from A to L^2, where A is either X_0 or L^2 (where $X_0 = \{x \in X \ / \ \sum_{i,j} x_{i,j} = 0\}$). In the case when $A = X_0$, then we extend K to the whole L^2 by setting $K(x) = +\infty$ if $x \in L^2 \backslash X_0$. Notice that with these assumptions, we can define K^{-1} on $ImK = \{z \in L^2$ such that $\exists x \in A$ with $z = K(x)\}$. If f and g are in X_0, then let us define:

$$\langle f, g \rangle_{\mathcal{H}} = \langle f, Kg \rangle_{L^2} \tag{1}$$

This defines an inner product on $X_0 = \{x \in X \ / \ \sum_{i,j} x_{i,j} = 0\}$. Examples:

1. When $K = Id$, then $\mathcal{H} = L^2$, and (2) is the ROF model [30].
2. When $K = -\Delta$, then $\mathcal{H} = H = \{f \in L^2, \nabla f \in L^2\}$.
3. When $K = -\Delta^{-1}$, then $\mathcal{H} = H^{-1} = (H_0^1)^*$ (see [1] for the definition of H^{-1}), and (2) is the OSV model [28].

2.3 TV-Hilbert Model

The model studied in [6] is the following:

$$\inf_u \left(J(u) + \frac{\lambda}{2} \|f - u\|_{\mathcal{H}}^2 \right) \tag{2}$$

Some mathematical results about this problem are provided in [6] (see also [2] for similar results in the case of image denoising and deblurring). In particular, the existence and uniqueness of a solution for (2) is proved. A modification of Chambolles's projection algorithm [8] is also proposed for computing the solution of problem (2):

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(K^{-1}\mathrm{div}\,(p^n) - \lambda f))_{i,j}}{1 + \tau|(\nabla(K^{-1}\mathrm{div}\,(p^n) - \lambda f))_{i,j}|} \tag{3}$$

where $p^0 = 0$.

Theorem 1. *If* $\tau \leq \frac{1}{8\|K^{-1}\|_{L^2}}$, *then* $\frac{1}{\lambda} K^{-1}\mathrm{div}\, p^n \to \hat{v}$ *as* $n \to \infty$, *and* $f - \frac{1}{\lambda} K^{-1}$ $\mathrm{div}\, p^n \to \hat{u}$ *as* $n \to \infty$, *where* $\hat{u}$ *is the solution of problem (2) and* $\hat{v} = f - \hat{u}$.

In [6], the authors apply their framework to solve the problem of image denoising. Here, we intend to use (2) to carry out frequency and directional adaptive image decomposition. Indeed, by choosing the kernel K in a suitable way, we can emphasize the weight of some frequencies and some directions. To construct the "texture-norm" we use Gabor wavelets.

The projection algorithm proposed in [6] to solve (2) is given by (3). In fact, one needs to use K^{-1} and not K to solve (2) with this algorithm. It is therefore easier to construct K^{-1} (so that K has some good properties, but without computing K explicitly). K needs to be a non negative symmetric linear operator. Here we even assume that K is positive. This implies that K^{-1} is also a symmetric positive linear operator.

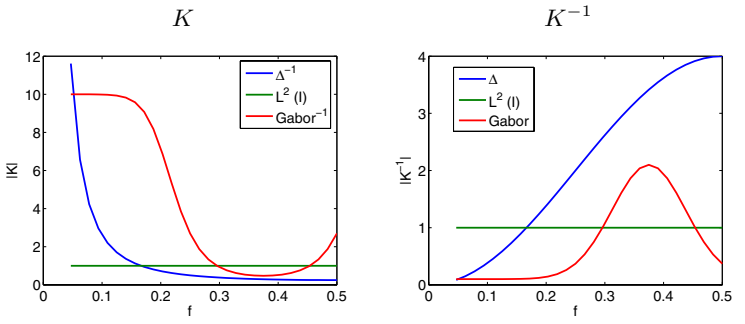


Fig. 1. The kernel K and its inverse K^{-1} for the OSV, ROF and the proposed TV-Gabor model

Remark on a Possible Alternative Construction: K being a positive symmetric operator, there exists a unique positive symmetric linear operator, denoted by $\sqrt{K}$, such that $\sqrt{K}^2 = K$. In particular, we have $\|f-u\|_{\mathcal{H}}^2 = \langle f-u, K(f-u)\rangle_{L^2} = \|\sqrt{K}(f-u)\|_{L^2}^2$. We can then rewrite problem (2) as:

$$\inf_u \left(J(u) + \frac{\lambda}{2}\|\sqrt{K}(f-u)\|_{L^2}^2 \right) \tag{4}$$

In fact, instead of K^{-1}, it also may be interesting to construct $\sqrt{K}^{-1}$. In what follows, we only focus on K^{-1}, but our construction can be applied to $\sqrt{K}^{-1}$ as well.

3 Texture-Specific Kernels

In [6] it was shown that the difference between the OSV model [28] and ROF model [30] could be understood as frequency weighting of the L^2 norm for the H^{-1} fidelity term of OSV. The frequency weighting of the square norm is proportional to $\frac{1}{(2\pi f)^2}$, which corresponds to the Δ^{-1} operator in the frequency domain, see Fig. 1 . The low frequencies are therefore highly penalized in the fidelity term, considerably reducing the eroding effect compared with ROF. This has proved to be an efficient tool for image denoising [28,5]. However, other linear kernels could be used for adaptive frequency algorithms.

In this section we address the problem of designing a family of kernels for image decomposition. The operator K is a convolution operator, therefore K^{-1} in the Fourier domain is simply its inverse. Moreover, K^{-1} is also a convolution operator. We denote by H the associated filter, and in the rest of the section we focus on the designing of this filter.

In the $u+v$ decomposition model K penalizes frequencies that are not considered as part of the texture component. Therefore K^{-1} could be interpreted as the frequencies which should mainly be included in the texture part. A general and simple characterization of textures could be done using Gabor functions. These functions would typically describe the type of textures we would like to extract. Naturally, they apply as good candidates for K^{-1}. As already mentioned, the inverse kernel is actually the one needed

in the numerical implementation. Thus our proposed design strategy is to use Gabor functions for constructing the inverse kernel. Notice that other design methods could be used. We use the function:

$$g(x) = \cos(2\pi\nu x)\,\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-x^2}{2\sigma^2}\right) \tag{5}$$

This gives the following values for the filter H:

$$h_k = \cos(2\pi\nu k)\,\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-k^2}{2\sigma^2}\right) \tag{6}$$

$\nu \in (0, 0.5]$ is the frequency of the texture. σ is related to the width of the band-pass around this frequency. A small σ in the spatial domain means a wide band-pass in the frequency domain. If we know the frequency of the texture we want to get, it is then interesting to use a large σ (which means a small band-pass in the frequency domain). Note that some restrictions apply for choosing σ, see Lemma 2 in Section 4. Actually, σ cannot be very large, which may be interpreted as a form of an uncertainty principle.

(5) is a one dimensional filter. There are a few methods to then design a two dimensional filter. One possibility is to consider the product $g(x)g(y)$. We will analyze this possibility later. Another choice to construct our filter H is to use rotationally invariant Gabor wavelets as:

$$g(x,y) = \cos\left(2\pi\nu\sqrt{x^2+y^2}\right)\frac{1}{\sqrt{2\pi}\sigma^2}\exp\left(\frac{-x^2-y^2}{2\sigma^2}\right) \tag{7}$$

Such a choice will give better numerical results when the texture is known to be rotationally symmetric.

Directions: Many textures are not rotationally symmetric. It is therefore interesting to add this direction information in our filter H. To do so, we just need to use a $1D$ filter as (5), and then rotate it so that it fits the direction of the texture. A possible improvement is to use an ellipse (see [13] for instance).

We propose a way to construct a 2D kernel K^{-1} (in fact of the associated filter H) out of a 1D filter:

$$H_x = \left(h_{\frac{d-1}{2}}, \ldots, h_1, h_0, h_1, \ldots, h_{\frac{d-1}{2}}\right) \tag{8}$$

where d is the dimension of the filter H_x, and h_k is given by (6). Since K^{-1} is symmetric, we also choose H_x to be symmetric. We then set $H = H_x * H_y$, where H stands for the filter associated to K^{-1}, $*$ denotes convolution, and $H_y = H_x^T$, where T stands for transpose. Remark: for the convolution, we consider periodic boundary conditions.

4 Eigenvalues

In this section, we compute the eigenvalues of K^{-1}, and give a sufficient condition so that they are positive.

The filter H associated with K^{-1} should define a linear symmetric positive operator. By construction, H defines a linear symmetric operator. But as we will see, we have

to impose some conditions on the values h_k of the filter so that it is positive. We recall that a linear symmetric operator is positive if and only if its eigenvalues are positive (this can even be taken as a definition). To get the positivity for H, we are therefore lead to compute its associated eigenvalues (the ones of the associated linear mapping). Since we have constructed H out of two 1-D filters, we are in fact interested in the eigenvalues of these filters (since they will give us the eigenvalues of K^{-1}). Since K^{-1} is positive, we also impose the constraint that H_x is positive.

The filtering of an image of size $N \times M$ by H_x corresponds to a linear mapping from $\mathbb{R}^{NM}$ to $\mathbb{R}^{NM}$ (this is the reason why we speak of the eigenvalues of the filter H, which are in fact the eigenvalues of the corresponding linear mapping). Let us denote by A_x (resp A_y) the matrix of size $(NM)^2$ associated to H_x (resp H_y). An image I is a matrix $(I_{i,j})$, with $1 \leq i \leq N$ and $1 \leq j \leq M$. We rewrite it as a 1 Dimensional vector I_k, with $1 \leq k \leq NM$, using $I_k = I_{i,j}$ if $k = M(i-1)+j$.

Since A_x and A_y have a very particular form (they are both circulant matrices), we can compute the exact values of their eigenvalues, as stated by the following result:

Proposition 1. *The eigenvalues of A_x are: $\left\{ h_0 + 2\sum_{k=1}^{\frac{d-1}{2}} h_k \cos\left(\frac{2\pi qk}{M}\right),\ 0 \leq q \leq \frac{M}{2} \right\}$ and the ones of A_y are: $\left\{ h_0 + 2\sum_{k=1}^{\frac{d-1}{2}} h_k \cos\left(\frac{2\pi qk}{N}\right),\ 0 \leq q \leq \frac{N}{2} \right\}$.*

Proof. The proof is just a consequence of the fact that A_x and A_y are circulant matrix. We refer the interested reader to [7] for the details.

Now that we have computed the eigenvalues of A_x and A_y, we can get the ones of K^{-1}. Since A_x and A_y commute, the eigenvalues of K^{-1} are contained in the set:

$$\left\{ P_1(\omega_M^p) P_2(\omega_N^q),\ 0 \leq q \leq \frac{M}{2},\ 0 \leq q \leq \frac{N}{2} \right\} \tag{9}$$

Since the eigenvalues of A_x and A_y are positive, so are the ones of K^{-1}. If we denote by $\gamma^x_{\min}$ (resp γ^y_{min}) the smallest eigenvalue of A_x (resp A_y) and by $\gamma^x_{\max}$ (resp $\gamma^y_{\max}$) the largest eigenvalue of A_x (resp A_y), then, if γ is an eigenvalue of K^{-1}, we have: $\gamma^x_{min}\gamma^y_{min} \leq \gamma \leq \gamma^x_{\max}\gamma^y_{\max}$. From this last point, we deduce in particular that

$$\|K^{-1}\|_{L^2} \leq \gamma^x_{\max}\gamma^y_{\max} \tag{10}$$

Lemma 1. *If we choose $\tau \leq \frac{1}{8\gamma^x_{\max}\gamma^y_{\max}}$ in algorithm (3), then the algorithm converges.*

Proof. This a direct consequence of (10) and of Theorem 1.

Unfortunately, the eigenvalues of K^{-1} can be negative. The next lemma gives a sufficient condition for the eigenvalues of K^{-1} to be positive.

Lemma 2. *If $h_0 \geq 2\sum_{k=1}^{\frac{d-1}{2}} |h_k|$ then the eigenvalues of A_x, A_y and K^{-1} are positive.*

Proof. This is a consequence of Proposition 1 and of (9).

Notice that the above condition for h_0 is only a sufficient condition. The eigenvalues can still be positive in less restrictive cases, and can be computed explicitly for the designed kernel (see Proposition 1).

By using Lemma 2 and the explicit values of h_k given by (6), we can derive more explicit sufficient conditions about the positivity of the eigenvalues of K^{-1}. In particular, we can show that if σ is small enough, then the eigenvalues of H are positive, see more details in [7].

5 Parameter Selection

In this section, we suggest a method to select the weight parameter for a proper decomposition of an image. The authors are not aware of any other suggested method on how to choose the value of λ for decomposition. Therefore we first discuss shortly the solutions at present that are used for denoising and explain the difficulties that arise in decomposition.

For the denoising problem, one often assumes that the variance of the noise σ^2 is known *a-priori* or can be well estimated from the image. As the v part in the denoising case should contain mostly noise, a natural condition is to select λ such that the variance of v is equal to that of the noise, that is $\text{var}(v) = \sigma^2$. Such a method was used in [30] in the constrained ROF model, and this principle dates back to Morozov [24] in regularization theory. A modern approach, suggested recently in [15,18], is to try to optimize a criterion, such as the Signal-to-Noise Ratio (SNR). It was shown that this method can achieve better results than the constrained formulation, in terms of SNR and visually. This method also relies on an estimation of the noise variance.

Both of the above approaches cannot be applied for finding λ in decomposition. Here we do not know of a good way to estimate the texture variance, also there is no performance criterion like the SNR, which can be optimized. Therefore we should resort to a different approach.

Our approach follows the work of Mrazek-Navara [25], used for finding the stopping time for denoising with nonlinear diffusions. The method relies on a correlation criterion and assumes no knowledge of noise variance. As shown in [15], its performance is inferior to the SNR-based method of [15] and to an analogue of the variance condition for diffusions. For decomposition, however, the approach of [25], adopted for the variational framework, may be a good basic way for the selection of λ.

In this paper the general decomposition framework is of the form:

$$E_{Structure}(u) + \lambda E_{Texture}(v), \quad f = u + v, \tag{11}$$

where u and v minimize the above total energy. Our goal is to find the right balance between the energy terms, or the value of λ, which produces a meaningful structure-texture decomposition.

Let us define first the (empirical) notions of mean, variance and covariance: the mean is $\bar{q} \doteq \frac{1}{|\Omega|}\int_\Omega q d\Omega$, the variance is $V(q) \doteq \frac{1}{|\Omega|}\int_\Omega (q-\bar{q})^2 d\Omega$, and the covariance is $\text{cov}(q,r) \doteq \frac{1}{|\Omega|}\int_\Omega (q-\bar{q})(r-\bar{r}) d\Omega$. We would like to have a measure that defines orthogonality between two signals and is not biased by the magnitude (or variance) of

the signals. A standard measure in statistics is the correlation, which is the covariance normalized by the standard deviations of each signal:

$$\mathrm{corr}(q,r) \doteq \frac{\mathrm{cov}(q,r)}{\sqrt{V(q)V(r)}}.$$

By the Cauchy-Schwarz inequality it is not hard to see that $\mathrm{cov}(q,r) \leq \sqrt{V(q)V(r)}$ and therefore $|\mathrm{corr}(q,r)| \leq 1$. When the correlation is 0 we refer to the two signals as *not correlated*. This is a necessary condition (but not a sufficient one) for statistical independence. It often implies that the signals can be viewed as produced by different "generators" or models.

To guide the parameter selection of a decomposition we use the following assumption:

Assumption: The texture and the structure components of an image are not correlated. This assumption can be relaxed by stating that the magnitude of the correlation of the components is very low. Let us define the pair (u_λ, v_λ) as the one minimizing (11) for a specific λ.

Following the above assumption, to find a suitable parameter λ, we are led to consider the following problem:

$$\lambda^* = \mathrm{argmin}_\lambda \left(|\mathrm{corr}(u_\lambda, v_\lambda)| \right). \tag{12}$$

In practice, one generates a scale-space using the parameter λ (in our formulation, smaller λ means more smoothing of u) and selects the parameter λ^* as the first local minimum of the correlation function between the structural part u and the oscillating part v. See also [15,16,18,17,25,6] for related approaches.

This selection method can be very effective in simple cases with very clear distinction between texture and structure. In these cases $\mathrm{corr}(u,v)$ behaves smoothly, reaches a minimum approximately at the point where the texture is completely smoothed out from u, and then increases, as more of the structure gets into the v part (see Fig. 2).

For more complicated images, there are textures and structures of different scales and the distinction between them is not obvious. In terms of correlation, there is no more a single minimum and the function may oscillate.

As a first approximation of a decomposition with a single scalar parameter, we suggest to choose λ after the first local minimum of the correlation is reached. At this stage we cannot claim a fully automatic mechanism for the parameter selection that always works, but rather a highly relevant measurement that should be taken into consideration in future development of automatic decompositions.

6 Numerical Results

We show some numerical results obtained with the new TV-Gabor model on Figures 2 to 4.

In Figure 2, the texture is a periodic signal of frequency $1/\pi \approx 0.32$. In this case we use a rotationally symmetric Gabor function of frequency 0.25 and $\sigma = 1$ (no directional knowledge is incorporated). As expected, the decomposition in this case is very

good. In the next two examples we focus on the ability of the model to have directional selectivity of the textural part, a main feature that clearly distinguishes the TV-Gabor model from the previous ones. In case the textural directions are not known beforehand, we suggest to select them by the dominant peaks in the Fourier domain in medium and high frequencies. This can give basic but sufficient information for designing the kernel (choosing frequency and preferred direction). The Fourier transforms of the input images are shown on the top row, second column of Figures 3 and 4. In Figure 3 the original image is composed of two types of textures and a synthetic structural part. We would like to extract the periodic texture in the ellipses, and not the small squares on the top right. This type of selectivity is hard, but is achieved quite well. Edges of the structural part are kept sharp, and clearly outperforms any linear kernel that would be designed to achieve a similar goal. Compared to $TV - L^2$ (Fig. 3, bottom right) one observes that both textures are mostly in the v part. Also there is some more erosion of the structure (seen in the brighter triangle in the v component) and some "left-overs" of the ellipses-texture in the u part. The comparison was made such that both v parts of TV-Gabor and $TV - L^2$ have the same L^2 norm.

In Figures 4 we show another example of directional decomposition of part of a Dollar note image. In this case, we use the directional TV-Gabor model in the y direction to capture the forehead textures. For comparison, we also display the result with the standard $TV - L^2$ model. As the textures are quite fine with low contrast, we show a contrast enhanced version of v, by multiplying the v part by 4. Again here, both v components have the same L^2 norm. One clearly sees the high directional selectivity of the TV-Gabor model on the left, versus the non-selectiveness of $TV - L^2$.

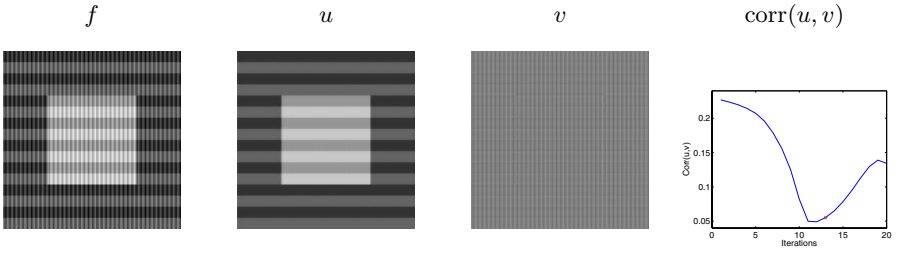


Fig. 2. Decomposition of a simple image by TV-Gabor

7 Conclusion

In this paper, we presented a new general variational model for the image decomposition problem. Given an original image f, the objective is to split the image into two components, u containing the geometrical information and v the textural information.

We introduced a TV-Gabor model which leads us to adaptive frequency and directional image decomposition. Our modeling is based on minimizing a functional with two terms, the first one is the total variation semi-norm and the second one is a Hilbert-space norm adapted to the texture component of the image. In the case when we have some additional information about the texture, then we can take advantage of it by

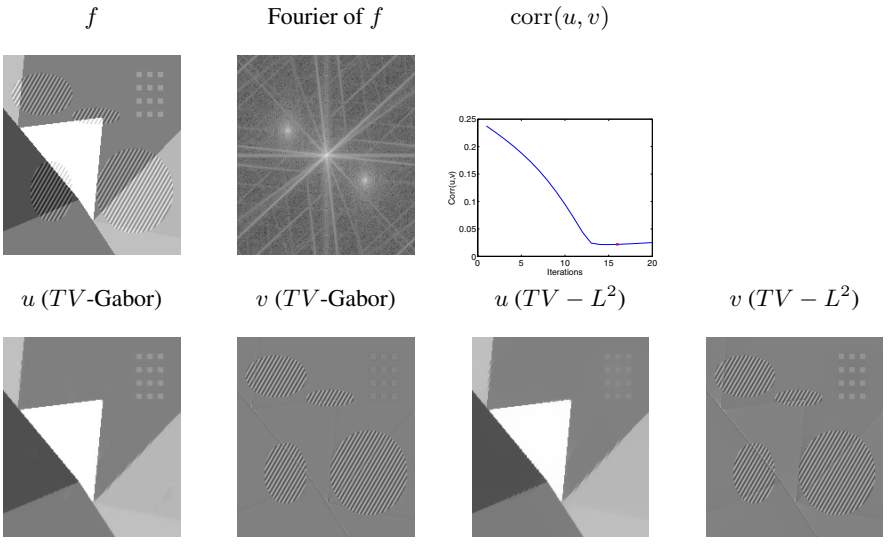


Fig. 3. u, v components of the decomposition of a synthetic image with textures of specific frequency and orientation by TV-Gabor and $TV - L^2$. The TV-Gabor can be more selective and reduce the inclusion in v of undesired textures / small-structures like the small blocks on the top right. Also erosion of large structures is reduced (more apparent in the brighter triangle).

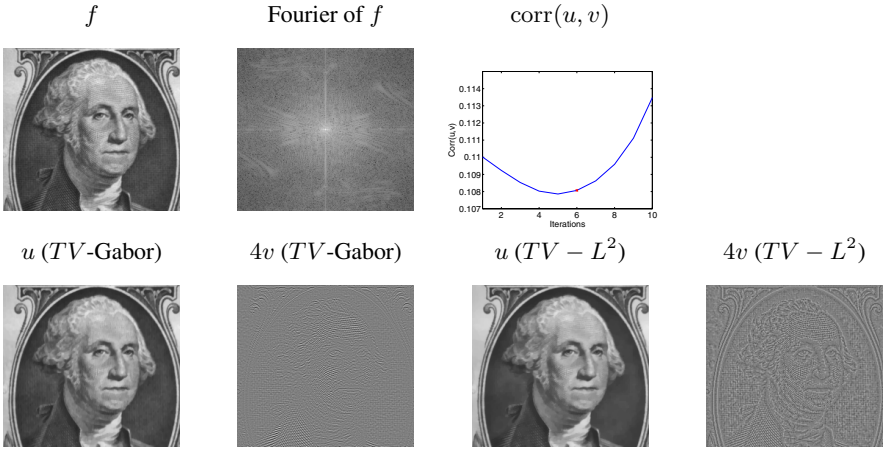


Fig. 4. Decomposition of a Dollar note image by TV-Gabor in the y direction and by TV-L^2. For better visualization, the v part is multiplied by 4.

incorporating this knowledge in the functional. We have designed and studied the corresponding filters, and we have illustrated this new approach with numerical examples.

In this paper we presented a way to design simple texture-specific filters based on Gabor functions. Other, more sophisticated methods could be incorporated to this

framework, such as ones based on wavelets [31]. In future works we intend to explore these issues. Notice that a straightforward extension of the new TV-Gabor model to multiple selected directions, is to use the linearity of the Hilbert fitting term and simply add several directional kernels.

In addition, a way to select the value of λ, the weight parameter between the two norms, was suggested. This is based on a natural orthogonality assumption between the structure and the texture parts.

An important future generalization for the $u + v$ decomposition is to consider a multi-scale approach, as done e.g. in [32,17,16,27,19].

References

1. R. Adams. *Sobolev Spaces*. Pure and applied Mathematics. Academic Press, Inc, 1975.
2. A. Almansa, V. Caselles, G. Haro, and B. Rouge. Restoration and zoom of irregularly sampled, blurred and noisy images by accurate total variation minimization with local constraints, 2004. submitted.
3. G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.
4. J.F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *Journal of Mathematical Imaging and Vision*, 22(1):71–88, January 2005.
5. J.F. Aujol and A. Chambolle. Dual norms and image decomposition models. *International Journal on Computer Vision*, 63(1):85–104, June 2005.
6. J.F. Aujol and G. Gilboa. Implementation and parameter selection for BV-Hilbert space regularizations, 2004. UCLA CAM Report 04-66.
7. J.F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition – modeling, algorithms, and parameter selection, 2005. UCLA CAM Report 05-10, ftp://ftp.math.ucla.edu/pub/camreport/cam05-10.pdf.
8. A. Chambolle. An algorithm for total variation minimization and applications. *JMIV*, 20:89–97, 2004.
9. A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(3):167–188, 1997.
10. A. Chambolle, R.A. De Vore, N. Lee, and B.J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transcations on Image Processing*, 7(3):319–335, March 1998.
11. T. Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation, 2004. CAM report 04-07, to appear in SIAM Journal on Applied Mathematics.
12. D.L. Donoho and M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, December 1995.
13. D. Dunn and W.E. Higgins. Optimal Gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, 4(7):947–964, July 1995.
14. D. Gabor. Theory of communication. *J. Inst. of Electrical Engineering*, 93(3):429–457, 1946.
15. G. Gilboa, N. Sochen, and Y.Y. Zeevi. Estimation of optimal PDE-based denoising in the SNR sense. To appear in IEEE Trans. Image Processing, see http://www.math.ucla.edu/~gilboa/.
16. G. Gilboa, N. Sochen, and Y.Y. Zeevi. Variational denoising of partly-textured images by spatially varying constraints. submitted.

17. G. Gilboa, N. Sochen, and Y.Y. Zeevi. Texture preserving variational denoising using an adaptive fidelity term. In *Proc. VLSM 2003, Nice, France*, pages 137–144, 2003.
18. G. Gilboa, N. Sochen, and Y.Y. Zeevi. Estimation of the optimal variational parameter via SNR analysis. In *Scale-Space '05*, volume 3459 of *Lecture Notes in Computer Science*, pages 230–241, April 2005.
19. C. Groetsch and O. Scherzer. Inverse scale space theory for inverse problems. In *Scale-Space '01*, volume 2106 of *Lecture Notes in Computer Science*, pages 317–325, 2001.
20. A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
21. F. Malgouyres. Minimizing the total variation under a general convex constraint for image restoration. *IEEE transactions on image processing*, 11(12):1450–1456, December 2002.
22. S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
23. Yves Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations, March 2001. The Fifteenth Dean Jacquelines B. Lewis Memorial Lectures.
24. V. A. Morosov. On the solution of functional equations by the method of regularization. *Soviet Math. Dokl.*, 7:414–417, 1966.
25. P. Mrázek and M. Navara. Selection of optimal stopping time for nonlinear diffusion filtering. *IJCV*, 52(2/3):189–203, 2003.
26. M. Nikolova. A variational approach to remove outliers and impulse noise. *JMIV*, 20(1-2):99–120, 2004.
27. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation based image restoration, 2004. CAM report 04-13, to appear in SIAM Journal on Multiscale Modeling and Simulation.
28. S.J. Osher, A. Sole, and L.A. Vese. Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 1(3):349–370, 2003.
29. M. Porat and Y.Y. Zeevi. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):452–468, July 1988.
30. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
31. J.L. Starck, M. Elad, and D.L. Donoho. Image decomposition: separation of texture from piecewise smooth content, 2003. To appear in IEEE Transactions on Image Processing.
32. E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (BV, L^2) decompositions. *SIAM Journal on Multiscale Modeling and Simulation*, 2(4):554–579, 2004.
33. L.A. Vese and S.J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19:553–572, 2003.
34. W. Yin, D. Goldfarb, and S. Osher. Total variation based image cartoon-texture decomposition, 2005. UCLA CAM Report 05-27, ftp://ftp.math.ucla.edu/pub/camreport/cam05-27.pdf.
35. M. Zibulski and Y. Y. Zeevi. Analysis of multi-window Gabor-type schemes by frame methods. *J. Appl. Comp. Harmon. Anal.*, 4(2):188–221, 1997.

From Inpainting to Active Contours

François Lauze and Mads Nielsen

The IT University of Copenhagen,
Departement of Innovation,
Rued Langgaardsvej 7, 2300 Kbh S, Denmark
{francois, malte}@itu.dk

Abstract. We introduce a novel type of region based active contour using image inpainting. Usual region based active contours assume that the image is divided into several semantically meaningful regions and attempt to differentiate them through recovering dynamically statistical optimal parameters for each region. In case when perceptually distinct regions have similar intensity distributions, the methods mentioned above fail. In this work, we formulate the problem as optimizing a "background disocclusion" criterion, a disocclusion that can be performed by inpainting. We look especially at a family of inpainting formulations that includes the Chan and Shen Total Variation Inpainting (more precisely a regularization of it). In this case, the optimization leads formally to a coupled contour evolution equation, an inpainting equation, as well as a linear PDE depending on the inpainting. The contour evolution is implemented in the framework of level sets. Finally, the proposed method is validated on various examples.

1 Introduction: Some Challenging Problems

The two following pictures represent first: the flag of the Greenland territory and second: an object added on a smooth background via addition of the intensities, similar to the presence of transparent layers, that we will call "pseudo calcification" in the sequel. A typical case would indeed be the one of an X-ray showing

a calcifit deposit on some soft tissue. Contour based active contour methods can provide a segmentation of the circle, although sensitive to the initialization, region based active contours using statistical parameters of intensity/color distributions for each regions, will fail to isolate the circle in the Greenland's flag from the rest of the image, since the two regions have the same statistics! In the pseudocalcification image the difficulty comes also from hardly visible edge

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 97–108, 2005.

information, and the reality is often much worse than this artificial image, while finding discriminating statistical invariants may also prove very difficult.

In an attempt to overcome these problems, we introduce a novel methodology, within the framework of active contours. This belongs to the region based family of active contours, but instead of considering that the image domain is partitioned in two (or several regions), our approach consists in considering the following problem: if the region $\tilde{\Omega}$ containing the object of interest is known, and if we have enough prior information on the type of background, can we reconstruct the occluded background image? If we assume a positive answer to that question, then we ask whether a given region Ω contains an object occluding the background in the following way: if Ω delimits indeed the area occupied by a foreground object, then there should be sufficiently large difference between the observed image and the reconstructed one *within this region*. This leads naturally to a variational problem: if Ω is a given region of the image plane D, u_0 the observed image, $u(\Omega) = \mathcal{I}(u_0, \Omega)$ the reconstructed background and if we call $J(\Omega)$ this background/foreground difference, we may look for the true region $\tilde{\Omega}$ as an extremum for $J(\Omega)$. The goal of this work is to propose such a measure, first in general terms, and then a specific instanciation based on a relatively simple variational inpainting formulation, which essentially corresponds to the TV inpainting of Chan and Shen [8] and a simple measure for discrepancy between two images, based on pixel value differences. From it, we derive a corresponding active contour algorithm.

The paper is organized as follows. In section 2, we give a short review of active contour algorithms. In section 3, we introduce an "disocclusion quality measure", compute its Gâteaux derivative and deduce our inpainting based active contour algorithm from it. Several experiments are presented in section 4 and we conclude in section 5.

2 A Short Review of Active Contours

Since the seminal paper of Kass, Witkin and Terzopoulos [15], active contours, or snakes, have been used rather extensively in computer vision and medical imaging, for the purpose of segmentation, in order to overcome the locality problem of edge detectors. They are curves with built in regularity properties and preferences for edges in an image. Although very simple to implement, they suffer among other of initialization problems and necessitates reparameterization. In order to overcome initialization problems several solutions were proposed, including balloon forces [10], gradient vector flows [25]. Geodesic active contours, proposed independently by [7] and [16] introduced a parameterization independent formulation. All these models deal only with contours, not with the regions they separate. Based on a simplification of the Mumford-Shah segmentation functional [18], Chan and Vese proposed a region based algorithm in [9], that leads to a contour evolution coupled with the estimation of the mean values in the regions delimited by this contour. More complex statistical descriptors have been proposed instead of the mean, as histogram matching in [2]. In a series

of papers, Paragios and Deriche proposed a new paradigm called Geodesic Active Regions where both contour based and region based terms are used (see for instance [20]). Many declinations on these ideas have been proposed in order to tackle the variety of situation encontered in Computer Vision and Medical Imaging. We note nevertheless that in region based active contours, the different regions one want to recover form, together with their boundaries, a partition of the image domain, and that the respective contents of these regions are usually assumed to be independent of each others. While this assumption is sufficient in many applications, the images shown in the previous section show that it may fail.

3 Inpainting Based Active Contours

In this section we introduce background disocclusion ideas and a corresponding variational formulation that will lead to an active contour evolution equation.

Let D be the image domain, u_b the background image defined on D. Let $\tilde{\Omega} \subset D$ an unknown region and u_f the foreground image with support $\tilde{\Omega}$. We suppose that the observed image u_0 is obtained through a combination operation

$$u_0 = \mathcal{C}(u_b, u_f).$$

Our basic problem is to be able to determine, from u_0, the region $\tilde{\Omega}$ and information on u_b and u_f. We now assume that if $\tilde{\Omega}$ is known, we can confidently estimate u_b using an *inpainting* or *disocclusion* operation $\overline{u}_b = \mathcal{I}(u_0, \tilde{\Omega})$. We make the two following assumptions:

1. u_0 and $\overline{u}_b$ should differ significantly inside the foreground region $\tilde{\Omega}$
2. the foreground region should be relatively small.

3.1 Variational Formulation

These two assumptions are used to form a "disocclusion quality measure", that attributes a numerical score to a given region $\Omega \subset D$ belonging to an admissible set $\mathcal{P}$ of regions of D,

$$J(\Omega) = \int_{\Omega} L\left(\mathcal{I}(u_0, \Omega), u_0, \Omega\right) dx \tag{1}$$

where the "Lagrangian" $L : \mathbb{R} \times \mathbb{R} \times \mathcal{P} \to \mathbb{R}, (x, y, A) \mapsto L(x, y, A)$ incorporates measures of the difference between x and y as well as measure of the size of its last argument.

The true region $\tilde{\Omega}$ should be an extremal point of $J(\Omega)$ and we therefore search for a necessary condition for the functional $J(\Omega)$ to achieve an extremum. In the sequel of this work, we consider the case of noisy additive signals $u_0 = u_b + u_f + \eta$, η being some noise, as it would be the case e.g. for X-rays or reflection.

With this additive model, an elementary way to measure the discrepancy between u_0 and $\overline{u}_b$ in the region of inpainting consists in summing up a function

of their pixelwise intensity differences, while a measure of the region can be given by its area, or a function of it. A simple form for the function L incorpoarting these measures is

$$L(x, y, A) = \frac{|x - y|^q}{|A|^p} \tag{2}$$

with $|A|$ denoting the area of the set A, and $p > 0$, $q = 1$ or 2, and we therefore want to *maximize* $J(\Omega)$. Another choice that we considered has been[1]

$$L(x, y, A) = \pm \frac{x - y}{|A|^p}. \tag{3}$$

In both these situations, the resulting $J(\Omega)$ can be seen as a "generalized expectation" of $u_0 - \overline{u}_b$ on Ω, an expectation that we want to maximize.

We follow Aubert et al. in [2] for the computation of $J'(\Omega)$ using shape derivative tools. Let Γ the boundary of Ω, $\mathbf{N}$ the inward normal to Γ, the Gâteaux derivative of $J(\Omega)$ in the direction of $\mathbf{V}$ is, using the chain rule,

$$\begin{aligned}\langle J(\Omega)', \mathbf{V}\rangle = &\int_\Omega L_x\left(\mathcal{I}(u_0, \Omega), u_0, \Omega\right) \mathcal{I}_s(u_0, \Omega, \mathbf{V})\, dx \\ &+ \int_\Omega L_s\left(\mathcal{I}(u_0, \Omega), u_0, \Omega, \mathbf{V})\right) dx - \int_\Gamma L\left(\mathcal{I}(u_0, \Omega), u_0, \Omega\right)(\mathbf{V} \cdot \mathbf{N})\, da(x). \end{aligned} \tag{4}$$

In order to obtain a contour evolution equation, we need to explicit the first term, involving the shape derivative $\mathcal{I}_s(u_0, \Omega, \mathbf{V})$ of the inpainting with respect to the inpainting domain, as well as the second term, involving the shape derivative of the Lagrangian $L(x, y, A)$. We will introduce the inpainting operation and compute the shape derivative related term in the next subsection and in subsection 3.3 we will compute the shape derivative of the Lagrangian term and the resulting curve evolution equation.

3.2 Inpainting and Related Shape Derivative

In this work, we have focused on a simple inpainting formulation: $u = \mathcal{I}(u_0, \Omega)$ is defined as

$$u = \arg.\min_v \mathcal{F}(v, u_0, \Omega)$$

with

$$\mathcal{F}(v, u_0, \Omega) = \frac{\lambda}{2} \int_{D \setminus \Omega} (v - u_0)^2 + \int_D \frac{1}{r} \langle |\nabla v| \rangle_\varepsilon^r \, dx \tag{5}$$

where $\langle t \rangle_\varepsilon = \sqrt{t^2 + \varepsilon^2}$ is sometimes referred to as the "Japanese bracket" and $1 \leq r \leq 2$. For $r = 1$, $\varepsilon = 0$, this corresponds to the Total Variation inpainting of Chan and Shen [8]. We will assume nevertheless that $\varepsilon > 0$ when $1 \leq r < 2$,

[1] Assumptions and rationale behind this choice are the following: we know a priori that the sign of the difference $u_b - u_f$ is constant on $\tilde{\Omega}$. In that situation, if the noise η is white, then it should be averaged out in $J(\Omega)$.

this makes the solution of the minimization unique (standard arguments show the uniqueness of the solution in the case $r = 2$ and a formal justification in our settings for r and ε will be presented below) and allows to define the inpainting operator $(u_0, \Omega) \mapsto \mathcal{I}(u_0, \Omega)$. This calculation leads, at least formally, to an computable form for $u_s(\mathbf{V}) := \mathcal{I}_s(u_0, \Omega, \mathbf{V})$.

One must note that, in choosing such a simple inpainting, we implicitly assume a high level of regularity for the partially unknown background, a potentially important restriction for the domain of validity of the final algorithm.

We now perform the derivation of the inpainting related shape derivative. If v is a test function on D, define $f(\tau) = \mathcal{F}(u + \tau v)$. Since u is assumed to be a minimizer for $\mathcal{F}$, the first variation of $\mathcal{F}$ in the direction v, $f'(0)$, must vanish and it provides the optimality condition

$$0 = \int_D \left(\lambda \chi_{D\setminus\Omega}(u - u_0)v + \langle|\nabla u|\rangle_\varepsilon^{r-2} \nabla u \cdot \nabla v\right) dx \tag{6}$$

$$= \int_D \underbrace{\left(\lambda \chi_{D\setminus\Omega}(u - u_0) - \operatorname{div}\left(\langle|\nabla u|\rangle_\varepsilon^{r-2} \nabla u\right)\right)}_{\mathcal{F}_u(u, u_0, \Omega)} v \, dx \tag{7}$$

$$= \langle \mathcal{F}_u(u, u_0, \Omega), v \rangle \tag{8}$$

($\chi_{D\setminus\Omega}$ denotes the characteristic function of $D\setminus\Omega$). The inpainting PDE is the Euler-Lagrange equation

$$\mathcal{F}_u(u, u_0, \Omega) = 0, \tag{9}$$

extracted from equation (7), which allows to implicitly define $u = \mathcal{I}(u_0, \Omega)$ by

$$\mathcal{F}_u(\mathcal{I}(u_0, \Omega), u_0, \Omega) = 0.$$

Assuming that $\mathcal{F}_u$ is differentiable with respect to its first and last variables, this expression allows to compute $u_s(\mathbf{V})$ via the implicit function theorem:

$$u_s(\mathbf{V}) = -\left(\mathcal{F}_{uu}(u, u_0, \Omega)\right)^{-1} \mathcal{F}_{us}(u, u_0, \Omega, \mathbf{V}) \tag{10}$$

$\mathcal{F}_{uu}$ denoting the partial differential of $\mathcal{F}_u$ with respect to its first variable and will be explicited below as a linear elliptic operator, while $\mathcal{F}_{us}$ denotes the shape derivative of $\mathcal{F}_u$ and can be computed as

$$\mathcal{F}_{us}(u, u_0, \mathbf{V}) = \lambda(u - u_0)(\mathbf{V} \cdot \mathbf{N})\delta_\Gamma$$

δ_Γ being the Dirac distribution along Γ. The first term of the Gâteaux derivative (4) can be rewritten

$$-\lambda \int_D \chi_\Omega L_x(u, u_0, \Omega) T\left((u - u_0)(\mathbf{V} \cdot \mathbf{N})\delta_\Gamma\right) dx$$

where $u = \mathcal{I}(u_0, \Omega)$, $T = \left(\mathcal{F}_{uu}(u, u_0, \Omega)\right)^{-1}$ and χ_Ω is the characteristic function of Ω. Using the formal adjoint T^* of T, one rewrites this term as

$$-\lambda \int_D T^*\left(\chi_\Omega L_x(u, u_0, \Omega)\right)(u - u_0)(\mathbf{V} \cdot \mathbf{N})\delta_\Gamma dx$$

and finally as the boundary integral

$$-\lambda \int_\Gamma T^* \left(\chi_\Omega L_x(u, u_0, \Omega)\right)(u - u_0)(\mathbf{V} \cdot \mathbf{N}) da(x).$$

We provide now an expression for the operator $T = \mathcal{F}_{uu}^{-1}$. Setting $g(\tau) = \langle \mathcal{F}_u(u + \tau w, u_0, \Omega), v\rangle$ for a test funtion w on D, the operator $\mathcal{F}_{uu}$ is obtained from

$$\langle \mathcal{F}_{uu}(u, u_0, \Omega)(w),\, v\rangle = g'(0).$$

A straightforward computation using for instance expression (6) gives

$$g'(0) = \int_D \left(\chi_{D\setminus\Omega} v\, w + \nabla v^T A[\nabla u]\nabla w\right) dx \tag{11}$$

$$= \int_D \left(\chi_{D\setminus\Omega} w - \operatorname{div}(A[\nabla u]\nabla w)\right) v\, dx \tag{12}$$

the last equality by integration by part and adequate boundary conditions, and where $A[\nabla u]$ is the 2×2 *symmetric* matrix

$$\langle |\nabla u| \rangle_\varepsilon^{r-4} \begin{pmatrix} (r-1)u_x^2 + u_y^2 + \varepsilon^2 & (r-2)u_x u_y \\ (r-2)u_x u_y & (r-1)u_y^2 + u_x^2 + \varepsilon^2 \end{pmatrix} \tag{13}$$

which, under the assumptions we made on r and ε is positive and definite. We can also note that the *quadratic form* $\langle \mathcal{F}_{uu}(u, u_0, \Omega)(v), v\rangle$ is $f''(0)$, the second variation of $\mathcal{F}$ in the direction of v and is positive definite, as it can be seen from expression (11).This means that the inpainting energy is strictly convex, explaining the uniqueness of the minimizer. Expression (12) provides the expression of $\mathcal{F}_{uu}(u, u_0, \Omega)$ as the differential operator

$$\mathcal{F}_{uu}(u, u_0, \Omega)(w) = \lambda \chi_{D\setminus\Omega} w - \operatorname{div}(A[\nabla u]\nabla w)$$

Positive definiteness also implies here that $\mathcal{F}_{uu}$ is formally invertible, and from (11) and the symmetry of $A[\nabla u]$, it is self-adjoint:

$$\langle \mathcal{F}_{uu}(u, u_0, \Omega)(w),\, v\rangle = \langle w,\, \mathcal{F}_{uu}(u, u_0, \Omega)(v)\rangle.$$

The operator $T = \mathcal{F}_{uu}^{-1}$ which is consequently self-adjoint, $T = T^*$, is given by $v \mapsto T(v)$, defined as the unique solution of

$$\lambda \chi_{D\setminus\Omega} w - \operatorname{div}(A[\nabla u]\nabla w) = v. \tag{14}$$

3.3 Curve Evolution and Level Set Formulation

Before being able to write the curve evolution equation, we need to compute now the second term in equation (4). The shape derivative in the direction of $\mathbf{V}$ of the area $|\Omega|$ is given by $-\int_\Gamma (\mathbf{V} \cdot \mathbf{N})\, da(x)$ and with the Lagrangian L being defined by either (3) or (2), one gets

$$\int_\Omega L_s(u, u_0, \Omega, \mathbf{V})\, dx = p \frac{J(\Omega)}{|\Omega|} \int_\Gamma (\mathbf{V} \cdot \mathbf{N})\, da(x).$$

Collecting all these calculations ,we find that the Gâteaux derivative $\langle J(\Omega)', \mathbf{V}\rangle$ is given by $\int_\Gamma F(\mathbf{V}\cdot\mathbf{N})da(x)$ where F is

$$p\frac{J(\Omega)}{|\Omega|} - \lambda(u-u_0)T\left(\chi_\Omega L_x(u,u_0,\Omega)\right) + L(u,u_0,\Omega) \tag{15}$$

and since our goal is to *maximize* the "disocclusion criterion" $J(\Omega)$, this leads to the following evolution equation $\frac{\partial\Gamma}{\partial t} = F\mathbf{N}$. This curve evolution is rewritten into the Osher-Sethian framework of level sets [24] as $\varphi_t = F|\nabla\varphi|$ where φ represents implicitly $\Gamma(t)$ and $\Omega(t)$ via $\Gamma(t) = \{\varphi(\mathbf{x},t) = 0\}$ and $\Omega(t) = \{\varphi(\mathbf{x},t) < 0\}$.

3.4 Numerics

A first order space convex scheme is used to solve this equation. Using the notations of [24], it is

$$\varphi_{ij}^{n+1} = \varphi_{ij}^n + \Delta t\left[\max(F_{ij}^n,0)\nabla^+ + \min(F_{ij}^n,0)\nabla^-\right].$$

This leads to the following algorithm:

1. Choose an original contour Γ^0, compute φ^0 as the signed distance function $\varphi^0(\mathbf{x}) = \mathrm{dist}_{\Gamma^0}(\mathbf{x})$, and $\Omega^0 = \{\mathbf{x}, H(-\varphi^0(\mathbf{x})) = 1\}$, where H is the 1-dimensional Heaviside function
2. for $i = 0$ to $N-1$ or a convergence condition has been fulfilled
   (a) compute the inpainting $u^n = \mathcal{I}(u_0,\Omega^n)$ and $T\left(\chi_\Omega L_x(u^n,u_0,\Omega^n)\right)$ and the disocclusion measure $J(\Omega^n)$.
   (b) compte F^n from the previous calculations
   (c) compute φ^{n+1} withe the above scheme
   (d) extract Γ^{n+1} and Ω^{n+1} from it
   (e) reinitialize φ^{n+1} as a signed distance $\varphi^{n+1}(\mathbf{x}) = \mathrm{dist}_{\Gamma^{n+1}}(\mathbf{x})$.
3. return u^N and Ω^N

The time step Δt is chosen at each iterations as $1/\|F^n\|_\infty$. The numerical scheme we have used for the inpainting $\mathcal{I}(u_0,\Omega^n)$ follows closely the discretization proposed by Chan and Shen in [8]. The computation of $T\left(\chi_\Omega L_x(u^n,u_0,\Omega^n)\right)$ as the solution w of the PDE

$$\lambda\chi_{D\setminus\Omega}w - \mathrm{div}(A[\nabla u^n]\nabla w) = \chi_\Omega L_x(u^n,u_0,\Omega^n) \tag{16}$$

requires some care. Indeed, apart from the trivial case $r = 2$ in (13) where $A[\nabla u^n]$ reduces to the 2×2 identity matrix, the divergence term will normally contain spatial cross-derivatives and thus needs special attention for the discretization. For that purpose we have used the scheme proposed by Rybak in [23], keeping Chan and Shen discretization for the derivatives of u^n. With this, the linear system resulting from (16) is solved using a Gauss-Seidel scheme.

Finally, an important point to mention is that the Heaviside function used is implemented with almost no regularization, as opposed to way it is handled in the level set implementation of most region based algorithms. Too much regularization would biase the inpainting.

4 Experiments

Our experiments have been performed on synthetic as well as real data. They present an increasing level of complexity in the image content. The inpainting used has been the regularized Total Variation one, i.e. $r = 1$ in the energy (5), and the function L being the one given by (2), with $q = 1.0$.

Experiment 1. Four black squares on a white background. The parameters are $\lambda = 1$, $p = 0.25$. This is illustrated in the left part of Fig. 1, where the first row shows a snapshot of the contour evolution at iterations 1, 7, 15 and 30, the second row shows the corresponding domains and the last row shows the inpainting results.

Experiment 2. Dark non convex object on a light background with 30% added Gaussian noise. In this experiment, $\lambda = 0.1$, $p = 0.55$. This is illustrated in the right part of Fig. 1 The first row shows a snapshot of the contour evolution at iterations 1, 10, 20 and 30, the second row shows the corresponding domains and the last one the inpaintings.

Experiment 3. The next experiment uses the same image as experiment 2. It illustrates some stability with respect to the initial contour, as well as the role of the area penalizer exponent p in L and is illustrated in Fig. 2. We ran two experiments, the first with a value of $p = 0.55$, as in the previous case and the second with a value of $p = 0.60$. In both cases, the λ weight in the inpainting was set to 0.1. In the first case, a correct segmentation has been achieved, while in the second, the right part of the horizontal ellipse could not be fully recovered.

Experiment 4. The "pseudo calcification" image. This image is made of a smooth background (a 4th order polynomial in the image coordinates), with an object added (by adding intensity values). While the intensities in background are in the range [0,1], the original intensity of the added object is constant, equal to 0.065. Although we were able to produce some good quality measure using the function L given by (3), it proved difficult to control. So we used the same function as for the other examples, with $p = 0.5$. In order to prevent the inpainter/denoiser to smooth too much the background, we chose $\lambda = 8$ in (5). This experiment is illustrated in Fig. 3. The first row shows the original object and the pseudo calcification. The two next rows show snapshots of contour evolution at different iterations, for two different initializations. The second initialization, while "more creative" is also closer to the true contour and allows for a much faster convergence (260 iterations in the first case, 50 in the second).

Experiment 5. The next experiment presented uses Greenland's flag with added Gaussian noise, with a standard deviation of 50% of the intensity range of the image. We used the function L with $lambda = 0.1$ and $p = 0.85$ this time. Results are shown in Fig. 4, where the evolution of the disocclusion measure is also plotted.

Experiment 6. The last experiment presented here is somewhat an attempt at the impossible. Indeed, using such a simple inpainting method, the algorithm is normally targeted to images with relatively simple content, at least, with a reasonable smoothness assumption. Nevertheless to test the possibilities of our method, we have used a rather complex input image, the third frame from the `Ettlinger-Tor` sequence [2], frequently used in optical flow estimation. The result presented here are certainly of far lesser quality than some that can be

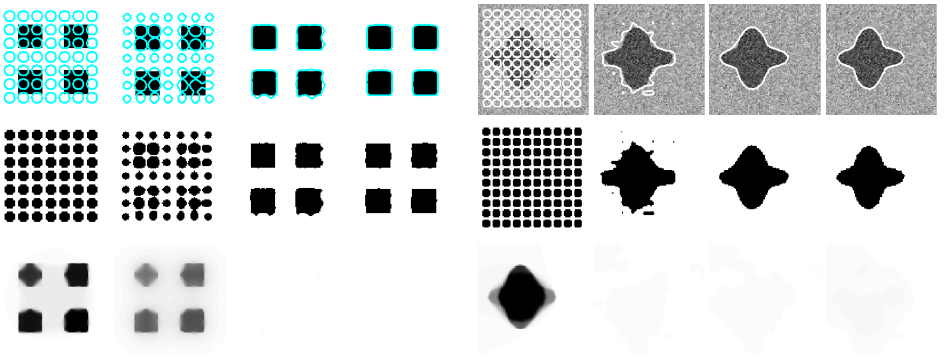


Fig. 1. Experiments 1 and 2. See text.

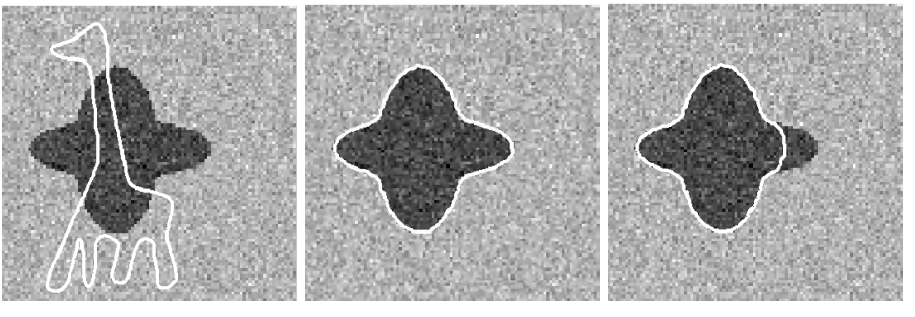


Fig. 2. Experiment 3. See text.

obtained with state of the art algorithms, but show that our method can produce meaningful results, even when the background smoothness assumption is largely violated in this case. The parameters used here where $\lambda = 0.5$ and $p = 2^{-7}$, a very small value. This experiment is illustrated in Fig. 5.

[2] This sequence is copyright (C) 1998 by H. H Nagel, KOGS/IAKS Universität Karlsruhe.

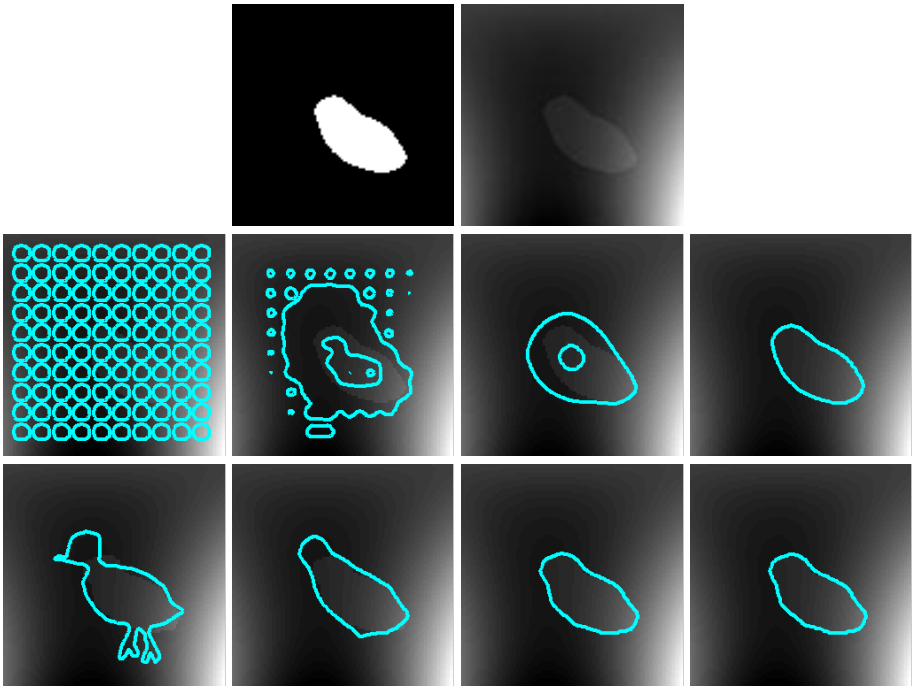


Fig. 3. Experiment 4. See text.

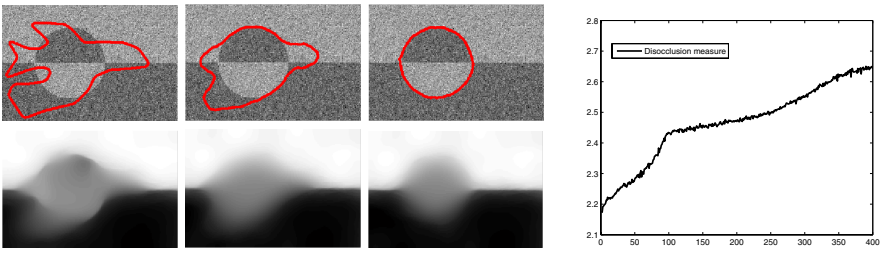


Fig. 4. Experiment 5. See text.

Fig. 5. Experiment 6. See text.

5 Summary and Conclusion

In this paper, we have presented a novel approach for region based active contour segmentation, using the idea of optimal background disocclusion, performed by an inpainting algorithm, which led to a generic variational formulation. We have considered a simple variational algorithm and have derived a contour evolution equation, which has been illustrated on several examples. The resulting algorithm is computationally expensive, since several partial differential equations have to be solved at each iteration. Depending on the size and complexity of the input, it took up to 20mn! There is clearly space here for optimization. In particular, we are working on implementing multigrid solvers for the elliptic equations encountered, multigrid approaches can be extremely efficient, see for instance the recent paper of Bruhn et al. in [5]. Other directions of research are the use of different "Lagrangians" for the disocclusion measures, as well as more sophisticated inpainting algorithms, where we can nevertheless maintain a reasonable efficiency.

References

1. Boston, MA, June 1995. IEEE Computer Society Press.
2. G. Aubert, M. Barlaud, S. Jehan-Besson, and O. Faugeras. Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM Journal of Applied Mathematics*, 63(6):2128–2154, 2003.
3. C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001.
4. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In Kurt Akeley, editor, *Proceedings of the SIGGRAPH*, pages 417–424. ACM Press, ACM SIGGRAPH, Addison Wesley Longman, 2000.
5. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Variational Optical Flow Computation in Real Time. *IEEE Transactions on Image Processing*, 14(5):608–615, May 2005.
6. V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.
7. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision* [1], pages 694–699.
8. T. Chan and J. Shen. Mathematical models for local nontexture inpainting. *SIAM journal of appl. Math*, 62(3):1019–1043, 2002.
9. T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.
10. Laurent D. Cohen and Isaac Cohen. Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.
11. D. Cremers, C. Schnoerr, and J. Weickert. Diffusion-snakes: Combining statistical shape knowledge and image information in a variational framework. In *1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001.
12. Daniel Cremers, Timo Kohlberger, and Christoph Schn orr. Nonlinear Shape Statistics in Mumford Shah Based Segmentation. In *European Conference on Computer Vision*, pages 93–108, 2002.

13. L.C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. Proceedings of the American Mathematical Society, 1998.
14. A. Karlsson and N. Overgaard. Theory for Variational Area-Based Segmentation Using Non-Quadratic Penalty Functions. In *International Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, 2005. To appear.
15. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, London, June 1987.
16. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of the 5th International Conference on Computer Vision* [1], pages 810–815.
17. S. Masnou and J.M. Morel. Level lines based disocclusion. *International Conference on Image Processing*, III:259–263, 1998.
18. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–684, 1989.
19. S. Osher and N. Paragios, editors. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003.
20. N. Paragios and R. Deriche. Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation, Special Issue on Partial Differential Equations in Image Processing, Computer Vision and Computer Graphics*, 13(1/2):249–268, march/june 2002.
21. Mikaël Rousson, Thomas Brox, and Rachid Deriche. Active unsupervised texture segmentation on a diffusion based space. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 699–704, Madison, Wisconsin, USA, June 2003.
22. Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In *European Conference on Computer Vision*, volume 2, pages 78–92, May 2002.
23. I. V. Rybak. Monotone and Conservative Difference Schemes for Elliptic Equations with Mixed Derivatives. *Mathematical Modelling and Analysis*, 9(2):169–178, 2004.
24. J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.
25. Chenyang Xu and Jerry L. Prince. Gradient Vector Flow: A New External Force for Snakes. In *International Conference on Computer Vision and Pattern Recognition*, pages 66–, 1997.

Sobolev Active Contours

Ganesh Sundaramoorthi[1], Anthony Yezzi[1], and Andrea Mennucci[2]

[1] School of Electrical Engineering, Georgia Institute of Technology, Atlanta, USA
[2] Department of Mathematics, Scuola Normale Superiore, Pisa, Italy

Abstract. All previous geometric active contour models that have been formulated as gradient flows of various energies use the same L^2-type inner product to define the notion of gradient. Recent work has shown that this inner product induces a pathological Riemannian metric on the space of smooth curves. However, there are also undesirable features associated with the gradient flows that this inner product induces. In this paper, we reformulate the generic geometric active contour model by redefining the notion of gradient in accordance with Sobolev-type inner products. We call the resulting flows Sobolev active contours. Sobolev metrics induce favorable regularity properties in their gradient flows. In addition, Sobolev active contours favor global translations, but are not restricted to such motions. This is particularly useful in tracking applications. We demonstrate the general methodology by reformulating some standard edge-based and region-based active contour models as Sobolev active contours and show the substantial improvements gained in segmentation and tracking applications.

1 Introduction

Active contours, introduced by Kass *et al.* [1], have been widely used for the segmentation problem. One undesirable feature of Kass's model is that the energy used to derive a flow is dependent on parametrization. Formulations for geometric energies, which do not depend on the particular parametrization of the curve, were later introduced for edge-based active contours [2,3] and region-based active contours [4,5,6]. In order to define the notion of gradient of such energies, an inner product on the set of perturbations of a curve is needed. All of these previous works on geometric active contours use the same geometric L^2-type inner product, which we refer to as H^0, to define a gradient. However, recent work in [7,8] has shown that the Riemannian metric on the space of curves induced by the H^0 inner product is pathological.

In addition to the pathologies of the Riemannian structure induced by H^0, there are also undesirable features of H^0 gradient flows, some of which are listed below.

- First, there are no regularity terms in the definition of the H^0 inner product. That is, there is nothing in the definition of H^0 that discourages flows that are not smooth in the space of curves. By smooth in the spaces of curves, we mean that the surface formed by plotting the evolving curve as a function of time

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 109–120, 2005.

is smooth. Thus, when energies are designed to depend on the image that is to be segmented, the H^0 gradient is very sensitive to noise in the image. As a result, the curve becomes unsmooth instantly. Therefore, in geometric active contours models, a penalty on the curve's length is added to keep the curve smooth in addition to keeping the variational problem well-posed. However, this changes the energy that is being optimized.

- Second, H^0 gradients, evaluated at a particular point on the curve, depend locally on derivatives of the curve. Therefore, as the curve becomes unsmooth, as mentioned above, the derivative estimates become inaccurate, and thus, the curve evolution becomes inaccurate. Moreover, for region-based and edge-based active contours, the H^0 gradient at a particular point on the curve depends locally on image data at the particular point. Although region-based energies may depend on global statistics, such as means, the H^0 gradient still depends on local image data. The H^0 gradient of image dependent energies "encourages" points on the evolving curve to move "independently" to decrease energy rather than encouraging the points to move collectively. This restricts the gradient at a particular point from "seeing" information located at other points of the curve, which implies sensitivity to noise and local features.
- Third, all geometric active contours require that the unit normal to the evolving curve be defined. As such, the evolution does not make sense for polygons. Moreover, since in general, a H^0 active contour does not remain smooth, one needs viscosity theory to define the flow.
- Fourth, if the energy depends on n derivatives of the curve, then the H^0 gradient has $2n$ derivatives of the curve. Since the corresponding level set flows with higher than two derivatives are not known to have a maximum principle, level set methods [9] cannot be used. This forces one to use particle methods to implement the flow. However, flows with higher than two derivatives are generally difficult to implement because of numerical artifacts.
- Lastly, as a specific example, the gradient ascent for arclength, i.e., backward heat flow, is not stable. This is quite odd in an intuitive manner because there is nothing in the definition of length itself that indicates that a flow to increase length is unstable.

In this paper, we consider using inner products arising from Sobolev spaces to define gradients. Note that a first order Sobolev-like inner product defined on an equivalence class with respect to a group has been used in the context of shape analysis [10], but not for defining gradient flows.

2 General Theory

2.1 Structure on the Space of Curves

Let $\mathcal{C}$ denote the set of smooth embedded curves in $\mathbb{R}^2$, which is a differentiable manifold [11]. For $C \in \mathcal{C}$, we denote by $T_C\mathcal{C}$ the tangent space of $\mathcal{C}$ at C, which is isomorphic to the set of smooth perturbations of the form $h : S^1 \to \mathbb{R}^2$ where S^1 denotes the circle. We now define inner products on $T_C\mathcal{C}$.

Definition 1. *Let $C \in \mathcal{C}$, L be the length of C, and $h, k \in T_C\mathcal{C}$. Let $\lambda > 0$. We assume h and k are parametrized by the arclength parameter of C.*

1. $\langle h, k \rangle_{H^0} := \frac{1}{L} \int_0^L h(s) \cdot k(s) \mathrm{d}s$
2. $\langle h, k \rangle_{H^1} := \langle h, k \rangle_{H^0} + \lambda L^2 \langle h', k' \rangle_{H^0}$
3. $\langle h, k \rangle_{\tilde{H}^1} := \overline{h} \cdot \overline{k} + \lambda L^2 \langle h', k' \rangle_{H^0}$

where $\overline{h} := \frac{1}{L} \int_0^L h(s) \mathrm{d}s$, and the derivatives are with respect to arclength.

Note that we have introduced length dependent scale factors for convenience of later computations. It is easy to verify that the above definitions are inner products. One can easily generalize the previous definitions to $\tilde{H}^n$ by simply replacing the first derivative with the n^{th} derivative, and to H^n by adding in components of the form $\left\langle h^{(m)}, k^{(m)} \right\rangle_{H^0}$ where $m \leq n$. Also, other versions of H^1 and $\tilde{H}^1$are possible [8]. We now define the notion of gradient of a functional $E : \mathcal{C} \to \mathbb{R}$.

Definition 2. *Let $E : \mathcal{C} \to \mathbb{R}$.*

1. *If $C \in \mathcal{C}$ and $h \in T_C\mathcal{C}$, then the* variation of E *is* $\mathrm{d}E(C) \cdot h = \frac{\mathrm{d}}{\mathrm{d}t} E(C + th)\big|_{t=0}$, *where $(C + th)(\theta) := C(\theta) + th(\theta)$ and $\theta \in S^1$.*
2. *Assume $\langle , \rangle_C$ is an inner product on $T_C\mathcal{C}$. The* gradient of E *is a vector field $\nabla E(C) \in T_C\mathcal{C}$ that satisfies $\mathrm{d}E(C) \cdot h = \langle h, \nabla E(C) \rangle_C$ for all $h \in T_C\mathcal{C}$.*

For each $C \in \mathcal{C}$, note that $\mathrm{d}E(C)$ is a linear operator on $T_C\mathcal{C}$. If $\mathrm{d}E(C)$ is bounded, then the notion of operator norm can be defined. The operator norm of $\mathrm{d}E(C)$ with respect to an inner product $\langle , \rangle_C$, which induces a norm $\| \cdot \|_C$, is

$$\|\mathrm{d}E(C)\| = \sup_{h \in T_C\mathcal{C} \setminus \{0\}} \frac{|\mathrm{d}E(C) \cdot h|}{\|h\|_C}. \tag{1}$$

If the gradient of E exists, then by the Cauchy-Schwartz inequality, we have that $h = \nabla E(C)$ attains the supremum on the right hand side of (1). Note for $\lambda \to +\infty$, translations have norm approaching zero with respect to the norm induced by H^1 and $\tilde{H}^1$. In light of the interpretation of the gradient as the perturbation that attains the supremum in (1), it follows that translations are favored for gradients in H^1 and $\tilde{H}^1$ as $\lambda \to +\infty$ if they reduce energy.

2.2 Relation Between H^1 and $\tilde{H}^1$

We show that the norms associated with the inner products H^1 and $\tilde{H}^1$, *i.e.*,

$$\|h\|_{H^1} = \sqrt{\int_0^L \frac{1}{L} |h(s)|^2 + \lambda L |h'(s)|^2 \mathrm{d}s}, \quad \|h\|_{\tilde{H}^1} = \sqrt{|\overline{h}|^2 + \lambda L \int_0^L |h'(s)|^2 \mathrm{d}s}$$

are equivalent.

We first derive a simple Poincaré inequality: from $h(u) - h(v) = \int_v^u h'(s)\mathrm{d}s$ we derive that $\sup_u |h(u) - \overline{h}| \leq \int_0^L |h'(s)|\mathrm{d}s$, and then

$$\sqrt{\int_0^L |h(s) - \overline{h}|^2 \mathrm{d}s} \leq \sqrt{L} \sup_u |h(u) - \overline{h}| \leq \sqrt{L} \int_0^L |h'(s)|\mathrm{d}s \leq L\sqrt{\int_0^L |h'(s)|^2 \mathrm{d}s},$$

which is the Poincaré inequality.

We now prove the equivalence of the two norms. By Hölder's inequality, we have that $|\overline{h}|^2 \leq \frac{1}{L}\int_0^L |h(s)|^2\mathrm{d}s$ so that $\|h\|_{\tilde{H}^1} \leq \|h\|_{H^1}$. On the other hand, note that, $\frac{1}{L}\int_0^L |h(s) - \overline{h}|^2\mathrm{d}s = \frac{1}{L}\int_0^L |h(s)|^2\mathrm{d}s - |\overline{h}|^2$, so that

$$\begin{aligned}
\|h\|_{H^1}^2 &= \int_0^L \frac{1}{L}|h(s)|^2 + \lambda L |h'(s)|^2 \mathrm{d}s \\
&= \frac{1}{L}\int_0^L |h(s) - \overline{h}|^2 \mathrm{d}s + \int_0^L \lambda L |h'(s)|^2 \mathrm{d}s + |\overline{h}|^2 \\
&\leq |\overline{h}|^2 + L(1+\lambda)\int_0^L |h'(s)|^2 \mathrm{d}s \leq (1 \vee (L^2(1+\lambda)))\|h\|_{\tilde{H}^1}^2
\end{aligned}$$

where $\vee$ denotes maximum. Note that we have not established any relation between the geometry of the inner products H^1 and $\tilde{H}^1$; however, in the next sections, we show that the gradients from H^1 and $\tilde{H}^1$ have similar properties.

2.3 Comment on H^n for $n \geq 2$

As alluded to in Section 2.1, translations are favored for H^1 and $\tilde{H}^1$ gradients when $\lambda \to +\infty$. This can be quite important for tracking applications where the object to be tracked is usually translating. One may wonder whether using higher order Sobolev inner products, H^n and $\tilde{H}^n$ for $n \geq 2$, will favor higher order polynomial motions of degree n. Note however, that any polynomial perturbation defined on S^1, the circle, must be constant to conform to periodic boundary conditions. Thus, higher than order $n = 1$ Sobolev gradients also favor just translations. In this sense, there is not an advantage of using higher order Sobolev gradients. However, one gains added regularity of the gradient flow in using higher order Sobolev gradients.

3 H^1 and $\tilde{H}^1$ Gradients

In this section, we describe how to compute first order Sobolev gradients from the H^0 gradient. Denote by $f = \nabla_{H^0} E(C)$ the gradient of E with respect to the H^0 inner product at C. We would like to compute first the H^1 gradient at C. Assuming $g = \nabla_{H^1} E(C)$ exists, we have for all $h \in T_C\mathcal{C}$,

$$\mathrm{d}E(C) \cdot h = \langle h, g \rangle_{H^0} + \lambda L^2 \langle h', g' \rangle_{H^0} = \langle h, g - \lambda L^2 g'' \rangle_{H^0}$$

where we have integrated by parts and noted that we have periodic boundary conditions. Since gradients are unique (if they exist), in particular, the H^0 is unique, we must have that

$$f(s) = g(s) - \lambda L^2 g''(s) \quad \text{where } s \in [0, L]. \tag{2}$$

Note that this is an ODE defined on $[0, L]$ with periodic boundary conditions, that is, all derivatives match on the boundary of $[0, L]$.

Now we take a similar approach to compute the $\tilde{H}^1$ gradient. Assuming $g = \nabla_{\tilde{H}^1} E(C)$ exists, we have

$$\mathrm{d}E(C) \cdot h = \overline{h} \cdot \overline{g} + \lambda L^2 \left\langle h', g' \right\rangle_{H^0} = \left\langle h, \overline{g} - \lambda L^2 g'' \right\rangle_{H^0}.$$

Again by uniqueness, we have that $f = \overline{g} - \lambda L^2 g''$. Noting periodic boundary conditions, we have that $\overline{g} = \overline{f}$, and so

$$f(s) = \overline{f} - \lambda L^2 g''(s) \quad \text{where } s \in [0, L] \tag{3}$$

and we have periodic boundary conditions.

3.1 Solving the ODEs

We want to solve first the ODE (2) for g. It suffices to solve (2) with the boundary conditions $g(0) = g(L)$ and $g'(0) = g'(L)$. One can show that $g(s) = \int_0^L k_\lambda(s, \hat{s}) f(\hat{s}) \mathrm{d}\hat{s}$, where $k_\lambda : [0, L] \times [0, L] \to \mathbb{R}$ satisfies the following conditions for all $s, \hat{s} \in (0, L)$

$$k_\lambda(s, \hat{s}) - \lambda L^2 \frac{\partial^2 k_\lambda}{\partial s^2}(s, \hat{s}) = \delta(s - \hat{s}) \tag{4a}$$

$$k_\lambda(0, \hat{s}) = k_\lambda(L, \hat{s}); \;\; \partial_s k_\lambda(0, \hat{s}) = \partial_s k_\lambda(L, \hat{s}); \;\; k_\lambda(\hat{s}+, \hat{s}) = k_\lambda(\hat{s}-, \hat{s}), \tag{4b}$$

and δ denotes the Dirac distribution. It can be shown that the solution to the previous system is $k_\lambda(s, \hat{s}) = K_\lambda(|s - \hat{s}|)$, where $K_\lambda : \mathbb{R} \to \mathbb{R}$ is given by

$$K_\lambda(s) = \frac{\cosh\left(\frac{s - \frac{L}{2}}{\sqrt{\lambda} L}\right)}{2L\sqrt{\lambda} \sinh\left(\frac{1}{2\sqrt{\lambda}}\right)}, \quad \text{for } s \in [0, L], \tag{5}$$

and K_λ is periodically extended to all of $\mathbb{R}$. We may write

$$\nabla_{H^1} E(s) = \int_C K_\lambda(\hat{s} - s) \nabla_{H^0} E(\hat{s}) \mathrm{d}\hat{s} = (K_\lambda * \nabla_{H^0} E)(s) \tag{6}$$

where the integral over C denotes any range of $\hat{s}$ that corresponds to one full period around the curve C (*e.g.* [0,*L*], [–*L*,0], [–*L*/2,*L*/2], etc.).

We now solve the second ODE (3). It suffices to solve (3) with the boundary conditions $g(0) = g(L)$, $g'(0) = g'(L)$, and the relation $\overline{f} = \overline{g}$. Integrating (3) twice yields

$$g(s) = g(0) + s g'(0) - \frac{1}{\lambda L^2} \int_0^s (s - \hat{s})(f(\hat{s}) - \overline{f}) \mathrm{d}\hat{s}. \tag{7}$$

Using (7), applying the boundary conditions, and noting that $\overline{g} = \overline{f}$, after some manipulation, yields

$$g'(0) = -\frac{1}{\lambda L^3}\int_0^L s(f(s) - \overline{f})\mathrm{d}s \quad \text{and} \quad g(0) = \int_0^L f(s)\tilde{K}_\lambda(s)\mathrm{d}s \tag{8}$$

where the kernel function $\tilde{K}_\lambda$ is given by

$$\tilde{K}_\lambda(s) = \frac{1}{L}\left(1 + \frac{(s/L)^2 - (s/L) + 1/6}{2\lambda}\right), \quad s \in [0, L]. \tag{9}$$

Note that $\tilde{K}_\lambda(0) = \tilde{K}_\lambda(L)$ and thus we may periodically extend $\tilde{K}_\lambda$ as before. In this case, we may rewrite, $g(0) = \int_C f(\hat{s})\tilde{K}_\lambda(\hat{s})\mathrm{d}\hat{s}$, where, again, the integral over C denotes any range of $\hat{s}$ that corresponds to one full period over C. Now if we shift the arclength parameterization of the curve, we obtain a convolution formula for g at any point s. Therefore,

$$\nabla_{\tilde{H}^1} E(s) = \int_C \tilde{K}_\lambda(\hat{s} - s)\nabla_{H^0} E(\hat{s})\mathrm{d}\hat{s} = (\tilde{K}_\lambda * \nabla_{H^0} E)(s). \tag{10}$$

3.2 Properties of the Kernels

Note the following formal properties of K_λ and $\tilde{K}_\lambda$:

$$K_\lambda''(s) = \frac{1}{\lambda L^2}\left(K_\lambda - \delta(s)\right) \text{ and } \tilde{K}_\lambda''(s) = \frac{1}{\lambda L^2}\left(\frac{1}{L} - \delta(s)\right), \quad s \in [0, L). \tag{11}$$

The first property is just the relation in (4a), and the second is obtained through differentiation of $\tilde{K}_\lambda$. Using these relations, it is easy to see that $K_\lambda * f$ and $\tilde{K}_\lambda * f$ formally solve (2) and (3), respectively. Next, note that

$$\int_C K_\lambda(\hat{s})\mathrm{d}\hat{s} = 1 \quad \text{and} \quad \int_C \tilde{K}_\lambda(\hat{s})\mathrm{d}\hat{s} = 1 \tag{12}$$

for all $\lambda > 0$. Also observe that $K_\lambda \geq 0$ for all $\lambda > 0$, and that $\tilde{K}_\lambda \geq 0$ only when $\lambda \geq 1/24$. Finally, it is easy to verify that as $\lambda \to +\infty$, $K_\lambda \to 1/L$ and $\tilde{K}_\lambda \to 1/L$. See Fig. 1 for plots of K_λ and $\tilde{K}_\lambda$.

3.3 Properties of Sobolev Gradients

First note, from formulas (6) and (10), that the H^1 and $\tilde{H}^1$ gradients are geometric, *i.e.*, they do not depend on a particular parametrization chosen for the curve. This is also evident from the definition of these inner products. The formulas (6) and (10) show that there may be a tangential component of the gradients; but these tangential components may be ignored when considering gradient flows. This is different from H^0 where if the energy is geometric, then the gradient will have only a normal component.

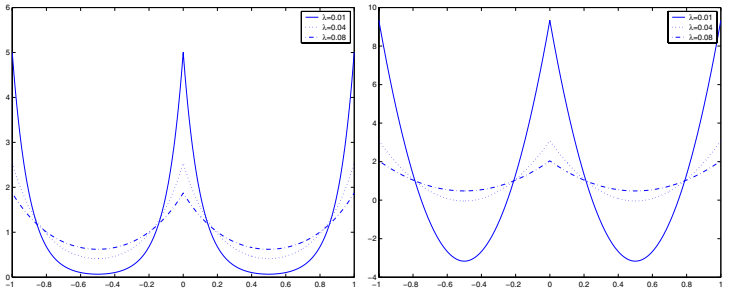


Fig. 1. Plots of K_λ (left) and $\tilde{K}_\lambda$ (right) for various λ with $L = 1$. The plots show the kernels over two periods.

Because H^1 and $\tilde{H}^1$ gradients are given by integrals of H^0, given in formulas (6) and (10), integration by parts and the relations in (11) imply that two derivatives of the curve can be moved to derivatives on the kernels. This means that H^1 and $\tilde{H}^1$ gradients involve two fewer derivatives of the curve than H^0 gradients involve. Note that H^0 gradients have twice the number of derivatives of the curve as is defined in the energy E to be optimized. Thus, fourth order evolution equations of curves in H^0 may reduce to second order equations in H^1 and $\tilde{H}^1$. A similar remark can be made for H^n and $\tilde{H}^n$ gradients; these gradients require $2n$ less derivatives of the curve than the H^0 gradient requires.

The property that the integral of both the kernels is unity (12) implies that the H^1 gradient can be interpreted as a weighted average of the H^0 gradient; the same interpretation holds for $\tilde{H}^1$ when $\lambda > 1/24$. In light of this weighted average interpretation, we see that Sobolev gradients are less sensitive to noise and local features than H^0 gradients are. Moreover, the property that the kernels approach $1/L$ as $\lambda \to +\infty$ shows that, in this case, the H^1 and $\tilde{H}^1$ gradients approach pure translations equal to the average value of the H^0 gradient, as expected from the interpretation of gradient noted in Section 2.1.

3.4 Advantages of $\tilde{H}^1$ over H^1

There is a computational advantage of using the $\tilde{H}^1$ gradient as opposed to the H^1 gradient since the formulas (7), (8) give the $\tilde{H}^1$ gradient as a single integral without convolution, as opposed to the necessary convolution for H^1. Another advantage of $\tilde{H}^1$ over H^1 is that we can eliminate the dependence on the parameter λ when implementing $\tilde{H}^1$ gradient flows. Observe from the kernel definition (9) that $\tilde{K}_\lambda$ is a sum of two terms: one that depends on λ and another that does not. Thus, the $\tilde{H}^1$ gradient is a sum of two components: one that depends on λ by a simple scale factor, and another that is independent of λ. The component that does not depend on λ is $\overline{\nabla_{H^0} E}$, which is a just a translation. The other component is a complex deformation. An algorithm to implement the $\tilde{H}^1$ gradient flow is to first evolve the curve by the translation component until this component becomes zero, then to evolve the curve by the deformation

component, and the process is repeated until convergence. Note that λ does not need to be chosen for evolving the deformation component because λ only changes the speed of the curve, not the geometry. Therefore, this algorithm also gives a way of separating the (rigid) motion of the curve from the deformation. Separating the motion from deformation has particular importance in tracking applications [12].

4 Some Sobolev Gradient Flows

In this section, we simplify the formulas (6) and (10) for some common geometric energies, note some interesting properties, and compare these with the usual H^0 gradients. In what follows, we use K to denote either the kernel (5) or (9), and ∇_1 will denote either the H^1 or $\tilde{H}^1$ gradient; when the distinction is needed, we will use the subscript λ on the kernels, and write H^1 or $\tilde{H}^1$.

4.1 Length and Weighted Length

We consider the geodesic active contour model [2,3]. The energy is $E(C) = \int_C \phi(C(s))\mathrm{d}s$ where $\phi : \mathbb{R}^2 \to \mathbb{R}^+$. Then the gradient with respect to H^0 is $\nabla_{H^0} E = L(\nabla\phi \cdot \mathcal{N})\mathcal{N} - L\phi\kappa\mathcal{N}$ where $\mathcal{N}$ is the unit inward normal, and κ is the curvature. Let us first note that $\nabla_{H^0} E = L\nabla\phi - L(\phi C')'$. Integrating by parts we find that

$$\frac{1}{L}\nabla_1 E = \nabla\phi * K - (\phi C')' * K = \nabla\phi * K - (\phi_s C) * K' - (\phi C) * K'',$$

where $\phi_s(\hat{s}) := \mathrm{d}/\mathrm{d}\hat{s}\phi(C(\hat{s}))$. Using the relations in (11), we find that

$$\nabla_{\tilde{H}^1} E = \frac{\phi C - \overline{\phi C}}{\lambda L} - L(\phi_s C) * \tilde{K}'_\lambda + L\nabla\phi * \tilde{K}_\lambda. \tag{13}$$

Of particular interest is when $\phi = 1$, that is $E = L$, the length of the curve. We see that $\nabla_{\tilde{H}^1} L = \frac{C - \overline{C}}{\lambda L}$. It is interesting to notice that the H^1 and $\tilde{H}^1$ gradient flows are stable for both ascent and descent while the H^0 gradient flow is only stable for descent. Note that the $\tilde{H}^1$ gradient flow constitutes a simple rescaling of the curve about its centroid. While the H^0 gradient descent smooths the curve, the $\tilde{H}^1$ gradient descent (or ascent) has neither a beneficial nor a detrimental effect on the regularity of the curve.

4.2 Area and Weighted Area

We consider region-based active contour models [5,6]. The energy is $E(C) = \int_{C_{in}} \phi \mathrm{d}A$ where C_{in} denotes the region enclosed by the closed curve C, $\phi : \mathbb{R}^2 \to \mathbb{R}$ and $\mathrm{d}A$ is the area form. The gradient with respect to H^0 is $\nabla_{H^0} E = -L\phi\mathcal{N} = -L\phi JC'$ and J is a rotation by 90° matrix. Integrating by parts we find that

$$\frac{1}{L}\nabla_1 E = -(\phi JC') * K = (\phi_s JC) * K + (\phi JC) * K'. \tag{14}$$

For the $\tilde{H}^1$ gradient, this simplifies to

$$\nabla_{\tilde{H}^1} E = \frac{J}{\lambda L^2} \int_0^L \Big(\phi C(\cdot + \hat{s}) - \overline{\phi C}\Big) \hat{s}\, \mathrm{d}\hat{s} + (\phi_s JC) * \tilde{K}_\lambda. \tag{15}$$

Of particular interest is when $\phi = 1$, that is $E = A$, the area enclosed by the curve. We see that $\nabla_1 A = (JC) * K'$. This simplifies to the gradient ascent/descent

$$C_t(s) = \pm \frac{J}{\lambda L^2} \int_0^L \Big(C(s + \hat{s}) - \overline{C}\Big) \hat{s} \mathrm{d}\hat{s} \tag{16}$$

in the $\tilde{H}^1$ gradient case.

4.3 Elastic Energy

Consider the geometric version of elastic energy defined by $E(C) = \int_C \kappa^2 \mathrm{d}s = \int_C \|C''\|^2 \mathrm{d}s$. It can be shown that the H^0 gradient is $\nabla_{H^0} E = L(2C^{(3)} + 3(C'' \cdot C'')C')'$. Thus, we find that

$$\frac{1}{L} \nabla_1 E = \Big(2C^{(3)} + 3(C'' \cdot C'')C'\Big)' * K = 2C'' * K'' - 3(C'' \cdot C'')C' * K'.$$

For the kernel $\tilde{K}_\lambda$, this simplifies to

$$\frac{1}{L} \nabla_{\tilde{H}^1} E = -\frac{2C''}{\lambda L^2} - 3(C'' \cdot C'')C' * \tilde{K}'_\lambda, \tag{17}$$

and thus, the gradient descent, up to a scale factor depending on λ and L, is

$$C_t = C_{ss} + \frac{3}{2} \int_s^{s+L} (C_{ss} \cdot C_{ss}) C_s \left(\frac{\hat{s} - s}{L} - \frac{1}{2}\right) \mathrm{d}\hat{s}. \tag{18}$$

4.4 Comparison of H^0 and H^1, $\tilde{H}^1$

We notice several advantages of the gradients flows for H^1 and $\tilde{H}^1$ gradients as compared with H^0 gradients. First note that both the expressions for edge-based and region-based active contour gradients with respect to H^1 and $\tilde{H}^1$ (13), (14) do not involve any derivatives of the curve. This is in contrast to H^0, which requires two derivatives for geodesic active contours and one derivative for region-based active contours. Hence, the Sobolev flows are defined for polygons, without the use of viscosity theory. Note that the expression in (13) does not require any more derivatives of ϕ than the expression for H^0 does. This is not the case for (14), which requires a derivative of ϕ. However, since ϕ_s is contained within a convolution, the possible noise generated by ϕ_s is mitigated. Alternatively, the original expressions (6) and (10) may be used if a derivative of ϕ is not desired to be computed. Notice the expressions of Sobolev gradients for the elastic energy (17) only require two derivatives of the curve; this is in contrast to the H^0 gradient, which requires four derivatives of the curve. Since there is no maximum principle for fourth order equations, the H^0 gradient of elastic energy cannot be implemented using level set methods. Thus, a particle method must be used; however, this is prone to numerical problems.

5 Simulations

In this section, we show some simulations of Sobolev active contours used for segmentation and tracking. In all the simulations done below, the results for the Sobolev active contours are done with the H^1 inner product with $\lambda = 10$, although higher λ produces similar results with higher regularity. Using $\tilde{H}^1$ gives the same results as to what are shown.

Figure 2 shows snapshots of evolutions that use the edge-based energy [2,3] for both the H^0 (top) and the H^1 (bottom) active contours to segment a noisy image. The initial contour is a shifted version of the true object with a slightly different radius. Notice that the H^0 active contour learns local features instantly, and therefore becomes stuck at an undesirable local minimum. On the other hand, the Sobolev H^1 active contour moves according to a global motion first, then when it cannot reduce energy by moving in a global manner, it begins to learn local features. As a result, the Sobolev active contour overcomes any undesirable local minima created by the noise. Figure 3 shows a similar experiment

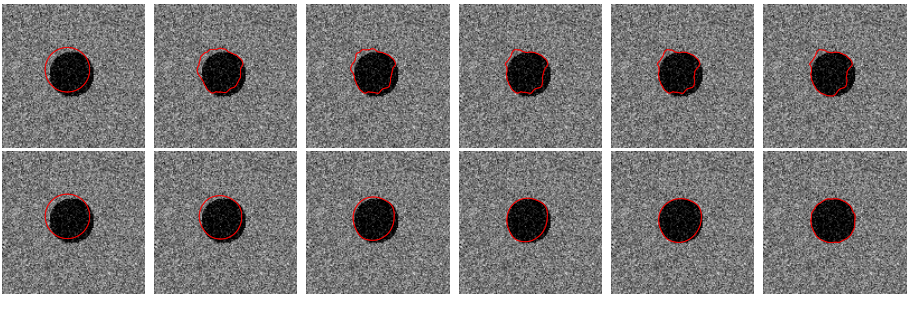


Fig. 2. Segmentation of a noisy image using edge-based H^0 active contour (top), and edge-based Sobolev active contour (bottom)

using the region-based energy [6]. The top row of the figure shows the result using the H^0 active contour. Note that a curvature term is added to keep the curve smooth. A curvature-data term ratio of 2000 to 1 is used for the H^0 active contour. Notice that although global statistics are used to define the energy, points on the H^0 active contour move independently without knowledge of other points on the curve. Thus, the curve becomes unsmooth instantly, being susceptible to local features, even though a high curvature weighting is used. The flow converges to an undesirable local minimum. On the contrary, the H^1 flow preserves the shape of the initial contour as it translates until translations are no longer favorable to reduce energy. Then the contour deforms from a square shape to a circular shape as fine scale features of the image are learned. Note that there was no curvature term added to the Sobolev active contour; the regularity is achieved solely through the inner product definition.

Figures 4 and 5 show examples of tracking a square that translates using both H^0 active contours (in red) and Sobolev active contours (in blue). The

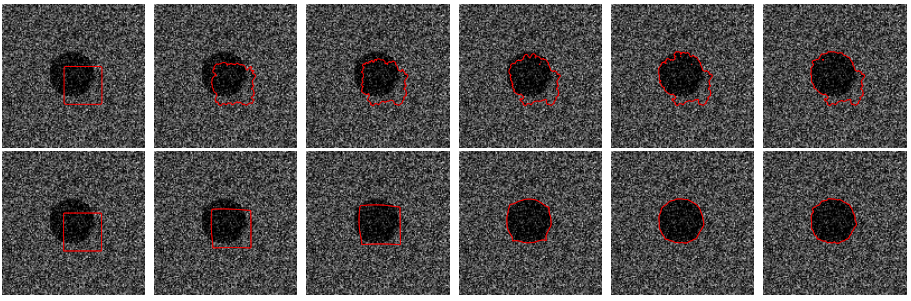


Fig. 3. Segmentation of a noisy image using region-based H^0 active contour with a length penalty (top), and region-based Sobolev active contour (bottom)

segmentation result from the previous frame is used as the initial contour for the next frame. The segmentation evolutions are run until convergence of both contours. The first example in Fig. 4 shows the result using the edge-based energy. The second example in Fig. 5 shows the result using the region-based Chan-Vese energy. A curvature regularizer for the H^0 region-based active contour at a ratio of 1000 to 1 was used to compensate for noise. Notice the H^0 active contours becomes stuck in a undesirable local minima after the initial movement of the object, and soon lose track of the object. The Sobolev active contours do not have this problem and successfully track the object.

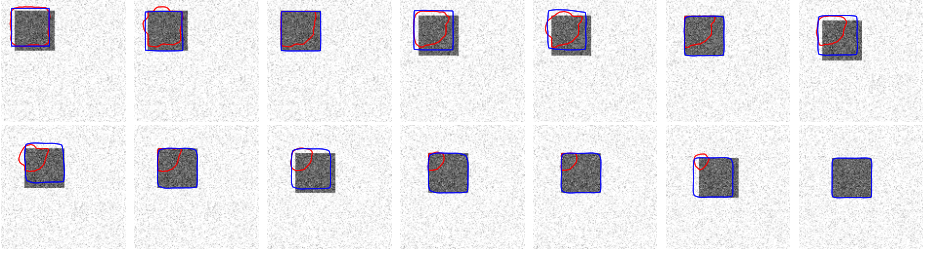


Fig. 4. Tracking a moving square in a noisy environment using edge-based H^0 active contour (red) and Sobolev active contour (blue)

6 Conclusion

We have introduced using Sobolev inner products on the set of perturbations of a curve rather than the traditional H^0 inner product used in all previous works on geometric active contours. We have demonstrated the general methodology for computing Sobolev gradients, and derived various flows with respect to Sobolev inner products. We have shown that Sobolev flows are smooth in the space of curves, are not as dependent on local image information as H^0

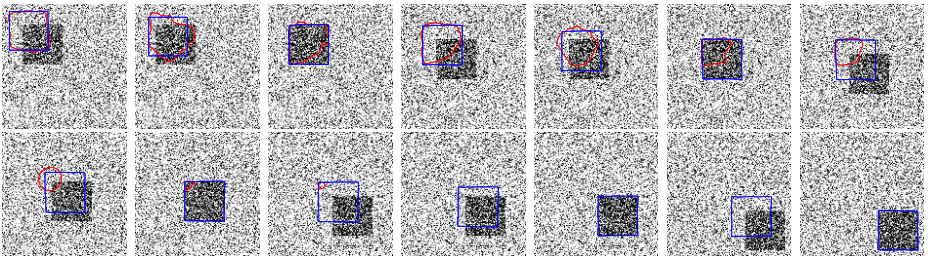


Fig. 5. Tracking a moving square in a noisy environment using region-based H^0 active contour (red) and Sobolev active contour (blue)

flows, are global motions which deform locally after moving globally, and do not require derivatives of the curve to be defined for region-based and edge-based energies.

Acknowlegdements. We thank George Roberts for aiding with simulations.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1** (1987) 321–331
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proceedings of the IEEE Int. Conf. on Computer Vision, Cambridge, MA, USA (1995) 694–699
3. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Gradient flows and geometric active contour models. In: Proceedings of the IEEE Int. Conf. on Computer Vision. (1995) 810–815
4. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. **42** (1989) 577–685
5. Yezzi, A., Tsai, A., Willsky, A.: A statistical approach to snakes for bimodal and trimodal imagery. In: Int. Conf. on Computer Vision. (1999) 898–903
6. Chan, T., Vese, L.: Active contours without edges. IEEE Transactions on Image Processing **10** (2001) 266–277
7. Michor, P., Mumford, D.: Riemannian geometries on the space of plane curves. ESI Preprint 1425, arXiv:math.DG/0312384 (2003)
8. Yezzi, A., Mennucci, A.: Metrics in the space of curves. Preprint, arXiv:math.DG /0412454 (2005)
9. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: algorithms based on the Hamilton-Jacobi equations. J. Comp. Physics **79** (1988) 12–49
10. Younes, L.: Computable elasic distances between shapes. SIAM J. Appl. Math. **58** (1998) 565–586
11. Kriegl, A., Michor, P.: The Convenient Setting of Global Analysis. Volume 53 of Mathematical Surveys and Monographs. American Mathematical Society (1997)
12. Soatto, S., Yezzi, A.J.: DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In: ECCV (3). (2002) 32–57

Advances in Variational Image Segmentation Using AM-FM Models: Regularized Demodulation and Probabilistic Cue Integration

Georgios Evangelopoulos, Iasonas Kokkinos, and Petros Maragos*

Computer Vision, Speech Communication and Signal Processing Group,
National Technical University of Athens, Greece
{gevag, jkokkin, maragos}@cs.ntua.gr
http://cvsp.cs.ntua.gr

Abstract. Current state-of-the-art methods in variational image segmentation using level set methods are able to robustly segment complex textured images in an unsupervised manner. In recent work, [18,19] we have explored the potential of AM-FM features for driving the unsupervised segmentation of a wide variety of textured images. Our first contribution in this work is at the feature extraction level, where we introduce a regularized approach to the demodulation of the AM-FM -modelled signals. By replacing the cascade of multiband filtering and subsequent differentiation with analytically derived equivalent filtering operations, increased noise-robustness can be achieved, while discretization problems in the implementation of the demodulation algorithm are alleviated. Our second contribution is based on a generative model we have recently proposed [18,20] that offers a measure related to the local prominence of a specific class of features, like edges and textures. The introduction of these measures as weighting terms in the evolution equations facilitates the fusion of different cues in a simple and efficient manner. Our systematic evaluation on the Berkeley segmentation benchmark demonstrates that this fusion method offers improved results when compared to our previous work as well as current state-of-the-art methods.

1 Introduction

The segmentation of textured images is a long standing problem in computer vision that has been addressed in the framework of variational methods using both boundary- and region-based techniques. The latter are commonly held as more appropriate for this specific problem due to the increased robustness offered by region-based criteria and the difficulty of texture boundary localization.

In the region-based scenario, informative features are used to drive the evolution process; image intensity on its own is a poor cue, since textured images are inherently of varying intensity. Multiband image filtering with filterbanks is commonly used as a preprocessing step that facilitates the extraction of texture information residing at different

* This work was supported in part by the Greek research program HRAKLEITOS, which is co-funded by the European Social Fund (75%) and National Resources (25%), the Greek GSRT research program PENED-2001 and the European NoE MUSCLE.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 121–136, 2005.

frequency channels. Even though the outputs of such a filterbank may accurately describe the texture signal, their high dimensionality can lead to suboptimal segmentations.

In recent work [18,19,37] the potential of modulation features derived using the AM-FM model of Bovik and coworkers [2,11,10,12] in driving the unsupervised segmentation of textured images has been explored. The low-dimensional texture representation resulting from Dominant Component Analysis (DCA) [10,12] offers information concerning the local contrast, scale and orientation of the texture signal and can be interpreted as describing the sinusoidal signal that best models a texture locally [18,20].

A problem faced by our algorithm is that at smooth areas texture features like orientation are meaningless and introduce erroneous information in the feature vector, while at object borders texture features indicate the presence of a textured region, leading again to suboptimal solutions. Further, the demodulation algorithm used for feature extraction includes high-order image derivatives, that introduce increased noise sensitivity and are not uniquely defined for discrete-time signals.

In this work, our contribution is twofold: first we introduce a regularized version of the algorithm used for feature extraction, involving generalized Gabor filtering and treating errors from inefficient discrete differentiations. Second, we propose a modification of the original Region Competition/ Geodesic Active Regions evolution rule that takes into account the locally estimated confidence in any of the low-dimensional modulation-based features. For this we build upon recent work [18,20] and provide probabilistic terms quantifying the confidence assigned to the extracted features, relying on a detection theoretic interpretation of the DCA algorithm.

Section II describes previous work and provides the background for later sections. In Section III the regularized demodulation algorithm is described, while Section IV presents the cue integration algorithm proposed. In Section V we demonstrate the merits of using the fused scheme and compare both visually and quantitatively our method's results to those obtained using current state-of-the-art features [38].

2 Previous Work: AM-FM Models and Unsupervised Segmentation Methods

2.1 AM-FM Texture Modeling

According to the multicomponent AM-FM model [10], a textured image can be modelled as the superposition of sinusoidal components:

$$I(x,y) = \sum_{k=1}^{K} a_k(x,y)\cos(\phi_k(x,y)),\ \boldsymbol{\omega}_k(x,y) = \nabla\phi_k(x,y), \tag{1}$$

where each of the K components is a non-stationary 2-D AM-FM signal, with *instantaneous amplitude* $a_k(x,y)$ and *instantaneous frequency* $\boldsymbol{\omega}_k$. The decomposition of an image in terms of this expression is an ill-posed problem, since one can devise an infinity of AM and FM signals yielding the same image. A separation into well-behaved individual AM-FM components can be accomplished by filtering with a multiband Gabor filterbank [1,7]; the output of each filter can then be represented as a mono-component

AM-FM signal with *narrowband* modulation components, that lend themselves to efficient demodulation algorithms and are intuitively interpretable. Specifically if $f(x,y)$ is a mono-component 2-D AM-FM signal

$$f(x,y) = a(x,y)\cos(\phi(x,y)), \tag{2}$$

its spatially-varying amplitude $a(x,y)$ can be interpreted as modeling local image contrast while the instantaneous frequency vector $\boldsymbol{\omega}(x,y) = \nabla\phi(x,y)$ describes locally emergent spatial frequencies [2,12]. Efficient estimation of the modulation components of the 2D AM-FM signals can be accomplished via the multidimensional energy separation algorithm [24] that uses a multidimensional energy operator [3]: Let $f(x,y)$ be a twice-differentiable continuous-space real-valued input function. The 2D energy operator Φ is defined by

$$\Phi(f)(x,y) \triangleq \|\nabla f(x,y)\|^2 - f(x,y)\nabla^2 f(x,y) \tag{3}$$

Let now f be a 2D spatial AM-FM signal as in (2). Under certain assumptions on the amplitude and frequency variations [24], applying Φ to f yields the energy product of the squared instantaneous amplitude and frequency magnitude $\Phi[a\cos(\phi)] \approx a^2||\boldsymbol{\omega}||^2$ with an approximation error bounded within a negligible range for locally narrowband signals. Applying Φ to the partial derivatives $f_x = \partial f/\partial x$, $f_y = \partial f/\partial y$ yields the 2D continuous *Energy Separation Algorithm (ESA)* [24]:

$$\sqrt{\frac{\Phi(f_x)}{\Phi(f)}} \approx |\omega_1(x,y)| \quad \sqrt{\frac{\Phi(f_y)}{\Phi(f)}} \approx |\omega_2(x,y)| \quad \frac{\Phi(f)}{\sqrt{\Phi(f_x)+\Phi(f_y)}} \approx |a(x,y)| \tag{4}$$

This algorithm can estimate at each location (x,y) the amplitude and the magnitude of the instantaneous vertical and horizontal frequencies of the spatially-varying 2-D AM-FM signal. The signs of the frequency signals are obtained from the signs of the carriers, approximated by the bandpass filter central frequencies. By replacing the partial derivatives with differences a variety of discrete energy operators emerge. A simple 2D case is:

$$\Phi_d(f)(i,j) = 2f^2(i,j) - f(i-1,j)f(i+1,j) - f(i,j-1)f(i,j+1) \tag{5}$$

Applying Φ_d to a 2D discrete AM-FM signal $f[i,j] = a[i,j]\cos(\phi[i,j])$ yields [24] a nonlinear energy product $\Phi_d[a[i,j]\cos(\phi[i,j])] \approx a^2[i,j](\sin^2(\Omega_1[i,j]) + \sin^2(\Omega_2[i,j]))$, where Ω_1, Ω_2 are the discrete-space instantaneous frequencies. The discrete ESA [24], can give estimates of instantaneous amplitude and frequencies of narrowband image components with an excellent spatial resolution and very low complexity.

2.2 Low-Dimensional AM-FM Features via Dominant Component Analysis

The previously described demodulation scheme yields a $3K$-dimensional feature vector at each point, where K is the number of filters used in the Gabor filterbank. Even though this descriptor offers rich information about the texture signal, it cannot be used as is for segmentation purposes, due to its high dimensionality.

A more compact descriptor can be extracted using the Dominant Component Analysis (*DCA*) [10,12] scheme: DCA picks at each image pixel the most active filterbank channel, demodulates its output and uses the resulting AM-FM features to represent the local texture structure. This offers at each image point a three-dimensional feature vector that retains essential information about the texture structure, describing its most prominent characteristics in terms of a sinusoidal signal.

At the heart of the DCA method lies the channel selection criterion used to pick the most active channel at each point; in the original work on DCA the local estimates of the amplitude envelopes A_k for each channel k were used, which are estimated as $A_k = \sqrt{\Im(g_k * I)^2 + \Re(g_k * I)^2}$, where I is the image and g_k is the impulse response of the k-th complex 2-D Gabor filter. In [18] we presented a detection theoretic interpretation of this 'maximum-amplitude' channel selection criterion, relating the A_k term with the log-likelihood of the image observations around the neighborhood of each point. Further, it was observed that using an energy-operator-based instead of an amplitude-based selection criterion offers a viable alternative, characterized by better localization accuracy and in [20] this alternative channel selection criterion was cast in a detection-theoretic framework as well.

2.3 Unsupervised Variational Textured Image Segmentation

Region-based techniques are commonly considered as more appropriate for textured image segmentation, since the application of boundary-based techniques and the related variational schemes of Snakes, Deformable models [15,6] and Geodesic Active Contours [5,16] is based on the detection of strong variation in texture features [22,31], which is a non-trivial problem.

Some of the first variational region-based textured image segmentation techniques [21,46] have used modified versions of the Mumford-Shah functional [28] appropriately modified to incorporate the multi-dimensional features used for texture description; building upon this work, current state-of-the-art algorithms in unsupervised variational region-based segmentation [45,33,43,38,13,4] rely on the level-set methodology [29,23,41,34] which has been established as an elegant and efficient mathematical tool for the solution of problems involving evolving interfaces, offering robustness and tractability.

A significant precursor of recent work has been the Region Competition [46] method, which has helped clarify and unify different variational criteria and has introduced a probabilistic flavor in the curve evolution literature. The core idea of this algorithm is the maximization of the probability of the image observations I, using a set of M regions R_i, within which the observations are assumed to follow a simple region-specific parametric distribution $P(\cdot; a_i)$; an additional term on the length of the region borders, Γ_i is used to give rise to the following functional:

$$J(\Gamma, \{a_i\}) = \sum_{i=1}^{M} \frac{\mu}{2} \int_{\Gamma_i} ds - \iint_{R_i} \log P(I; a_i) \tag{6}$$

We have omitted the penalty on the number of regions used in [46], since in our case this remains fixed throughout the segmentation process. Calculus of variations yields the

evolution of the region borders as the motion along the direction that assigns pixel observations to the region that models them better while maintaining the borders smooth:

$$\frac{\partial \Gamma_i}{\partial t} = -\mu\kappa\mathcal{N} + \log\frac{P(I;a_i)}{P(I;a_j)}\mathcal{N} \tag{7}$$

where κ is the curvature and $\mathcal{N}$ the outward normal of front Γ_i and j is the neighboring region that competes with i for the observations at the interface position. Parameter estimation for the distributions $P(\cdot;a_i)$ is performed in alternation with curve evolution yielding an adaptive unsupervised image segmentation scheme.

In [33] this evolution algorithm has been brought together with the level set methodology and combined with edge-based terms, giving rise to the Geodesic Active Regions (GAR) algorithm

$$\frac{\partial \Gamma_i}{\partial t} = \lambda\log\frac{P(I;a_i)}{P(I;a_j)} - (1-\lambda)\left[g(P_c)\kappa\mathcal{N} + (\nabla g(P_c)\cdot\mathcal{N})\mathcal{N}\right] \tag{8}$$

where $g(\cdot)$ a monotonically decreasing function, P_c the probability of a pixel belonging to a boundary and λ determines the relative weights assigned to region- and edge-based information. The last term is inspired from the Geodesic Active Contour Model [16,5] and forces the region borders to stay close to the locations where an edge detector responds strongly. Other successful algorithms like [43,45] can be seen to be of the same essence with the original Region Competition algorithm; for example the model of [43] is based on the cartoon approximation to the Mumford-Shah functional [28], which in turn is a special case of the Region Competition functional for Gaussian distributions with equal variances.

2.4 Texture Features for Unsupervised Variational Segmentation

As mentioned earlier, even though the outputs of a filterbank may provide a rich description of the texture signal, their high dimensionality can lead to suboptimal segmentations. In the supervised texture segmentation scenario e.g. [32] this problem is bypassed by choosing the channels that maximally separate different textures. It is however harder to tackle the unsupervised problem, since choosing the best channels is equivalent to projecting the features onto a subspace where some *unknown a-priori* classes become maximally separated; this is usually performed using heuristic criteria, as e.g. in [39,40].

In a recent attempt to alleviate the problem for unsupervised segmentation, Rousson et al. [38] have used a vector valued diffusion procedure to smooth a low-dimensional image descriptor, derived from local image derivatives. Combined with image intensity the resulting four-dimensional feature vector offers satisfactory results for the unsupervised segmentation of textured images. In the information-theoretic approach of [17] segmentation is accomplished without using a feature extraction stage, using as the sole criterion the maximization of the mutual information between region label and image intensity. In [13] distributions tuned to capture natural image statistics were shown to result in improved results when incorporated in curve evolution schemes for texture segmentation.

The representation of a texture in terms of its DCA features offers a low-dimensional feature vector, that is expressive enough for the discrimination of a wide variety of textures, encompassing information about texture strength, scale and orientation. Specifically, the feature vector we consider consists of the image intensity and the DCA components, namely amplitude, frequency magnitude and orientation. The distribution of the feature vector $P(\cdot; a_i)$ within region i is modelled as a product of a multivariate Gaussian for the first three dimensions and a von-Mises distribution for the orientation feature θ, which is analogous to the Gaussian distribution for orientational data:

$$P_{VM}(\theta; \theta_0, \kappa) \propto \exp(\kappa \cos(\theta - \theta_0)) \tag{9}$$

The parameters θ_0, κ of this distribution are estimated as in [9]. Segmentation results using the DCA feature vector can be found in Section V as well as in [19].

3 Regularized Demodulation

Let us consider an image I of K locally narrowband components $f_k(x, y)$ modelled by AM-FM signals and corrupted by a WSS zero-mean Gaussian noise field $w(x, y)$:

$$I(x,y) = \sum_{k=1}^{K} \underbrace{a_k(x,y)\cos(\boldsymbol{\omega}_{k0} \cdot (x,y) + \phi_k(x,y))}_{\approx f_k(x,y)} + w(x,y)$$

For each component f_k, assuming a negligible AM-FM modeling error, its instantaneous frequency is given by $\boldsymbol{\omega}_k = \boldsymbol{\omega}_{k0} + \nabla\phi_k(x, y)$, where the carrier $\boldsymbol{\omega}_{k0}$ is its mean frequency and $\phi_k(x, y)$ is the nonlinear phase part. The fundamental problem of demodulating the image I aims at estimating the instantaneous amplitudes $a_k(x, y)$ and frequencies $\boldsymbol{\omega}_k(x, y)$. Unavoidable modeling errors of any demodulation algorithm, the presence of noise, interference from neighbor spectral components, and space discretization of the signal derivatives are possible sources that can cause errors in the demodulation of each narrowband component $f_k(x, y)$. Robustness in the AM-FM demodulation problem can be achieved in various ways, e.g. by optimizing any one or some of the following problems: (1) Reduction of the error in modeling each narrowband component $f_k(x, y)$ by a 2D AM-FM signal while maintaining some smoothness in the estimated amplitude and frequency modulation signals. (2) Suppression of noise. (3) Suppression of neighbor spectral components while estimating one component. (4) Regularization of derivatives. Simultaneously achieving all the above goals is a complex optimization task, which remains an unsolved problem. We present a regularized 2D energy operator and a related regularized 2D ESA that address some of the above problems in more than one combinations.

In the 1D case [30], given a narrowband signal $f(x)$ to model by $a(x)\cos(\omega_0 x + \phi(x))$, problem (1) has been given an optimum solution based on the Hilbert transform and analytic signal, which minimizes the mean-squared energy $\int |a'(x)|^2 dx$ of the amplitude derivative. This yields an optimum carrier frequency ω_0 as the center of gravity of the one-sided power spectrum $\int_0^\infty |F(\omega)|^2 d\omega$ of the given signal $f(x)$. In [36] it has been shown that, the ESA estimates of instantaneous amplitude and frequency of $f(x)$

bare a modeling error comparable to that of the Hilbert transform. However ESA has a better space-time resolution and much lower complexity. A simultaneous solution to problems (2) and (3) has been given in [3] by using a filterbank of bandpass filters, each centered at the spectral mean location of each narrowband component. The bandpass filtering increases the SNR and reduces the bias and variance of the ESA estimates of instantaneous amplitude and frequencies.

Given discrete image data, problem (4) immediately arises since the energy operator involves two differential operators. As analyzed in [35] for the problem of edge detection, two *regularized solutions*, which minimize the sum of the data approximation error and the energy of the second derivative of the approximating function, are (i) spline interpolation and (ii) convolution of the image data by a function that can be closely modelled by a Gaussian. In our problem which deals with narrowband but not necessarily lowpass signal components the Gaussian filter response must be modulated by a sine with carrier equal to the spectral mean location of the component. This yields a Gabor filter. In [8], the spline and the Gabor regularization of the energy operator and of the ESA were compared for 1D signals. This comparison yielded a slight superiority of the Gabor ESA.

Motivated by all the above, we propose a *2D Gabor ESA* algorithm for simultaneous filtering and demodulation. Let $I(x,y)$ be the continuous image, $g(x,y)$ the impulse of a real[1] 2D Gabor filter, and $f(x,y) = I(x,y) * g(x,y)$ the output of the Gabor filter. Since, convolution commutes with differentiation, the continuous 2D energy operator combined with Gabor bandpass filtering becomes

$$\Psi(f) = \Psi(I * g) = \|I * \nabla g\|^2 - (I * g)(I * \nabla^2 g) \tag{10}$$

Thus, the differential operators have been replaced by derivatives of the Gabor filter. The final algorithm for the Gabor energy operator (EO) becomes: (1) Find analytically and store all required differential formulae of the Gabor function $g(x,y)$ evaluated on the pixel locations of a sampling grid $(i,j) = (i\Delta x, j\Delta y)$. We need three differential formulae: $g_x, g_y, \nabla^2 g$. (2) For estimating the instantaneous energy at the pixel locations (i,j) use the formula (10) of the combined continuous energy operator and Gabor filtering by using for each convolution the discrete convolution of the given image data $I[i,j]$ and the required Gabor derivative sampled at (i,j).

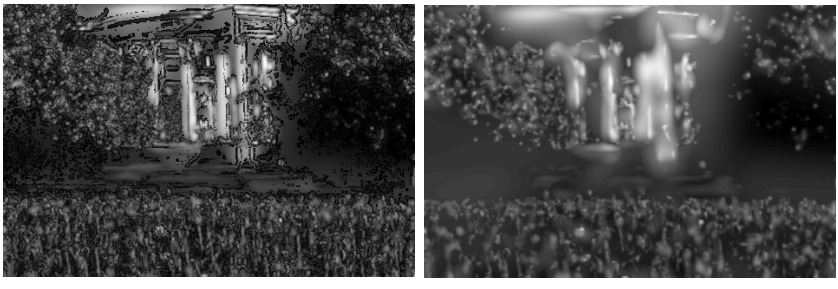
Similarly, for estimation of the instantaneous amplitude and frequency, the 2D Gabor ESA for demodulating $f = I * g$ consists of the following two steps. (1) Use the Gabor EO to compute the instantaneous energies of three image functions: $\Psi(f)$, $\Psi(f_x = I * g_x) = \|I * \nabla g_x\|^2 - (I * g_x)(I * \nabla^2 g_x)$ and $\Psi(f_y = I * g_y)$. For all three energies we need seven Gabor differential formulae: $g_x, g_y, g_{xx}, g_{yy}, g_{xy}, \nabla^2 g_x, \nabla^2 g_y$. (2) Use the evaluated energies in the formula of the 2D continuous ESA. The 2D Gabor EO is computationally more intensive than the corresponding discrete EO, since it needs three convolutions (compared with one for the discrete case), but adds robustness and improved performance.

[1] If we use a complex Gabor filter, then we can use a 2D energy operator for complex-valued signals f defined in [24] by $C(f) = \Psi[\Re(f)] + \Psi[\Im(f)]$, i.e. the sum of the real energy operator applied to the real and imaginary part of the complex signal.

One approach to reduce the total complexity of applying the Gabor EO or the ESA to all filter outputs is to use the following modified procedure: (1) Apply Gabor bandpass filters to obtain all narrowband components $f_k = I * g_k$. (2) To each component $f_k(x, y)$ apply the following *Regularized Energy Operator* (REO)

$$\Phi_\sigma(f_k) = \|f_k * \nabla G_\sigma\|^2 - f_k(f_k * \nabla^2 G_\sigma)$$

where $G_\sigma(x, y) = (1/2\pi\sigma^2)\exp[-(x^2 + y^2)/2\sigma^2]$ is an isotropic Gaussian at regularization scale σ, and ∇G_σ and $\nabla^2 G_\sigma$ are the well-known gradient-of-Gaussian and Laplacian-of-Gaussian operators.The REO needs only 3 convolutions of the narrowband image component f_k with the $\partial/\partial x$, $\partial/\partial y$ and ∇^2 of the Gaussian. For the corresponding regularized ESA, we need the three regularized energies $\Phi_\sigma(f_k)$, $\Phi_\sigma(\partial f_k/\partial x) = \|f_k * \nabla(\partial_x G_\sigma)\|^2 - f_k[f_k * \nabla^2(\partial_x G_\sigma)]$ and $\Phi_\sigma(\partial f_k/\partial y)$. All the Gaussian differential formulae are common for all filters and need to be computed once.



(a) (b)

Fig. 1. Regularized Features: Dominant Amplitude **(a)** ESA **(b)** Gabor ESA

4 Probabilistic Cue Integration

A problem faced by our algorithm, as well as most feature-based segmentation algorithms, is that at areas where the model underlying the feature extraction process fails to accurately capture signal behavior the features may be meaningless and can drive the segmentation to suboptimal solutions. For example, in the presence of edges the DCA-based amplitude is typically high, describing large oscillations while at smooth regions the orientation of the frequency vector varies erratically. In this section we introduce a method that renders the segmentation process immune to such problems by automatically choosing which features the evolution equations should rely upon.

Our method uses a confidence measure assigned to the features used to drive the segmentation process; this may be hard to obtain generally, but can be naturally accomplished in our case where a specific *generative* model can be devised, i.e. a model that for a specific set of parameters offers a prediction and a related likelihood expression for its observations. When multiple models are used for feature extraction, a posterior distribution can be defined using these likelihood expressions based on Bayes' rule, thereby indicating which of these most accurately captures the observations.

Below we briefly review our contributions [18,20] in providing such a framework for the DCA model and subsequently provide a modification of the Region Competition/GAR algorithm that allows for the fusion of the extracted features in a simple and efficient manner.

4.1 Generative Models for Features

The generative model we introduce for texture explains the image neighborhood around a specific point in terms of two sub-models, namely a texture model and a generic background model; the degree to which these two sub-models contribute to the explanation of an observation depends on its distance from the point around which the model is defined. In all of the following the treatment will consider the signals as one-dimensional, for the sake of notational clarity. The synthesized prediction $S^T(x)$ at point x of the texture sub-model defined around point 0 can be expressed as the sum of a sinusoidal of fixed frequency ω and phase offset ϕ and a DC component B: $S^T(x) = A\cos(\omega x + \phi) + B$, while the background sub-model is considered to be a uniform distribution. The spatial variation of our confidence in these two sub-models can be phrased as:

$$P(I(x)|x,0) = G(x)P(I(x)|S^T(x)) + (1 - G(x))c, \tag{11}$$

where $P(I(x)|x,0)$ is the likelihood of the observation at point x, given the model related to the texture hypothesis that is defined around point 0; $G(x)$ accounts for the locality of the modeling process and is taken to be a Gaussian function normalized so that $G(0) = 1$. The corruption of the texture synthesis $S^T(x)$ is modelled by $P(I(x)|S^T(x))$ which is a Gaussian distribution with mean $S^T(x)$ and unknown, constant variance while c is the constant term contributed by the uniform background model. Using the quadrature pair $g_e = \cos(\omega x)G(x)$ and $g_o = \sin(\omega x)G(x)$ of Gabor filters centered around frequency ω, in [18,20] it is shown that the quantity $A_k = \sqrt{(g_e * I)^2 + (g_p * I)^2}$ is related to the likelihood of the image observations under the probabilistic model outlined above. In this sense, picking the channel with the largest amplitude estimate amounts to choosing the channel that explains the data best.

To apply the same rationale to edge detection, we used the observation that edges are *phase congruent* signals [27], i.e. signals of the form $A\sum_k a_k \cos(\omega k + \phi)$; for $\phi = 0$ we derive line (triangular) edges and for $\phi = \frac{\pi}{2}$ step edges. Again, using an appropriate quadrature filter pair of odd and even filters we can derive an amplitude estimate related to the likelihood of the observations under this model [20]. A model that complements these two classes is the smooth signal model, for which we use the locally DC signal, using again an expression of the form Eq. (11) to account for the locality of the decision made.

Using these three models we are able to estimate the probability of a pixel having been generated by one of the three hypotheses considered based on Bayes' rule. Applying this approach on natural images gives visually appealing results as shown in Fig. 2, since textured areas are correctly discriminated from edges, matching closely what a human would call a texture or an edge. Given that the previously described models

do not take into account higher order structure or grouping cues, the results are quite satisfactory. A more extensive presentation of the approach outlined above can be found in [18,20].

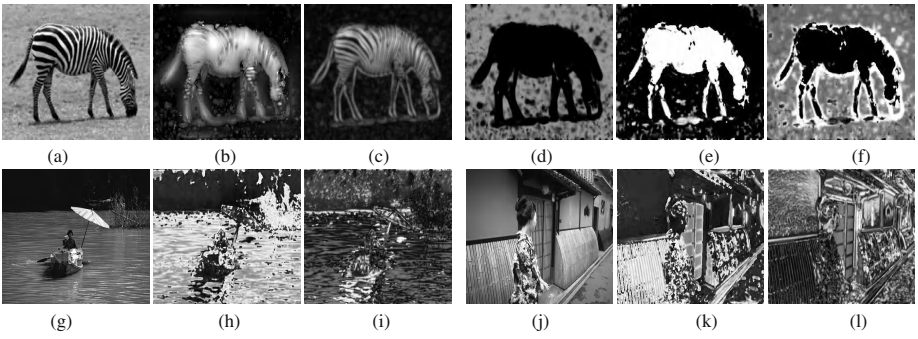


Fig. 2. From [20]: Textured areas can be successfully discriminated from edges, using simple generative models. Top row: (a) Input image, (b) texture and (c) edge model amplitude estimates, respectively and model-based probabilities of (d) smooth, (e) textured and (f) edge regions respectively. Bottom row: (g)/(j) Input images and posterior probabilities of (h)/(k) texture and (i)/(l) edge models respectively.

4.2 Cue Integration for Region Competition

In this section we present a modification of the Region Competition algorithm that can exploit the confidence measures assigned to the feature set used to drive the segmentation process. This fits naturally with the models used for feature extraction in our work, but its application is not constrained to these.

The rationale underlying the Region Competition evolution equations is to assign to each region the observations that it can most accurately explain in terms of its distribution. This is implemented using a probabilistic balloon force [6], that pushes the front of region i along its outward normal with a force proportional to:

$$\mathcal{O}_I = \log \frac{P_i(I; a_i)}{P_j(I; a_j)},$$

where j is the competing neighboring region. This quantity, termed $log - odds$ quantifies the degree to which the observation I is more likely under hypothesis i than under hypothesis j. Supra - Bayesian fusion methods [14] consider this quantity $\mathcal{O}_I$ as a random variable, which follows a Gaussian distribution conditioned on the actual class of the data I:

$$P(\mathcal{O}_I|i) \propto N(\mu_i, \sigma^2), \quad P(\mathcal{O}_I|j) \propto N(\mu_j, \sigma^2)$$

This quantifies the certainty associated with any decision made based on $\mathcal{O}_I$: a large σ, i.e. a low confidence in log-odd accuracy can diminish the effect of a large value of $\mathcal{O}_I$. From a good classifier we would generally expect that $\mu_i >> \mu_j$ with a low σ, which

means that if the data I are due to hypothesis i, then it is very probable that $\mathcal{O}_I$ will take a high value and vice versa.

Based on the above approach, the results of N classifiers using different features or different classification methods can be easily integrated [14]: their log-odds $\boldsymbol{\mathcal{O}} = [\mathcal{O}_1, \ldots, \mathcal{O}_N]$ are viewed as a multidimensional random variable that follows a Gaussian distribution conditioned on the class of the data i.e.

$$P(\boldsymbol{\mathcal{O}}|i) \propto N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}), \quad P(\boldsymbol{\mathcal{O}}|j) \propto N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma})$$

The posterior log-likelihood ratio given all the expert odds is then:

$$\log \frac{P(i|\boldsymbol{\mathcal{O}})}{P(j|\boldsymbol{\mathcal{O}})} = (\boldsymbol{\mathcal{O}} - \frac{\boldsymbol{\mu}_i + \boldsymbol{\mu}_j}{2})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

We thereby build a decision using as features the outputs of classifiers, instead of the actual features. For the special case where the classifier outputs are uncorrelated, we have a diagonal covariance matrix; further, by appropriately scaling and shifting the classifier outputs we can guarantee that $\boldsymbol{\mu}_i = -\boldsymbol{\mu}_j$ and all the elements of the $\boldsymbol{\mu}_i$ vector equal unity, so that we have

$$\log \frac{P(i|\boldsymbol{\mathcal{O}})}{P(j|\boldsymbol{\mathcal{O}})} = \sum_c \frac{\mathcal{O}_c}{\sigma_c^2}$$

This formula expresses a straightforward idea: when a classifier gives noisy results, i.e. has a large σ, a lower weight should be assigned to his decision and vice versa. The use of log-odds is particularly convenient, in that we express this weighting operation in terms of a summation.

In our case, we consider that the assignment of an observation to region i or region j is a decision taken by fusing the assessments of two experts, where the decision of the first is based on the texture features and that of the second on the intensity values. Using the previous notation we have

$$\mathcal{O}_T = \log \frac{P^T(F_T; a_i^T)}{P^T(F_T; a_j^T)}, \quad \mathcal{O}_S = \log \frac{P^S(F_S; a_i^S)}{P^S(F_S; a_j^S)}$$

where by $P^M(F_M; a_i^M)$ we denote the likelihood of the feature-set F_M extracted based on hypothesis M (texture-T or smooth-S) under the hypothesis-specific distribution P^M of region i, whose parameters are a_i^T. For the final decision, each expert's opinion is weighted by the probability of each hypothesis, estimated as described in the previous subsection; this way for textured areas the texture features have a larger impact on the evolution of the curve than the intensity features, and vice versa for smooth regions. Equation 6 thus becomes:

$$\frac{\partial \Gamma_i}{\partial t} = \left[\sum_{c \in T,S} w_c \log \frac{P^c(I_c; \alpha_i)}{P^c(I_c; \alpha_j)} - \mu\kappa \right] \mathcal{N} \tag{12}$$

where the index c ranges over the cues that are being fused and w_c is the related cue weight. Further, as in [33] one can introduce edge based information, but now exploiting the model-based probability of edge in which case the evolution equation becomes:

$$\frac{\partial \Gamma_i}{\partial t} = \left[\sum_{c \in T,S} w_c \log \frac{P^c(I_c;\alpha_i)}{P^c(I_c;\alpha_j)} \right] \mathcal{N} - w_e \left[g\kappa\mathcal{N} + (\nabla g \cdot \mathcal{N})\mathcal{N} \right] \tag{13}$$

with w_e being the probability of the edge hypothesis. This does not interfere with the edge detection procedure used to estimate ∇g, since it simply provides an indication of how important the edge information should be deemed.

We note here that the weights entering the fusion equations do not have to be estimated using the specific models described previously; as an alternative that we intend to explore in future work, the celebrated U+V decomposition [44] could be used to indicate regions with a strong texture component, and subsequently provide weights for a fusion algorithm.

5 Experimental Evaluation

As described in [19], curve evolution is implemented using level-set methods [41,42] along the lines of [33]. As in [40,38,45] the distribution of the data inside each region is learned in parallel with the evolution process, resulting in an adaptive scheme. For all the results presented in this work the regions have been initialized so as to partition the whole image in interleaved thin parallel strips, while we have observed that the results do not depend substantially on the initialization.

In Fig. 3 (a)-(d) we present the modulation features extracted via DCA and segmentation results using four alternative schemes: In Fig. 3(e) the results of curve evolution along the lines of [19] are shown, using the 3-dimensional DCA-based texture descriptor while in (f) we show results using the cue integration scheme described in this work. In (g) the orientation channel is added to the feature vector, with no performance degradation on smooth areas, contrary to (e), and improved boundary localization. In the

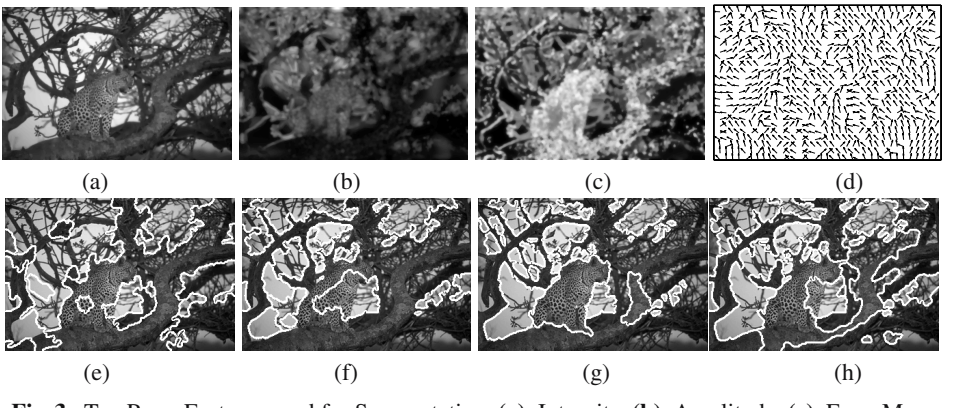
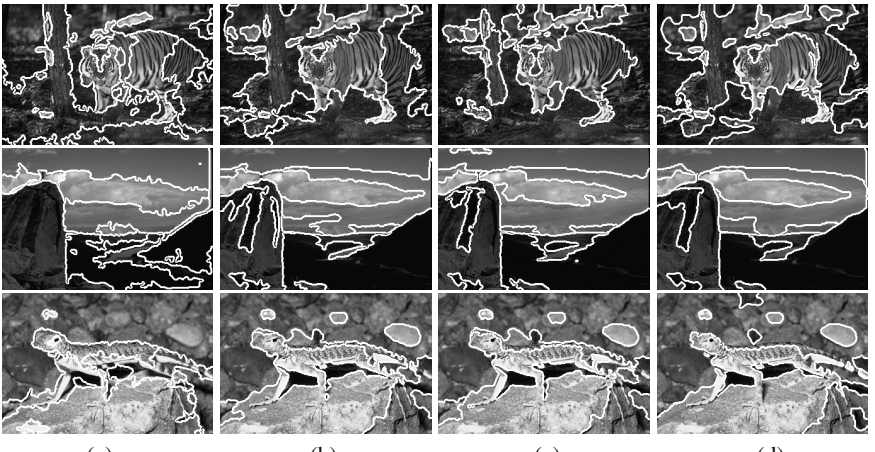


Fig. 3. Top Row: Features used for Segmentation. **(a)** Intensity, **(b)** Amplitude, **(c)** Freq. Magnitude, **(d)** Freq. Orientation vectors, Bottom Row: Segmentation results, using **(e)** the original evolution equations [46,32] as in [19], **(f)** the fusion evolution equations, excluding the orientation channel and **(g)** including the orientation channel, **(h)** Diffusion-based features [38] and the original evolution equations. Please see text for details.

fused cases, (f) and (g), the region boundaries tend to accurately capture the object borders, while in (e) the erroneously estimated texture features prevent them from doing so. Further results are provided in Fig. 4, where we generally observe that the fused features give better segmentations results, with the region borders accurately locating the object borders. The effect of the orientation features which behave erratically at smooth areas is diminished, due to the smaller weight assigned to the texture hypothesis at these regions.



(a) (b) (c) (d)

Fig. 4. Segmentation Comparisons: **(a)** original evolution equations, **(b)** the fusion evolution equations, excluding the orientation channel and **(c)** including the orientation channel, **(d)** Diffusion-based features [38]

In order to obtain quantitative results, segmentation results for an increasing numbers of fronts were derived for the whole Berkeley test set and compared to human segmentations based on the Bidirectional Consistency Error (BCE) measure introduced in [25]. This measure quantifies in a smooth manner the overlap between a machine generated segmentation and a set of manual segmentations, and is minimized when for every machine-generated segment there is at least one human-generated segmentation wherein the segment is contained as a whole and vice versa.

Initially we compared the performance of the raw AM-FM features used in [19] to that of the nonlinear diffusion-based feature set [38]; in all the related results presented herein some uncertainty is retained due to potential inaccuracies in the implementation of the algorithms in [38]. For this comparison, the orientation features were omitted, in order to avoid spoiling the feature vector at smooth regions. Comparing the histograms in Fig.s. 5(a)(b) the BCE can be seen to be lower for the AM-FM features, indicating their potential to accurately describe a textured region in terms of a low-dimensional feature vector. As already mentioned, the introduction of the orientation channel in the feature vector typically results in segmentation performance degradation in the cases of smoothly varying areas and edges; a point that we make in this paper is that using

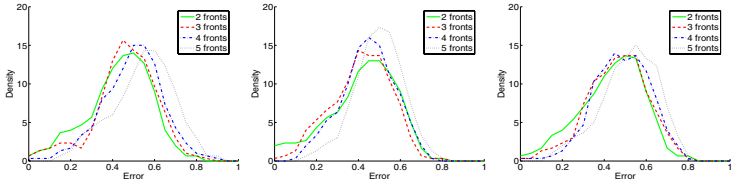


Fig. 5. Berkeley Benchmark evaluation: Histogram of Bidirectional Consistency Error [25] over 100 test images for varying number of fronts (2-5) for: **(a)** Diffusion-based features **(b)** Unfused data, without orientation **(c)** Fused data with orientation. Please see text for details.

a fusion algorithm counteracts these defects and allows the orientation channel to be incorporated in a manner that is immune to its behavior on smooth regions. This can be seen by a comparison between Figs. 5(b)(c), where the BCE is seen to have almost the same distribution; that the introduction of the orientation vector does not lead to a systematic improvement in the results can be due the fact that the data set provided in [26] does not contain many heavily textured images, where the information carried by the orientation channel is most valuable.

6 Conclusions

Multicomponent AM-FM models propose a powerful approach to the representation, analysis and segmentation of textured images. Our contributions presented herein lie in (1) the introduction of a regularized demodulation algorithm that can alleviate discretization problems and introduce increased noise robustness and (2) the probabilistic integration of features related to different image models, based on a modification of the region competition evolution equations. Systematic comparisons have demonstrated that the derived features compare favorably to those used by the current state-of-the-art methods, indicating the appropriateness of modulation features for unsupervised textured image segmentation.

References

1. Bovik, A., Clark, M., and Geisler, W. Multichannel Texture Analysis using Localized Spatial Filters. *IEEE Trans. PAMI 12*, 1 (1990), 55–73.
2. Bovik, A. C., Gopal, N., Emmoth, T., and Restrepo, A. Localized Measurement of Emergent Image Frequencies by Gabor Wavelets. *IEEE Trans. Information Theory 38* (1992), 691–712.
3. Bovik, A. C., Maragos, P., and Quatieri, T. F. AM-FM Energy Detection and Separation in Noise Using Multiband Energy Operators. *IEEE Trans. Signal Processing 41* (1993), 3245–3265.
4. Brox, T., and Weickert, J. A TV Flow based Local Scale Measure for Texture Discrimination. In *ECCV* (2004).
5. Caselles, V., Kimmel, R., and Sapiro, G. Geodesic Active Contours. *Int.l. J. of Comp. Vision 22*, 1 (1997), 61–79.

6. COHEN, L. On Active Contour Models and Balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding 53*, 2 (1991), 211–218.
7. DAUGMAN, J. Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filters. *Journal of the Optical Society of America 2*, 7 (1985), 160–169.
8. DIMITRIADIS, D., AND MARAGOS, P. Robust Energy Demodulation Based on Continuous Models with Application to Speech Recognition. In *Eurospeech* (2003).
9. GUMBEL, E., GREENWOOD, A., AND DURAND, D. The Circular Normal Distribution: Theory and Tables. *Journal of the American Statistical Association 48*, 261 (1953), 131–152.
10. HAVLICEK, J., AND BOVIK, A. Image Modulation Models. In *Handbook of Image and Video Processing*, A. Bovik, Ed. 2000, pp. 305–316.
11. HAVLICEK, J., HARDING, D., AND BOVIK, A. The Multi-Component AM-FM Image Representation. *IEEE Trans. Im. Proc. 5*, 6 (1996), 1094–1100.
12. HAVLICEK, J., HARDING, D., AND BOVIK, A. Multidimensional Quasi-Eigenfunction Approximations and Multicomponent AM-FM Models. *IEEE Trans. Im. Proc. 9*, 2 (2000), 227–242.
13. HEILER, M., AND SCHNORR, C. Natural Image Statistics for Natural Image Segmentation. *Int.l. J. of Comp. Vision 63*, 1 (2005), 5–19.
14. JACOBS, R. Methods for Combining Experts' Probability Assesements. *Neural Computation*, 7 (1995), 867–888.
15. KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active Contour Models. In *ICCV* (1987).
16. KICHENASSAMY, S., KUMAR, A., OLVER, P., TANNENBAUM, A., AND YEZZI, A. Gradient Flows and Geometric Active Contour Models. In *ICCV* (1995).
17. KIM, J., FISHER, J., YEZZI, A., CETIN, M., AND WILLSKY, A. Nonparametric Methods for Image Segmentation using Information Theory and Curve Evolution. In *Int.l Conf. on Image Processing* (2002).
18. KOKKINOS, I., EVANGELOPOULOS, G., AND MARAGOS, P. Advances in Texture Analysis: Energy Dominant Components and Multiple Hypothesis Testing. In *Int.l Conf. on Image Processing* (2004).
19. KOKKINOS, I., EVANGELOPOULOS, G., AND MARAGOS, P. Modulation-Feature based Textured Image Segmenation Using Curve Evolution. In *Int.l Conf. on Image Processing* (2004).
20. KOKKINOS, I., AND MARAGOS, P. A Detection-Theoretic Approach to Texture and Edge Discrimination. In *4th International Workshop on Texture Analysis and Synthesis* (2005).
21. LEE, T. S., MUMFORD, D., AND YUILLE, A. Texture Segmentation by Minimizing Vector Valued Energy Functionals. In *ECCV* (1992).
22. MALIK, J., AND PERONA, P. Preatentive Texture Discrimination with Early Vision Mechanisms. *J. Opt. Soc. Amer. (A) 7*, 5 (1990), 923–932.
23. MALLADI, R., SETHIAN, J., AND VEMURI, B. Shape Modeling with Front Propagation: A Level-Set Approach. *IEEE Trans. PAMI 17*, 2 (1995), 158–175.
24. MARAGOS, P., AND BOVIK, A. Image Demodulation Using Multidimensional Energy Separation. *Journal of the Optical Society of America 12*, 9 (1995), 1867–1876.
25. MARTIN, D. *An Empirical Approach to Grouping and Segmentation*. PhD thesis, Univ. California at Berkeley, 2002.
26. MARTIN, D., FOWLKES, C., TAL, D., AND MALIK, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *ICCV* (2001).
27. MORRONE, C., AND BURR, D. Feature Detection in Human Vision: a Phase-Dependent Energy Model. *Proceedings of the Royal Society of London B 235* (1988), 221–245.

28. MUMFORD, D., AND SHAH, J. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *Communications on Pure and Applied Mathematics 42*, 5 (1989), 577–685.
29. OSHER, S., AND SETHIAN, J. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics 79* (1988), 12–49.
30. PAPOULIS, A. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1984.
31. PARAGIOS, N., AND DERICHE, R. Geodesic Active Contours for Supervised Texture Segmentation. In *CVPR* (1999).
32. PARAGIOS, N., AND DERICHE, R. Geodesic Active Regions: A new Paradigm to Deal with Frame Partition Problems in Computer Vision. *Journal of Visual Communication and Image Representation 13*, 1 (2002), 249–268.
33. PARAGIOS, N., AND DERICHE, R. Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation. *Int.l. J. of Comp. Vision 46*, 3 (2002), 223–247.
34. PARAGIOS, N., AND OSHER, S. *Geometric Level Set Methods in Imaging, Vision & Graphics*. 2002.
35. POGGIO, T. A., VOORHEES, H., AND YUILLE, A. A Regularized Solution to Edge Detection. *J. Complexity 4*, 2 (1988), 106–128.
36. POTAMIANOS, A., AND MARAGOS, P. A Comparison of the Energy Operator and Hilbert Transform Approaches for Signal and Speech Demodulation. *Signal Processing 37*, 1 (1994), 95–120.
37. RAY, N., HAVLICEK, J., ACTON, S., AND PATTICHIS, M. Active Contour Segmentation Guided by AM-FM DCA. In *Int.l Conf. on Image Processing* (2001).
38. ROUSSON, M., BROX, T., AND DERICHE, R. Active Unsupervised Texture Segmentation on a Diffusion-Based Feature Space. In *CVPR* (2003).
39. SAGIV, C., SOCHEN, N. A., AND ZEEVI, Y. Y. Texture Segmentation via a Diffusion-Segmentation Scheme in the Gabor Feature Space. In *2nd International Workshop on Texture Analysis and Synthesis* (2002).
40. SANDBERG, B., CHAN, T., AND VESE, L. A Level-Set and Gabor-based Active Contour Algorithm for Segmenting Textured Images. Tech. Rep. 02-39, UCLA CAM, 2002.
41. SETHIAN, J. *Level Set Methods*. Cambridge University Press, 1996.
42. SETHIAN, J. A. A Fast Marching Method for Monotonically Advancing Fronts. *Proc. Nat.l Academy of Sciences of USA 93*, 4 (1996), 1591–1595.
43. VESE, L., AND CHAN, T. Active Contours Without Edges. *IEEE Trans. Im. Proc. 10*, 2 (2001), 266–277.
44. VESE, L., AND OSHER, S. J. Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing. *Journal of Scientific Computing 19*, 1-3 (2003), 553–572.
45. YEZZI, A., TSAI, A., AND WILLSKY, A. A Statistical Approach to Snakes for Bimodal and Trimodal Imagery. In *ICCV* (1999).
46. ZHU, S., AND YUILLE, A. Region Competition: Unifying Snakes, Region Growing and Bayes/MDL for Multiband Image Segmentation. *IEEE Trans. PAMI 18*, 9 (1996), 884–900.

Entropy Controlled Gauss-Markov Random Measure Field Models for Early Vision*

Mariano Rivera, Omar Ocegueda, and Jose L. Marroquin

Centro de Investigacion en Matematicas A.C.,
Guanajuato, Gto., Mexico 36000
{mrivera, omar, jlm}@cimat.mx

Abstract. We present a computationally efficient segmentation–restoration method, based on a probabilistic formulation, for the joint estimation of the label map (segmentation) and the parameters of the feature generator models (restoration). Our algorithm computes an estimation of the posterior marginal probability distributions of the label field based on a Gauss Markov Random Measure Field model. Our proposal introduces an explicit entropy control for the estimated posterior marginals, therefore it improves the parameter estimation step. If the model parameters are given, our algorithm computes the posterior marginals as the global minimizers of a quadratic, linearly constrained energy function; therefore, one can compute very efficiently the optimal (Maximizer of the Posterior Marginals or MPM) estimator for multi–class segmentation problems. Moreover, a good estimation of the posterior marginals allows one to compute estimators different from the MPM for restoration problems, denoising and optical flow computation. Experiments demonstrate better performance over other state of the art segmentation approaches.

1 Introduction

Image segmentation from different attributes (such as gray level, local orientation or frequency, texture, motion, color, etc.) is typically formulated as a clustering problem. Although generic clustering algorithms as K-Means or ISODATA have been used with relative success [8], the consideration of spatial interactions among pixel labels provides additional, useful constraints on the problem [2][3] [6][7] [9][10] [11][12] [13][14] [17][18] [19][20] [21][22] [23][24].

Therefore, most successful algorithms for image segmentation taked into account both the observed pixel values (which are generally noisy) and pixel context information.

We assume that an image of features, g, in the regular lattice L, is an assembly of K disjoint regions, $R = \{R_1, R_2, ..., R_K\}$. Moreover, such features are generated with a generic parametric model ϕ, with $\theta = \{\theta_1, \theta_2, ..., \theta_K\}$ as the corresponding parameters for each region, i.e.:

$$g_r = \sum_{k=1}^{K} \phi_{kr} b_{kr} + \eta_r \tag{1}$$

* This work was partially supported by CONACYT, Mexico (grants 40722 and 46270).

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 137–148, 2005.

where $r = [x, y]^T$ is the position of a pixel in the regular lattice L; $\phi_{kr} \stackrel{def}{=} \phi(\theta_k, r)$ is the parametric model of the actual value of the feature at pixel r; θ_k is the parameter set corresponding to the k^{th} region; b_{kr} is an indicator variable equal to one if the pixel r was generated with the model k and equal to zero otherwise and η is an additive, independent, identically distributed noise process. In the general case, the one we consider here, the parameter set is unknown and needs to be estimated.

Bayesian regularization framework has been successfully used for finding the solution to these problems [6][10] [11][12] [13][14] [18] [19][20] [21][22] [24][25]. In this framework, the solution is computed as a statistical estimator from a posterior probability distribution. In particular, one needs to estimate the label map, b, and the model parameters, θ, from the posterior distribution $P_{b,\theta|g}$. If it is assumed independence between b and θ and a uniform prior distribution for θ, then this posterior distribution is given by:

$$P_{b,\theta|g} = P_{g|b,\theta} P_b / P_g \tag{2}$$

where the likelihood of the whole label field is obtained from a mixture model :

$$P_{g|b,\theta} = \prod_k \prod_r (v_{kr})^{b_{kr}};$$

with v_{kr} as the probability (individual likelihood) that the observed value at pixel r was generated with model k (that uses the set of parameters θ_k). For instance, if η is Gaussian with zero mean and variance σ^2:

$$v_{kr} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(g_r - \phi_{kr})^2}{2\sigma^2}\right], \tag{3}$$

for a real valued feature image g. In the framework of Bayesian Estimation Theory, b is modelled as a Markov Random Field (MRF) with prior distribution, P_b, in the form of a Gibbs distribution:

$$P_b = \frac{1}{z} \exp\left[-\beta \sum\nolimits_C V_C(b)\right]; \tag{4}$$

where z is a normalization constant and V_C is a potential such that it assigns larger probabilities to smooth label fields than to granular ones and β is a positive parameter. The most popular potential is the Ising model:

$$V_{krs}(b) = \begin{cases} -1 \; if \; b_{kr} = b_{kr} \;\; \forall k \\ \;\; 1 \; if \; b_{kr} \neq b_{ks}. \end{cases} \tag{5}$$

Finally, P_g is a normalization constant independent of the unknowns b and θ.

In most cases, approximate solutions for this complex estimation problem are found by 2-step procedures,[2][4][12][14][20] in which the best segmentation, given the parameters is found in the first step, and the optimal estimator for the parameters, given the segmentation is found in the second step, iterating these 2 steps until convergence. Usually, one chooses the models, ϕ in such a way that the maximum a posteriori (MAP) estimator for the parameters θ given b is relatively easy to compute. For instance, flat, planar or spline models have successfully been used [12][14][18]. However, the MAP

estimator for the label field requires the solution of a combinatorial optimization problem. Graph–Cuts based algorithms[2][3] [7][19][23] can be used to compute the exact MAP estimator in the case of binary segmentation or an approximation for problems with more than two classes. The problem here is that using a "hard" segmentation in the first step of the procedure, makes the 2-step algorithm prone to get trapped in local minima, producing suboptimal results. A better strategy is to compute, instead of a set of binary indicator variables, their expected value (i.e., the posterior marginal distributions), in which case the 2-step procedure is equivalent to the Expectation-Maximization (EM) algorithm [4][15][20]. Upon convergence, a hard segmentation may be computed, if desired, using, for instance, the MPM estimator [11]:

Definition 1 (MPM Estimator). *The MPM estimator of the label field is given by:*

$$b_{kr}^{MPM} = \begin{cases} 1 \text{ if } \pi_{kr} \geq \pi_{lr}, \text{ for } k \neq l \\ 0 \text{ otherwise}, \end{cases} \tag{6}$$

where π_r is the marginal probability distribution of the pixel r.

An estimation, p, of the true marginals, π, can be computed with Markov Chain Monte Carlo (MCMC) based methods [6][10]. In such a case, samples $\{b^{(1)}, b^{(2)}, \ldots, b^{(N)}\}$, of the posterior distribution, $P_{b,\theta|g}$, are used to compute the empirical marginals: $p_{kr} = \frac{1}{N}\sum_{j=1}^{N} b_{kr}^{(j)}$, that satisfy

$$E[p_{kr}] = \pi_{kr} \tag{7}$$

where $E[\cdot]$ denotes the expectation operator. The problem with these methods lies in their high computational cost.

A more efficient approach considers the empirical marginals as a vector-valued random field (i.e., a random measure field) that needs to be modeled. There are 2 main models that have been proposed: one based in a Mean Field (MF) approximation [20][24], and the other in a Gauss-Markov Measure Field (GMMF) model [13]. Both of them, however, have certain drawbacks: the MF approach leads to algorithms that are relatively slow and sensitive to noise, while the GMMF approach presented in [13] produces estimators for the marginal distributions that differ significantly from the true ones, in the sense that these distributions (one for each pixel) have significantly higher entropy than the ones found asymptotically by MCMC approaches. If the model parameters are known, this difference is not too important, since usually the modes of these distributions (and hence, the MPM estimator) are correct; if the model parameters are unknown, however, this high entropy produces an unstable behavior of the EM algorithm, producing bad results. The goal of this paper is to present a better GMMF model for the empirical marginals that produces estimates that are in agreement with the true ones, and that can be efficiently computed.

2 Entropy Controlled GMMF (EC-GMMF) Models

The use of GMMF models for estimating the posterior marginal distributions is based in the following:

Theorem 1 (Gauss-Markov Measure Fields (GMMF)). *Let the binary label field b be a Markov random field (MRF) with posterior distribution (2) and v the likelihood field, then the empirical marginal field, p, is itself a MRF with posterior distribution:*

$$P_{p|v} = P_{v|p} P_p / P_v \tag{8}$$

with the following properties:

1. *It is Markovian with the same neighborhood system as b.*
2. *It is Gaussian, i.e. $P_{p|v} \propto \exp[-U(p; v)]$, where the energy $U(p; v)$ is a Quadratic, Positive Definite (QPD) function of p.*

The proof is presented in [13]. This theorem establishes important properties of the marginal probabilities, π, of the label field, b, with posterior distribution (2), but it does not determine the exact form of the QPD energy U. Given (7), and Theorem 1, π can be estimated as the MAP estimator of (8). In order to find a particular form for U, an additional consistency constraint is imposed in [13]:

Consistency Constraint 1 (GMMF). *If no prior information is provided [i.e. P_π is the uniform distribution] then the maximizer of (8) is $\pi^* = \bar{v}$, where $\bar{v}_{kr} = v_{kr} / \sum_j v_{jr}$.*

Based on these properties, the function U that is proposed in [13] is:

$$U(p; v) = \sum_{k=1}^{K} \sum_{r \in L} \left[(p_{kr} - \bar{v}_{kr})^2 + \frac{\lambda}{2} \sum_{s \in \mathcal{N}_r} \|p_{kr} - p_{ks}\|_2^2 \right], \tag{9}$$

where $\mathcal{N}_r = \{s \in L : |r - s| = 1\}$ is the set of nearest neighbor pixels to r. In spite of the fact that the minimization of (9) can be done with computationally efficient algorithms, the use of this equation has two disadvantages: first, it depends on the model parameters (via the likelihoods v) in a highly non-linear way (3), which makes difficult its incorporation in EM procedures, and second, the set of distributions ($\{p_r\}$ field) that minimizes (9) are relatively flat (i.e., they have high entropy), which makes them unsuitable for EM procedures.

In order to propose a new QPD function U that overcomes these difficulties, we need to relax the consistency constraint 1; the new constraint is:

Consistency Constraint 2 (EC-GMMF). *If no prior information is provided, the mode of the optimal estimators for the posterior marginal distributions π^* coincides with the maxima of the corresponding likelihoods v, i.e., the MPM estimator for the label field, computed using π^* coincides with the maximum likelihood estimator.*

With this relaxed constraint, we may introduce $\log v$ instead of v in the data term, to get a quadratic dependence on the model parameters θ (assuming a Gaussian noise model). Entropy control is introduced by adding a penalization term of high entropy distributions; to keep quadratic the energy function, we use the Gini coefficient [5] (instead of the Shannon's entropy [22]): $-\sum_k \sum_r p_{kr}^2$; so that we finally get:

$$U_{EC}(p, \theta) = \sum_k \sum_r \left[p_{kr}^2 (-\log v_{kr} - \mu) + \frac{\lambda}{2} \sum_{s \in \mathcal{N}_r} (p_{kr} - p_{ks})^2 \right] \tag{10}$$

subject to

$$\sum\nolimits_k p_{kr} = 1, \ \ \forall r \ \ \text{and} \ \ p_{kr} \geq 0, \ \ \forall k, r; \tag{11}$$

where the parameter μ controls the entropy of the marginals. Note that for $\mu < 2\lambda$, we can assure that (10) is QPD. It is found that the performance of the estimation algorithms does not depend critically on the precise value for μ; we have used $\mu = 0.1\lambda$ in all the experiments reported here.

2.1 Computation of the Optimal Estimators

The minimization of (10) may be carried out by a 2-step (EM) procedure. The MAP estimators for the marginals, p^{MAP}, and the models, θ^{MAP}, are computed by iterating alternated minimizations of (10) w.r.t. p and θ, respectively. These minimizations have the following interesting properties:

Theorem 2 (EC-GMMF Convergence). *Assuming that U_{EC} is QPD with respect to p and V_{kr} is a uni-modal distribution, we have:*

(i) If θ is given, the problem of minimizing U_{EC} w.r.t. p, subject to the constraints (11) has a unique local minimum, which coincides with the constrained global minimum.
(ii) If p is given, the problem of minimizing U_{EC} w.r.t. θ has a unique local minimum, which coincides with the global minimum.
(iii) The iterated alternate minimizations of U_{EC} w.r.t. p and θ converges, at least, to a local minimum.

The proof of (i) and (ii) follows from the facts that U_{EC} is a QPD function of p for fixed θ, and of θ for fixed p, and the constraints (11) are linear (see [16]). (iii) Follows from (i) and (ii) and from the fact that $U_{EC} \geq 0$.

Last theorem establishes that any descent algorithm will converge to the global minimum in the E and M steps of the EM procedure. In particular, if the model parameters are given, one may find very efficiently the optimal (MPM) segmentation even for multi-class segmentation problems, which represents a significant advantage over algorithms like those based on graph cuts, which guarantee global optimality only for 2-class problems.

In the algorithm we propose here, the equality constraint in (11) may be incorporated using the Lagrangian method: the Lagrangian, that incorporates the equality constraints, is given by:

$$\mathcal{L}(p, \theta) = U_{EC}(p, \theta) - \sum_r \gamma_r \big(1 - \sum\nolimits_k p_{kr}\big) \tag{12}$$

where γ are the Lagrange multipliers of the equality constraints. Now, we define $n_{kr} \stackrel{def}{=} \lambda \sum_{s \in \mathcal{N}_r} p_{ks}$ and

$$m_{kr} \stackrel{def}{=} (-\log v_{kr} - \mu) + \lambda \sharp \mathcal{N}_r. \tag{13}$$

Algorithm 1. Gauss-Seidel Implementation of Parametric Segmentation

1: Set the initial model parameters θ_0 and initially set $p_0 = v$;
2: Given the tolerance $\epsilon > 0$;
3: **for** $i = 1, 2, \ldots$ **do**
4: **for all** the pixels r **do**
5: **for all** the models k **do**
6: Compute p_{ikr} with (14);
7: Project $p_{ikr} = \max\{0, p_{ikr}\}$;
8: **end for** all the models
{The renormalization of the p_{ir} can be performed here}
9: **end for** all the pixels
10: Update the models θ_i with (15);
11: **if** $\|p_i - p_{i-1}\| < \epsilon$ **then**
12: STOP with solution $p^*_{kr} = p_{ikr}$ and $\theta^* = \theta_i$;
13: **end if**
14: **end for**

where $\sharp S$ is the cardinality of the set S. Equating to zero the gradient of (12) w.r.t. p, solving for p_{kr} and substituting in the equality constraint (11), one finally obtains the Gauss-Seidel update equation:

$$p_{kr} = \frac{n_{kr}}{m_{kr}} + \frac{1 - \sum_{l=1}^{N} \frac{n_{lr}}{m_{lr}}}{\sum_{l=1}^{N} \frac{m_{kr}}{m_{lr}}}. \tag{14}$$

Note, however, that p_{kr} computed with (14) does not necessarily satisfy the non-negativity constraint in (11). If such a constraint is violated, one makes the negative p_{kr} equal to zero, and renormalizes the vector p_r. In our experiments we found that this simple method works properly and is faster than more sophisticated methods(such as gradient projection).

In practice, one gets better performance, in terms of computational efficiency, if the θ variables are updated after every Gauss-Seidel iteration, instead of waiting until convergence of the E step, i.e. by using a Generalized EM algorithm (GEM) [15]. One gets then a direct procedure, in which the p and θ variables are simultaneously optimized.

Given that θ is not constrained, it may be computed, after every update of the p field, by

$$\theta^{MAP} = \arg\min_{\theta} \sum_{k} \sum_{r} \left[-p_{kr}^2 \log v_{kr}(\theta)\right]. \tag{15}$$

The complete procedure is summarized in algorithm 1. Line 10 in such an algorithm is generic and depends on the specific feature model; in the next section we present a particular case.

3 Experiments

The purpose of the experiments in subsection 3.1 is to evaluate the relative performance of the EC-GMMF model with respect to other state of the art segmentation methods.

Three aspects are evaluated: Noise robustness, computational efficiency and entropy control. In subsection 3.2 we show an application of our method to other Computer Vision problems: Image Denoising and Optical Flow estimation. We also show that using the same p^* obtained, we can directly compute different estimators that serve as solutions to Piecewise Constant and Piecewise Smooth regularization.

3.1 Numerical Experiments

The normalized test images (two models synthetic image and Lenna's portrait) were corrupted with additive Gaussian noise with mean zero and variance 0.30 and 0.05 respectively. The first experiment, illustrated by Fig. 1, demonstrates the robustness of

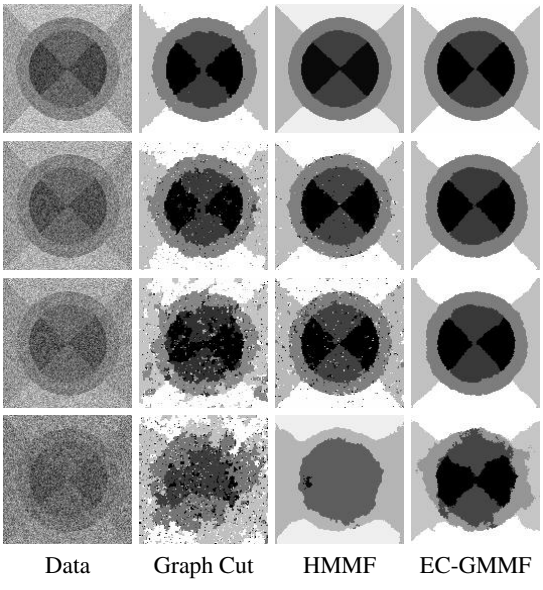


Fig. 1. Segmentation method performance for different level noise, see text for details

the EC-GMMF method to noise. The task is to segment and estimate the models' parameters for a five models synthetic image with levels 1,2,3,4 and 5. First column in Fig. 1 shows the data corrupted with Gaussian noise with: $\sigma = 0.7, 1.0, 1.2, 2.0$, respectively. The second column shows the segmentations computed with a multi–way graph cut based algorithm. This algorithm has the drawback of being based on a MAP criterion, and it is known that for low SNR data the MPM estimator exhibits better performance. The third column shows the results computed with other state of the art parametric segmentation method, namely HMMF, which in [14] is shown to have better performance than the Mean Field and MCMC-based EM procedures. Our results are consistent with the ones reported by the authors [14]: HMMF models are more robust to noise than graph-cuts based algorithms; however, we found that HMMF algorithm is very sensitive to the precise selection of the parameters' values: the noise variance,

the regularization parameter, the initial p–field values and the minimization algorithm parameters (i.e. the friction coefficient and the step size for the gradient projection Newtonian descent algorithm [14]). In part, such a difficulty lies in the fact that the energy function, in the HMMF model, is not convex, which makes the descent algorithm prone to be trapped by local minima. Last column shows segmentations computed with the proposed EC-GMMF method. The experiment shows the superior performance of the proposed method: EC-GMMF produces acceptable segmentations even for low SNR data and in a fraction of the computational time of the compared methods. The number of iterations for all the algorithms were 500 in all cases. The initial estimates for the models ($\phi_{rk} = \theta_k$, with $\theta_k \in \Re$) where uniformly distributed in the dynamic range of the noisy data; we initialize $p_{0kr} = v_{kr}$ and $p_{0kr} = 1/5$ for the EC-GMMF and the HMMF algorithms, respectively. In this case $-log v_{kr} = (g_r - \theta_k)^2$. So that step 10 in algorithm 1 is computed, for each θ model at the i^{th} iteration, with: $\theta_{ik} = \frac{\sum_r g_r p_{ikr}^2}{\sum_r p_{ikr}^2}$.

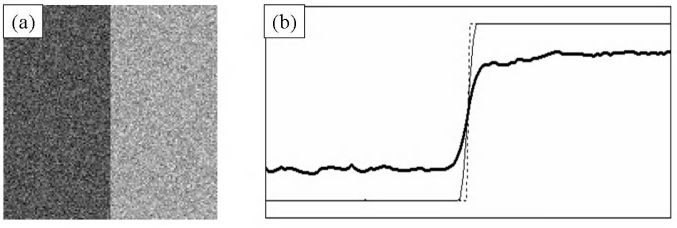


Fig. 2. Comparison of the robustness to noise of different methods. (a) Test image. (b) Computed marginals corresponding to the central row (see text).

Figure 2-(b) shows a comparison of the computed marginals for the central row in figure 2-(a). The model parameters are in this case assumed known, i.e. $\phi_{1r} = 0$ and $\phi_{2r} = 1$, for all the pixels r. The thin solid line corresponds to the p_2 marginals computed with the Gibbs Sampler algorithm (a MCMC method) after 2,000 iterations, with a computational time of 15.92 sec. The heavy line shows the marginals computed with the original GMMF formulation (with $\lambda = 10$ in $0.18 secs.$) and the dotted line, the marginals computed with the EC-GMMF method in 1 second (with $\lambda = 10$ and $\mu = 3$ in $0.33 secs.$). The proposed EC-GMMF approximates very closely the marginals computed with the MCMC method but at a fraction of the computational time. If no entropy control is applied in the EC-GMMF formulation (i.e., for $\mu = 0$), then the computed marginals are similar to the ones computed with the original GMMF method.

For the multi–class problem such entropy reduction can be observed in the maximum p-value maps (i.e $\max_k p_{kr}$). Figure 3 shows the max p–values for the case of 10 models computed with: (a) Gibbs Sampler (comp. time, 318.12 secs.), b) original GMMF (comp. time 0.75 secs.) and (c) EC-GMMF (comp. time, 2.03 secs.), respectively. The apparently low entropy of the Gibbs sampler results may be explained by an incomplete convergence, even after about 5 min.

Next experiment compares the computational efficiency of the EC-GMMF as the number of models is incremented. Parametric segmentations of the Lena's portrait were

Gibbs Sampler GMMF EC-GMMF

Fig. 3. Maximum value of the marginals: Dark pixels denotes low (high entropy) values

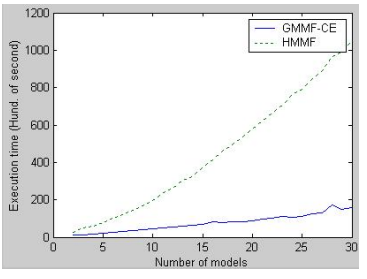


Fig. 4. Comparison of the computational times for the EC-GMMF and HMMF algorithms for different number of models

performed with the HMMF and the EC-GMMF algorithms. Figure 4 shows the corresponding computational times; the Gibbs sampler method (not shown in the plot) required 5 min for 10 models.

3.2 Optimal Estimators for Piecewise Smooth Reconstruction

The fact that one can obtain very precise approximations for the marginals, allows one to compute estimators other than the MPM, which give very good results for piecewise smooth restoration problems. These estimators may be obtained even with fixed models that sample uniformly the search space, producing highly efficient methods. The idea is to compute, instead of the posterior mode (MPM estimator), the mean or median of the estimated posterior marginal distribution at each pixel. The resulting estimators will have sharp discontinuities when these are present in the image, and produce smooth transitions between adjacent models in other places. The posterior mean is computed using: $\bar{f}_r = \sum_k \theta_k p_{kr}$.

This is illustrated in Fig. 5, where the first row shows the results for the piecewise smooth restoration of a noise-corrupted Lena image. Second row in Fig. 5 shows details of the corresponding images in first row. A similar procedure may be used for the computation of piecewise smooth optical flow from a pair of images [1][12][14][22]. In this case, the models are 2-vectors that correspond to discrete velocities that sample the space of allowed displacements: $\phi_{ij} = \theta_{ij} = [u_i, v_j]^T$ where we make a slight notation change by substituting the index model k by the more intuitive pair ij. For the example of Fig. 6 we use $u_i = v_i = d_m(2i/\Delta - 1)$ for $i = 0, 1, \ldots, \Delta)$, where

Fig. 5. First row: Explicit entropy control allows us restore images by using estimators other than the MPM see text). Second row: Details.

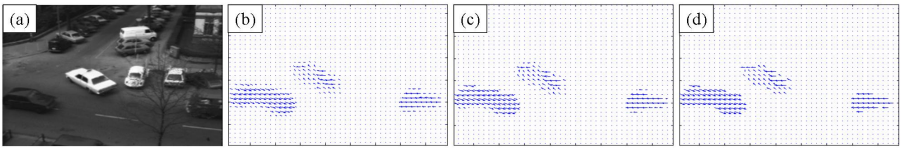


Fig. 6. Optical flow computation (a) Frame 8 of the Hamburg taxi and optical flow magnitude estimator with (b) $n = 1$, (c) $n = 2$ and (d) $n = \infty$ (mode)

d_m is the largest expected displacement and Δ is number of models in $[-d_m, d_m]$; we use $d_m = 2.5$ and $\Delta = 6$ with bi–cubic interpolation for the fractional displacement that corresponds to consider displacements of $\{\pm 2.50, \pm 1.66, \pm 0.83, 0\}$ in the x and y directions. Now, let f_1 and f_2 be two consecutive frames from a sequence. Then, assuming Gaussian noise, we have: $-\log v_{ijr} = [f_1(r) - f_r(r - \theta_{ij})]^2$. One may now compute an optimal estimator that interpolates smooth velocities between neighboring models, while preserving the motion discontinuities. Note that since in this case the models are only partially ordered, one cannot compute the median. Instead, one may compute a family of estimators that include the mode and the mean as special cases:

$$\bar{f}_r = \sum_{ij} \theta_{ij} w_{ijr} \tag{16}$$

with $w_{ijr} = p_{ijr}^n / \sum_{ij} p_{ijr}^n$ The parameter n controls the sharpness of the transitions between models: for large n the estimator corresponds to the mode, while for $n = 1$ one gets the mean.

Since the θ parameters are fixed, one finds the optimal p as the unique minimizer of U_{EC} using the Gauss-Seidel algorithm with projection that was described above. The only change that is necessary is in the definition of m_{kr} in (13), which changes to:

$$m_{ijr} \stackrel{def}{=} [f_1(r) - f_r(r - \theta_{ij})]^2 - \mu + \lambda \text{card}(\mathcal{N}_r). \quad (17)$$

Note that since (17) does not depends on p, m_{ijk} can be precomputed, and the computational cost is the same as that for the case of image restoration with fixed constant models.

Figure 6 shows the computed optical flow with the EC-GMMF method for different estimators. Panel (a) shows the frame 8 of the Hamburg taxi sequence (image f_1) the other image corresponds to the frame 9 (no shown). Panel (b) shows the mode optical flow, i.e. the vector field θ_{ij} corresponding to the largest marginal at each pixel. Panel (c) show the mean flow and panel(d) shows a robust mean computed with (16) and $n = 2$. The mean optical flow is obtained with a sub-pixel resolution and the edges are better preserved by the robust mean.

4 Summary and Conclusions

We have proposed an efficient parametric segmentation method based on Bayesian estimation with prior MRF models and the Expectation-Maximization (EM) procedure. This method estimates the model parameters and the posterior marginals in successive steps as minimizers of QPD energy functions subject to linear constraints, so that each step in the EM procedure has a unique minimum. We also showed that it is possible to implement the estimation process as a Generalized EM algorithm, in which one performs the minimization of the posterior energy with respect to the model parameters and the posterior marginals simultaneously, which decreases significantly the computational cost.

The key point for the superior performance of our method is the introduction of a quadratic term (derived from the Gini coefficient) that controls the entropy of the posterior marginals. This performance is demonstrated by numerical experiments that compare our approach with other state-of-the-art algorithms, such as minimum graph-cut and HMMF methods. The numerical experiments performed demonstrate that the proposed algorithm is more robust to noise and to the initial values for the parameters and significantly more efficient from a computational viewpoint.

It is important to remark that for the case of fixed models, the algorithm reduces to a single E step, and the posterior marginals computed with a simple and efficient Gauss-Seidel procedure correspond to the global optimum. Since these marginals are entropy-controlled, they approximate very well the true ones, and may be used to compute estimators different from the posterior mode (MPM), such as the posterior mean or median. These estimators have the property of interpolating smoothly the estimated feature between neighboring constant models, while preserving the discontinuities in the solution. Two early vision applications that take advantage of this fact are presented: piecewise smooth image reconstruction and optical flow computation.

References

1. S. Birchfield and C. Tomasi: Multiway cut for stereo and motion with slanted surfaces. ICCV (1999) 489-495.
2. Y. Boykov, O. Veksler, and R. Zabih: Fast approximate energy minimization via graph cuts. IEEE–PAMI, 23 (2001) 1222–1239.

3. Y. Boykov and M.-P. Jolly: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. ICCV, I, (2001) 105112.
4. A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Statist. Soc. B 39 (1977) 138 .
5. R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification*, 2nd. ed., John Wiley & Sons, Inc., New York, 2001
6. S. Geman and D. Geman: Stochastic relaxation, Gibbs distributions and Bayesian restoration of images. IEEE–PAMI, 6 (1984) 721–741, .
7. D. Greig, B. Porteous, and A. Seheult: Exact maximum a posteriori estimation for binary images. J. R. Statist. Soc. B, 51 (1989) 271279.
8. A.K. Jain and R.C. Dubes:Algorithm for Clustering Data. Englewood Cliffs, NJ: Prentice Hall (1998).
9. V. Kolmogorov and R. Zbih: What energy functions can be minimized via graph cuts. ECCV III (2002), 65–81.
10. S.Z. Li: Markov Random Field Modeling in Image Analysis. Springer-Verlag, Tokyo (2001).
11. J. Marroquin, S. Mitter and T. Poggio: Probabilistic solution of ill–posed problems in computational vision. J. Am. Stat. Asoc., 82 (1987) 76-89.
12. J.L. Marroquin, S. Botello, F. Calderon and B.C. Vemuri: The MPM-MAP algorithm for image segmentation. ICPR (2000).
13. J. L. Marroquin, F. Velazco, M. Rivera and M. Nakamura: Gauss-Markov Measure Field Models for Low-Level Vision. IEEE–PAMI, 23 (2001) 337–348.
14. J. L. Marroquin, E. Arce and S. Botello: Hidden Markov Measure Field Models for Image Segmentation. IEEE–PAMI, 25, (2003) 1380-1387.
15. R. Neal and R. Barry: A vew of the EM algorithm that justifies incremental, sparse, and others variants. in Learning in Graphical Models, M. Jordan (ed.) Dordrecht: Kluwer Academic Publishers, Boston MA. (1998) 355-368.
16. J. Nocedal and S. Wright: Numerical Optimization (Springer Series in Operational Resarch) Springer Verlag, New York (1999).
17. O. Picher, A. Teuner and B. Hosticka: An unsupervised texture segmentation algorithm with feature space reduction and knowledge feedback. IEEE Trans. Image Process. 7 (1998) 53–61.
18. M. Rivera and J.C. Gee:Image segmentation by flexible models based on robust regularized networks. ECCV, III (2002) 621–634.
19. M. Rivera and J.C. Gee, "Two-level MRF models for image restoration and segmentation," BMVC, 2 (2004) 809-818.
20. Tsai, J Zhang and A Willsky: Expectation-Maximization Algorithms for Image Processing Using Multiscale Methods and Mean Field Theory, with Applications to Laser Radar Range Profiling and Segmentation. Opt. Engineering, 40, 7, (2001) 1287-1301.
21. Z. Tu, S.C. Zhu and H.Y. Shum: Image Segmentation by Data Driven Markov Chain Monte Carlo. ICCV (2001) 131–138.
22. Y. Weiss and E.H. Adelson: A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. CVPR (1996) 321–326.
23. Z. Wu and R. Leaby: An optimal graph theoretical approach to data clustering: Theory and its applications to image segmentation. IEEE–PAMI, 11 (1993) 1101–1113.
24. J. Zhang: The mean field theory in EM procedures for Markov random fields. IEEE Trans. Signal Processing, 40 (1992) 2570–2583.
25. A. Tsai, J Zhang, A Wilsky: Multiscale Methods and Mean Field Theory in EM Procedures for Image Processing. Eight IEEE Digital Signal Processing Workshop (1998).

Global Minimization of the Active Contour Model with TV-Inpainting and Two-Phase Denoising

Shingyu Leung and Stanley Osher

Department of Mathematics, UCLA, Los Angeles, CA 90095, USA
{syleung, sjo}@math.ucla.edu
http://www.math.ucla.edu/~syleung
http://www.math.ucla.edu/~sjo

Abstract. The active contour model [8,9,2] is one of the most well-known variational methods in image segmentation. In a recent paper by Bresson et al. [1], a link between the active contour model and the variational denoising model of Rudin-Osher-Fatemi (ROF) [10] was demonstrated. This relation provides a method to determine the **global** minimizer of the active contour model. In this paper, we propose a variation of this method to determine the global minimizer of the active contour model in the case when there are missing regions in the observed image. The idea is to turn off the L^1-fidelity term in some subdomains, in particular the regions for image inpainting. Minimizing this energy provides a unified way to perform image denoising, segmentation and inpainting.

1 Introduction

Image segmentation, image restoration and image inpainting are a few basic yet important areas in image processing and computer vision. Traditionally, these closely related fields were developed independently. However, the use of the level set method and variational methods in recent years started to bring all these fields together. One example is the TV-inpainting model [7]. We can perform inpainting in a desired domain while applying the ROF model [10] to remove noise from the rest of the domain using only one energy functional.

There are two interesting recent developments about the connection between different fields in image processing. We will discuss later in this paper how they link different fields in an interesting way. The first development concerns the impulse-noise removal method and the variational method for image regularization. In two recent papers by Chan, Nikolova et al. [3,4], a two-phase method was proposed to remove impulse-type noise. For a true image $u^*(x)$ and an observed image $f(x)$ defined in a domain Ω, impulse-type noise is defined by

$$f(x) = \begin{cases} r(x) & \text{with probability } r_0 \\ u^*(x) & \text{with probability } (1 - r_0). \end{cases} \tag{1}$$

As an example, for the so-called salt-and-pepper noise, $r(x)$ is simply the maximum or the minimum of the image intensity. The main idea in those papers is to separate the denoising process into a noise detection phase and a noise removal phase. In the first stage,

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 149–160, 2005.

a median-type filter is applied to the observed image to detect the possible locations of the impulse noise. Then in the second phase, instead of replacing the intensity at all locations by the median intensity around a certain neighborhood, an L^1-regularization method is applied only to those locations reported in the first phase while keeping the other pixels unchanged. The resulting method was shown to be able to remove salt-and-pepper noise efficiently even at a very high noise level (for example $r_0 = 0.75$). The main reason for its success is that this method both retains those pixels that are unlikely to be polluted and maintains sharp edges in the whole image.

Another interesting development is a new model that uses a variational method for image segmentation [5,1]. The idea of the model is to minimize an energy functional consisting of a weighted TV-norm with an L^1-fidelity term. For segmentation of a binary image, the papers showed an equivalence between the new energy functional and that of the active contour model. This relation can be used to overcome the problem of the active contour model in which the energy function is not convex. Very often the snake will be trapped into a **local** minimum thus giving unsatisfactory segmentation results. The link between these two energies, as demonstrated in the above papers, provides a convenient way to determine the **global** minimizer of the active contour energy.

In this paper, we will combine these two recent advances in image processing. This provides an efficient way to bring image denoising (for both impulse-type and Gaussian-type noise), image segmentation, and image inpainting together. For low dimensional images, one could of course solve each of these tasks individually. It is computationally inexpensive to do so. As a result, it might not be obvious to see any advantage of having such a combined strategy for one or two dimensional images. However, for higher dimensional images, dealing with these problems separately might not be practical. The main difficulty concerns the computational time. With over 10^6 voxels, it could take days to analysis only one 3D scan. Therefore, by solving only one variational problem which performs all these tasks simultaneously, we could significantly speed up the computational time.

The rest of the paper is organized as follows. In Section 2, we will briefly review the denoising model by Chan, Nikolova et al. [3,4] and also restate the link between the active contour model and the ROF model as in [1]. A new model will be given in Section 3. Section 4 contains some details about the numerical implementation. Some numerical results are given in Section 5.

2 Two Recent Developments

2.1 A Two-Phase Method to Remove Impulse-Type Noise

Unlike the usual way to denoise impulse noise by applying the median-type filter to the image and replacing the image intensity everywhere, the idea in [3,4] is to separate the denoising processing into a noise-detection phase and a noise-removal phase. Mathematically, the first phase can be formulated as determining a noise candidate set $\mathcal{N} = \{x \in \Omega : f(x) \neq f_{\text{MF}}(x)\}$ in which Ω is the image domain, $f(x)$ is the observed image intensity at the pixel x, and $f_{\text{MF}}(x)$ is the intensity at x after applying a median-type filter, such as the classical median filter or the adaptive median filter. In the second phase, the following functional is minimized

$$F|_{\mathcal{N}}(u) = \int_{\mathcal{N}} \left\{ |u(x) - f(x)| + \frac{\beta}{2}[S_1(u) + S_2(u)] \right\} \tag{2}$$

where

$$S_1 = \int_{\mathcal{V}(x)\cap(\Omega\setminus\mathcal{N})} 2\,\phi[u(x) - f(y)]dy \text{ and } S_2 = \int_{\mathcal{V}(x)\cap\mathcal{N}} \phi[u(x) - u(y)]dy\,, \tag{3}$$

$\mathcal{V}(x)$ is the neighborhood centered at x and ϕ is an edge-preserving potential function. As seen in [3,4], one possible choice for ϕ is $\phi(t) = \sqrt{t^2 + \epsilon^2}$ with a small constant ϵ. The first term in the curve-bracket is an L^1-fidelity term. The terms in the square-bracket can be interpreted as an approximation of the total variation of u.

In the simple case when the noise can be separated accurately in the first step, the fidelity term is not important. This whole algorithm can then be simplified and the resulting variational method is the same as an image inpainting algorithm. For example, if ROF [10] or L^2 squared fidelity is used instead, we arrive at the TV-inpainting of [7]. That is, given an observed image f, one minimizes the following energy

$$E_1(u) = \int_\Omega |\nabla u| + \frac{1}{2}\int_\Omega \lambda(x)|u - f|^2 \tag{4}$$

where $\lambda(x) = 0$ if $f(x) = f_{\text{MF}}(x)$ and $\lambda(x) = \lambda_\infty \simeq \infty$ otherwise.

The idea of using a piecewise constant $\lambda(x)$ in TV-inpainting is not new [7]. However, it is interesting to see here the relationship between impulse-type noise removal and image inpainting by using a $\lambda(x)$ determined by a median-type filter.

2.2 Global Minimizer of the Active Contour Model

In the classical active contour model, the initial guess of the segmented image plays a very important role. We show in Figure 1 some minimizers of the active contour model. As we can see, different initial conditions in the evolution will give different segmented region. More importantly, none of these results corresponds to the *true* segmented results, i.e. curves which separate all regions with different intensities in the whole image. One reason for these unsatisfactory results is that the minimization problem of the active contour is not convex, and therefore it is very likely that the energy minimization could be trapped into a **local** minimium.

Recently, a few algorithms were proposed [5,1] to determine the **global** minimizers of some image segmentation models. In particular, an algorithm to determine the global minimizer of the active contour model based on the ROF model was given in [1]. The idea is to modify the ROF energy

$$E_{\text{ROF}}(u, \lambda) = \int_\Omega |\nabla u| + \frac{\lambda}{2}\int_\Omega |u - f|^2 \tag{5}$$

by first replacing the TV-norm by a weighted TV-norm and then, more importantly, changing the measure in the fidelity term from the square of the L^2-norm to the L^1-norm. This gives

$$E_2(u, \lambda) = \int_\Omega \tilde{g}(f)|\nabla u| + \lambda\int_\Omega |u - f| \tag{6}$$

in which $\tilde{g}(f) = 1/(1 + \beta|\nabla f|^2)$.

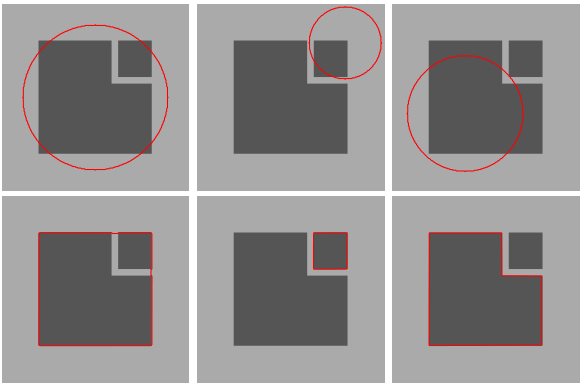


Fig. 1. Segmentation results using the active contour model. We show different initial configurations of the snake on the first row. The corresponding segmented results using these initial conditions are shown on the second row.

As pointed out in [1], if u is the characteristic function of a set Ω_C with boundary given by the curve C (i.e. $u = \mathbf{1}_{\Omega_C}$), the minimizer of the above energy E_2 is the same as the minimizer of the active contour energy

$$E_{\text{AC}}(C) = \int_C \tilde{g}(f) ds \tag{7}$$

with f approximated (in the sense of L^1) by a binary function of a region Ω_C.

Numerically, the minimization problem (6) is convex. This means that the method of gradient descent will converge to a unique minimizer, i.e. the global minimum of the energy function, independent of the initial condition. This equivalence is significant because by minimizing (6), one can determine the global minimium of the active contour model (7) avoiding both the danger of being trapped into any local minimum and the uncertainty in picking an initial configuration of the snake.

3 The New Energy

3.1 The Energy

Here, we propose a new model to combine those developments mentioned above. Given an observed image f, we minimize the energy

$$E(u) = \int_\Omega g(f)|\nabla u| + \int_\Omega \lambda(x)|u - f| \,. \tag{8}$$

This energy is similar to (6), except that $\lambda(x)$ is now changed into a function of the space variable and the weight in the weighted TV-norm is also modified. The function $\lambda(x)$ has the following properties.

$$\lambda(x) = \begin{cases} 0 & \text{TV-inpainting} \\ \lambda_0 & \text{Denoising} \\ \lambda_\infty \simeq \infty & u \text{ remains unchanged.} \end{cases} \tag{9}$$

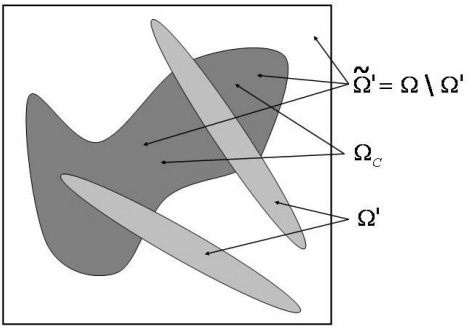


Fig. 2. Problem setting. Definition of the set Ω' (domain for inpainting), $\tilde{\Omega}'$ (compliment of Ω') and Ω_C (domain bounded by the curve C).

In the subdomain for image inpainting, $f(x)$ (and also $\tilde{g}(f)$) might not be known. We therefore simply set $g(f) = 1$, or $\beta = 0$. For the rest of the domain, we keep $g(f) = \tilde{g}(f)$.

Here we give some suggestions in picking such a function $\lambda(x)$ and also provide a variation in using the above minimization algorithm. For salt-and-pepper noise, the following $\lambda(x)$ works efficiently. We define $d(x)$ to be the difference in the intensities between the original image $f(x)$ and the modified image after applying the median-type filter $f_{\text{MF}}(x)$, i.e. $d(x) = |f(x) - f_{\text{MF}}(x)|$. Then one can set

$$\lambda_1(x) = \begin{cases} \lambda_\infty \text{ if } d(x) = 0 \text{ and } x \notin \Omega' \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{10}$$

where $\Omega' \subset \Omega$ is a given subdomain for doing image inpainting and this subdomain can also be characterized by an user predefined mask function. This means that if x is in the inpainting domain Ω' or if the noise-detector detects that the image at x is polluted (therefore $f(x)$ will be different from the intensity after applying the median-type filter $f_{\text{MF}}(x)$), then the intensity at x will be modified by a TV-type regularization. Otherwise, the intensity at that location will remain unchanged.

If the impulse noise is random-valued instead, one can use a similar $\lambda(x)$

$$\lambda_2(x) = \begin{cases} \lambda_0 \text{ if } d(x) = 0 \text{ and } x \notin \Omega' \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{11}$$

with $\lambda_0 << \lambda_\infty$.

For Gaussian-type noise, one can simply use

$$\lambda_3(x) = \begin{cases} \lambda_0 & \text{if } x \notin \Omega' \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In the case when the type of noise is not known *a priori*, one can try to minimize (8) iteratively. More specifically, given the observed image $u_0 = f$, for $m = 1, \cdots, m_{\max}$, one iteratively minimizes

$$E(u_m) = \int_\Omega g(u_{m-1})|\nabla u_m| + \int_\Omega \lambda_4(x)|u_m - u_{m-1}| \tag{13}$$

with

$$\lambda_4(x) = \begin{cases} \lambda_0 \text{ if } d(x) \le d^* \text{ and } x \notin \Omega' \\ 0 \qquad\quad \text{otherwise} \end{cases} \tag{14}$$

where d^* is a threshold in the intensity difference function $d(x) \equiv |u_{m-1} - (u_{m-1})_{\text{MF}}|$.

3.2 The Link Between Active Contour for Segmentation, Denoising and TV-Inpainting

We explain here the relations between the minimization of the energy functional (8), the active contour model and the TV-inpainting model. Assuming $\Omega' = \{x \in \Omega : \lambda(x) = 0\}$ is the subdomain for inpainting (note that $\mathcal{N} \subset \Omega'$) and $\tilde{\Omega}' = \Omega \setminus \Omega'$, we have

$$E(u) = \int_{\tilde{\Omega}'} g(f)|\nabla u| + \int_{\tilde{\Omega}'} \lambda_0|u - f| + \int_{\Omega'} |\nabla u| = E^1(u) + E^2(u) \tag{15}$$

where

$$E^1(v) = \int_{\tilde{\Omega}'} g(f)|\nabla v| + \int_{\tilde{\Omega}'} \lambda_0|v - f| \text{ and } E^2(w) = \int_{\Omega'} |\nabla w| \tag{16}$$

with $v : \tilde{\Omega}' \to [u_{\min}, u_{\max}]$ and $w : \Omega' \to [u_{\min}, u_{\max}]$. So minimizing $E(u)$ is the same as $\min_v E^1(v) + \min_w E^2(w)$, and the minimizer of $E(u)$ is given by $u = \mathbf{1}_{\tilde{\Omega}'}(x) \cdot v + \mathbf{1}_{\Omega'}(x) \cdot w$ in which $\mathbf{1}_{\tilde{\Omega}'}$ is the characteristic function of the set $\tilde{\Omega}'$.

First we consider the energy $E^1(v)$. If Ω_C is a set in $\tilde{\Omega}'$ whose boundary is denoted by C and if the minimizer of $E^1(v)$ is given by $v = \mathbf{1}_{\Omega_C}$, then we have

$$E^1(v) = \int_{\tilde{\Omega}'} g(f)|\nabla \mathbf{1}_{\Omega_C}| + \int_{\tilde{\Omega}'} \lambda_0|\mathbf{1}_{\Omega_C} - f| = \int_C g(f)ds + \int_{\tilde{\Omega}'} \lambda_0|\mathbf{1}_{\Omega_C} - f|\,. \tag{17}$$

Therefore, minimizing $E^1(v)$ in the subdomain $\tilde{\Omega}'$ in the case of a binary observed image is equivalent to minimizing the active contour energy in $\tilde{\Omega}'$ while approximating f (in the L^1 sense) in $\tilde{\Omega}'$ by a binary function of Ω_C.

For the energy $E^2(w)$ defined in the complement, Ω', we have $E^2(w) = \int_{\Omega'} |\nabla w|$ and the boundary condition $w|_{\partial\Omega'} = v|_{\partial\Omega'}$ where v is the minimizer of $E^1(v)$. In the case when v is binary on $\partial\Omega'$, we have $w = \mathbf{1}_{\Omega_{C'}}$ again. This gives

$$E^2(w) = \int_{\Omega'} |\nabla \mathbf{1}_{\Omega_{C'}}| = \int_{C'} ds\,. \tag{18}$$

This implies that when the value of v on the boundary $\partial\Omega'$ is binary, minimizing $E^2(w)$ in Ω' is equivalent to $\min_{C'} \int_{C'} ds$ while the end points of C' are fixed on $\partial\Omega'$. Further analysis on the behavior of TV-inpainting can be found in [6].

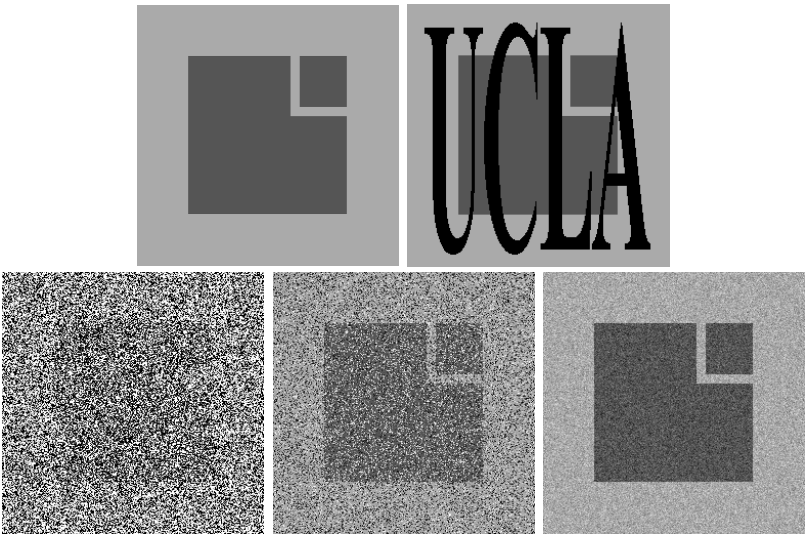


Fig. 3. (Example 1) (First row) The original true image and the corresponding image with an user defined mask (in black). (Second row) The original image with 75% salt-and-pepper noise, 50% random-valued impulse noise and additive Gaussian noise ($\sigma = 20$) respectively.

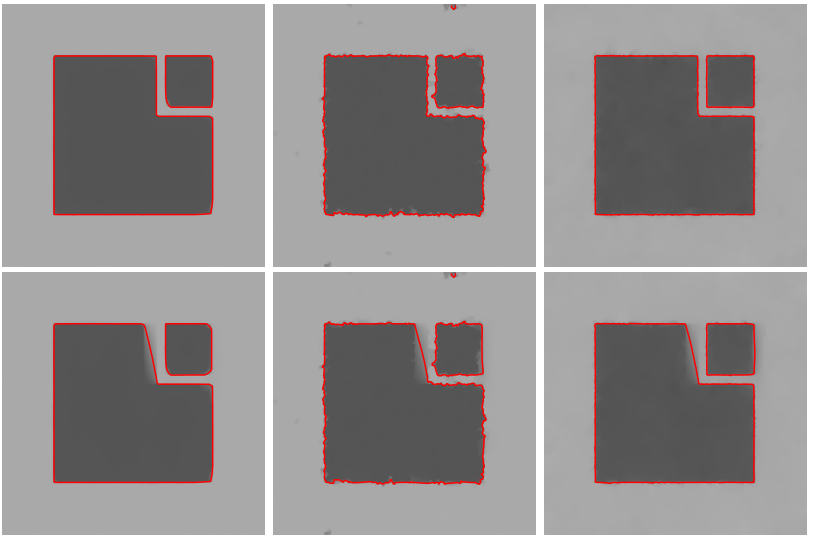


Fig. 4. (Example 1) The minimizer for the energy (8) **without** (first row) and **with** (second row) an extra mask. The left, middle and right columns show the denoising results with salt-and-pepper, random-valued impulse and addition Gaussian noise.

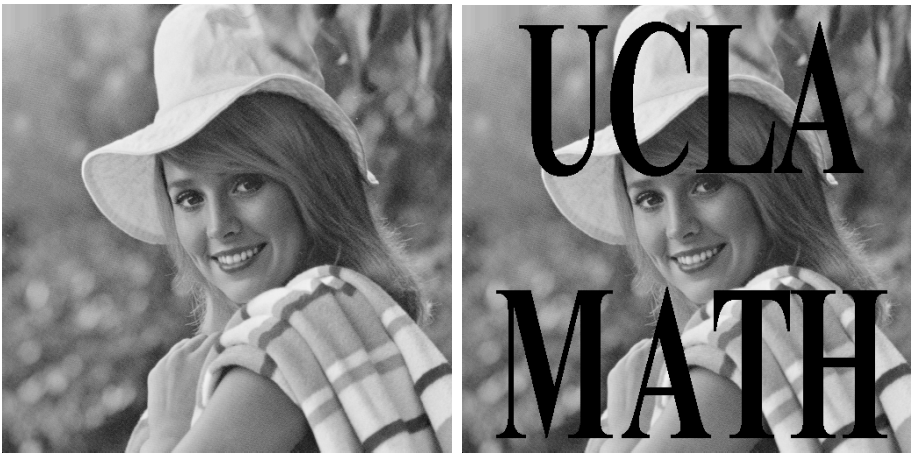


Fig. 5. (Example 2) The true image and the corresponding image with an user defined mask (in black)

Fig. 6. (Example 2) The true image with 75% salt-and-pepper noise, 50% random-valued impulse noise and additive Gaussian noise ($\sigma = 20$) respectively

4 Numerical Method

To minimize the above energy, we use the method of gradient descent. The Euler-Lagrange equation of the energy functional (8) is given by

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{g(x)}{|\nabla u|} \nabla u \right) - \lambda(x) \frac{u - f}{|u - f|} \,. \tag{19}$$

This equation can be solved numerically using the standard fully explicit scheme. In the current paper, however, we solve (19) using the Alternative Direction Explicit (ADE) technique.

5 Examples

In the following examples, we use $u^0(x, y) = 0$ as the initial condition for the Euler-Lagrange equation. Unlike the classical active contour/snake model, different initial

guesses used will give the **same global** minimizer of the segmentation model in the case of binary images.

5.1 Example 1

The true image used in this example has 256×256 pixels and is shown on the top left in Figure 3. The corresponding image with an user predefined mask is shown on the right on the same row. The black region is the domain Ω' where we want to perform TV-inpainting. The second row of Figure 3 shows the noisy versions of the true image.

The first row of Figure 4 shows the denoised image together with its segmentation with 75% salt-and-pepper noise, random-valued impulse noise and additive Gaussian noise. As we see from the figures, all noise is completely removed from the image. The red curves on the graphs are the boundaries of the segmented regions. Unlike the minimization of the active contour model, we can now easily reach the global minimum of the energy for these binary images regardless of the initial condition of the Euler-Lagrange equation. In the case of denoising together with image inpainting, the segmented results are shown on the second row of Figure 4. Again, all noise is completely removed from the image and we are able to fill in the missing part of the image using only one energy function.

5.2 Example 2

The true image of Elaine used in this example has 512×512 pixels and is shown on the left of Figure 5. On the right, we give the corresponding image with a predefined mask. 75% salt-and-pepper noise, 50% random-valued impulse noise and additive Gaussian noise with standard deviation $\sigma = 20$ are added to the original image and these observed images are shown in Figure 6. The minimizers of the energy functional (8) for these noisy images are shown in Figure 7. Since the original image is not binary, different level contours of the final figure will give different minimizers [1]. Using the case where salt-and-pepper noise and an external mask are added, we plot in Figure 8 the segmented results using the levels $u = 128$ and $u = 170$.

5.3 Example 3

Figures 9 shows the denoising results of a 3D brain MRI image. The number of voxels is $128 \times 256 \times 256$. The computational time is approximately 354 mins.

6 Conclusion

In this paper, we proposed combining the three most fundamental tasks in image processing - denoising, inpainting and segmentation. The idea is based on [5,1] but with the Lagrange multiplier, $\lambda(x)$, carefully chosen. At those locations where inpainting is required, we can set $\lambda(x)$ to be zero, which gives TV-inpainting. Indeed, TV-inpainting may not be the perfect model. For example, as mentioned in [7], TV-inpainting may not satisfy the so-called Connectivity Principle and may give unpleasant results to human vision. To improve this, one could modify the TV-regularization in the current

Fig. 7. (Example 2) The minimizer for the energy (8) **without** (first column) and **with** (second column) an extra mask. The top, middle and bottom rows show the denoising results with salt-and-pepper, random-valued impulse and addition Gaussian noise.

Fig. 8. (Example 2) Different segmented results with level (left) $u = 128$ and (right) $u = 170$ for the case with salt-and-pepper noise and an extra mask, i.e. from the top right picture in Figure 7

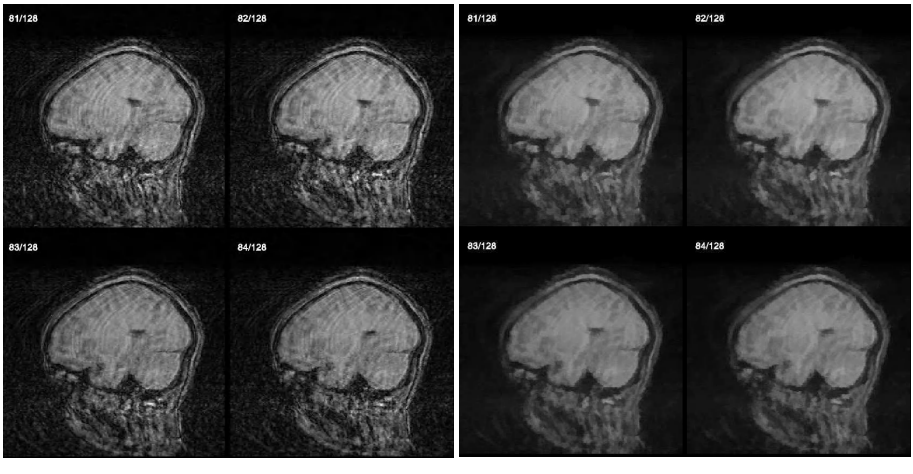


Fig. 9. (Example 3) (Left) A few slices of a noisy brain MRI image and (right) their corresponding denoised version

framework by a higher order interpolation model, for example, based on the Euler's Elastica [7]. As another drawback of the model, we indeed obtain multiple minimizers of the active contour model for non-binary images, as seen in Figure 8. This limitation is inherited from the original theory as described in [1].

Acknowledgment

This work was supported in part by NSF contract DMS-0312222 and NIH contract U54 RR021813.

References

1. X. Bresson, S. Esedoḡlu, P. Vandergheynst, J.P. Thiran, and S. Osher. Global minimizers of the active contour/snake model. *UCLA CAM Report (05-04)*, 2005.
2. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
3. R. Chan, C.-W. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *In preparation*, 2004.
4. R. Chan, C. Hu, and M. Nikolova. An iterative procedure for removing random-valued impuse noise. *In preparation*, 2004.
5. T.F. Chan, S. Esedoḡlu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *UCLA CAM Report (04-54)*, 2004.
6. T.F. Chan and S.H. Kang. Error analysis for image inpainting. *UCLA CAM Report (04-72)*, 2004.
7. T.F. Chan and J. Shen. Variational image inpainting. *Comm. Pure Appl. Math.*, 58(5):579–619, 2005.
8. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1996.
9. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A.J. Yezzi. Conformal curvature flows: From phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996.
10. L. Rudin, S.J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

Combined Geometric-Texture Image Classification

Jean-François Aujol[1] and Tony Chan[2]

[1] CMLA (CNRS UMR 8536) - ENS Cachan, France
aujol@cmla.ens-cachan.fr
http://www.cmla.ens-cachan.fr/∼aujol
[2] Dean, Division of Physical Science, College of Letters and Science, UCLA, USA
TonyC@college.ucla.edu
http://www.math.ucla.edu/∼chan/

Abstract. In this paper, we propose a framework to carry out supervised classification of images containing both textured and non textured areas. Our approach is based on active contours. Using a decomposition algorithm inspired by the recent work of Y. Meyer, we can get two channels from the original image to classify: one containing the geometrical information, and the other the texture. Using the logic framework by Chan and Sandberg, we can then combine the information from both channels in a user definable way. Thus, we design a classification algorithm in which the different classes are characterized both from geometrical and textured features. Moreover, the user can choose different ways to combine information.

Keywords: Classification, texture, geometrical image, decomposition, logic model, level-set, active contour, PDE, wavelets.

1 Introduction

The problem of image classification consists in assigning a label to each pixel of an image. This label indicates to which class belongs a pixel. The classification problem is closely related to the segmentation one, in the sense that the aim is to get a homogeneous partition of the image. In the classification problem, each region represents a class. Many classification models have been developed: region growing algorithms [30], stochastic algorithms [9,18,29], and most recently active contours [2,23,20,4,26], including statistical ones [16,14].

The approach which we use here is inspired from [23,34,4,24], and is based on active contours [2,11,28,35,13,31]. The partition we seek is a minimizer of a functional. We compute this minimizer by solving the associated PDE's system. These PDEs guide the interfaces (zero level sets) towards the boundary of the optimal partition thanks to internal (regularity of the interface) and external (data term and partition) forces. But, as far as we know, none of the existing classification models using a variational approach can deal with images containing both textured and non-textured areas (except for the work of [29]).

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 161–172, 2005.

For instance, [23,34,13] are designed to work with non-textures images whereas [20,25,4,17,21,10] are specifically devoted to textured images classification.

The idea of our method is as followed. Given an original image, we first split it into a geometrical component and a textured component thanks to the algorithm of [3]. We thus get two channels from our original image: a geometrical one, and a textured one. We construct a data term associated to the geometrical (resp. textured) component by using the method of [23] (resp. [4]). We combine them in a user definable way thanks to the logical framework introduced in [24], and we are then in position to carry out the classification of our original image containing both textured and non-textured areas. Since we use the logical framework of [24], a class can be characterized both by textured and non-textured features.

The novelty of the paper lies in the combined use of all these recent methods. The fact that a class can be characterized simultaneously by geometric and textured features has already been proposed in [29]. The specificity of our work is that thanks to the logical framework, we can combine these two information in a user definable way. Indeed, depending on the application, there can be many meaningful classifications for the same image. One of the main advantage of our framework is that the user can specify which classification result he is interested in. The use of a decomposition algorithm to split an image into a geometrical component and a textured component, in order to apply adapted algorithms to each part of the image, has already been introduced successfully in image inpainting [8] and in image nonlinear interpolation [7]. Notice that the classification part of our algorithm does not care how we compute the image decomposition, and therefore other methods could be used as well to get the two channels.

The plan of the paper is as followed. In Section 2, we first introduce the decomposition models of [3,5]. In Section 3, we present a general classification framework as used in [23,4]. We emphasize on the choice of the data term in Section 4. In Section 5, we recall the logic model of [24] and show how we can use it in our classification algorithm. We illustrate our approach with some numerical examples.

2 Decomposition Model

In [19], Meyer discussed the classical Rudin-Osher-Fatemi model [22]. He introduced a new model to split a given image f into a sum $u + v$ of a bounded variation component and a component containing the oscillating part of the image. The u component can be viewed as a sketch of the original image f. This model has first been successfully implemented by Vese and Osher [32]. In this paper, we will use the algorithm of [3]. See also [5,27] and references herein for other interesting approaches.

Let Ω be a bounded open set of $\mathbb{R}^2$ with Lipschitz boundary. The space used to model the geometrical component u of an image f is the space BV of functions with bounded variation. A function f belongs to $BV(\Omega)$ if it is in $L^1(\Omega)$, and if its total variation is finite. We will denote by $J(f)$ the total variation of f.

Formally, we have: $J(u) = \int_\Omega |\nabla u|$ We refer to [12,19,2] for a deeper insight in BV.

In [19], Meyer has proposed a new decomposition model:

$$\inf_{(u,v)\in BV\times G | f=u+v} (J(u) + \alpha\|v\|_G) \tag{1}$$

The Banach space G contains signals with large oscillations, and thus in particular textures and noise. A function belonging to G may have large oscillations and nevertheless have a small norm. We refer the reader to [19,3,5] for the definition as well as the properties of G. In [3], the authors have introduced the following functional:

$$\inf_{(u,v)\in BV\times \mu B_G} \left(J(u) + \frac{1}{2\lambda}\|f-u-v\|^2_{L^2} \right) \tag{2}$$

where $\mu B_G = \{v \in G(\Omega)/\|v\|_G \leq \mu\}$. The parameter λ controls the L^2-norm of the residual $f-u-v$. The smaller it is, the smaller the L^2-norm of the residual gets. And μ controls the G-norm of the oscillating part v. It is shown in [3] that solving (2) is a way to solve (1). The minimum of (2) is computed by minimizing alternatively in each variable u and v. We denote by P_K the orthogonal projection on a set K. When $K = \mu B_G$ for some $\mu > 0$, this projection is computed thanks to Chambolle's algorithm [12]. The algorithm of [12] is based on a fixed point method.

Algorithm:
1) Initialization: $u_0 = v_0 = 0$.
2) Iterations: $v_{n+1} = P_{\mu B_G}(f - u_n)$ (this amounts to minimizing (2) with respect to v when u is fixed), and $u_{n+1} = f - v_{n+1} - P_{\lambda B_G}(f - v_{n+1})$ (this amounts to minimizing (2) with respect to u when v is fixed).

With the decomposition algorithms of [3], we are thus able to isolate the geometrical component of an image, as well as its textured one. In the next section, we present a general classification framework which can be used on these two components [23,4].

3 Classification Framework

Our approach here is inspired from [23,4]. We will use the same classification framework.

Partition, Level Set Approach: The image is considered as a function $u_0 : \Omega \mapsto \mathbb{R}$ (where Ω is an open subset of $\mathbb{R}^2$).

We denote by $Cl_k = \{x \in \Omega \,/\, \text{x belongs to the class k}\}$. In order to get a functional formulation rather than a set formulation, we assume that for all $k = 1 \ldots K$, Cl_k is an open set Ω_k given by a Lipschitz function $\Phi_k : \Omega \to \mathbb{R}$ such that:

$$\begin{cases} \Phi_k(x) > 0 \text{ if } x \in \Omega_k \\ \Phi_k(x) = 0 \text{ if } x \in \partial\Omega_k \\ \Phi_k(x) < 0 \text{ otherwise} \end{cases}$$

(typically, Φ_k is the signed distance function to $\partial\Omega_k$). Ω_k is thus completely determined by Φ_k (i.e. $x \in \Omega_k \Leftrightarrow H(\Phi_k(x)) = 1$, where H is the Heaviside function). The collection of open sets $\{\Omega_k\}$ forms a partition of Ω if and only if $\Omega = \bigcup_k \Omega_k \bigcup_k \partial\Omega_k$, and if $k \neq l$ $\Omega_k \bigcap \Omega_l = \emptyset$. We also use the Dirac distribution δ. In order that all the expressions we write have a mathematical meaning, we use classical regular approximations δ_α and H_α of these distributions (see [23,4]). When $\alpha \to 0$, we have $\delta_\alpha \to \delta$ and $H_\alpha \to H$.

Functional: Our functional has three terms:

1)

$$F^A(\Phi_1, \ldots, \Phi_K) = \lambda \int_\Omega \left(\sum_{k=1}^{K} H_\alpha(\Phi_i) - 1 \right)^2$$

Minimizing this energy term ensures that the result is indeed a partition of the image. It penalizes the pixels which are unclassified, as well as the ones classified in at least two regions simultaneously. This type of energy term has first been introduced in [34]. Another way to impose the partition constraint has been recently proposed in [31].

2)

$$F^B(\Phi_1, \ldots, \Phi_K) = \sum_{k=1}^{K} \gamma_k |\partial\Omega_k| \tag{3}$$

This term penalizes the contours length, which prevents from having too irregular contours and a lot of small regions. It is possible to show that (the proof is given in [23]): $\lim_{\alpha \to 0} \int_\Omega \delta_\alpha(\Phi_k)|\nabla\Phi_k| dx = |\partial\Omega_k|$. Then, in practice, we seek to minimize:

$$F^B(\Phi_1, \ldots, \Phi_K) = \sum_{k=1}^{K} \gamma_k \int_\Omega \delta_\alpha(\Phi_k) |\nabla\Phi_k| \tag{4}$$

3)

$$F^C(\Phi_1, \ldots, \Phi_K) = \sum_{k=1}^{K} e_k \int_\Omega H_\alpha(\Phi_k) B_k \, dx \tag{5}$$

This last term stands for the data term, and we will explain how to construct it in the following sections.

Complete Functional: The functional we want to minimize is the sum of the three previous terms:

$$F(\Phi_1, \ldots, \Phi_K) = F^A(\Phi_1, \ldots, \Phi_K) + F^B(\Phi_1, \ldots, \Phi_K) + F^C(\Phi_1, \ldots, \Phi_K) \tag{6}$$

See [1] for a theoretical study of this functional in the two phases case.

Dynamical Scheme: To minimize (6), we compute the associate Euler-Lagrange equations, and we embed them in the following dynamical scheme ($k = 1 \dots K$):

$$\frac{\partial \Phi_k}{\partial t} = -\delta_\alpha(\Phi_k) \left[\lambda \left(\sum_{q=1}^{K} H_\alpha(\Phi_q) - 1 \right) - \gamma_k \text{div} \left(\frac{\nabla \Phi_k}{|\nabla \Phi_k|} \right) + e_k \left(\sum_{i=1}^{I} B_k^i(x) \right) \right] \tag{7}$$

where the initial condition $\Phi_k(0, x)$ is the Euclidean signed distance function to the zero level set Φ_k.

4 Specific Data Terms

In this section, we first present a data term adapted for the classification of non textured images [23], and we then introduce another one devoted to textured image classification [4].

Classification of non Textured Images: In [23], F^C is designed to carry out the classification of non textured images. In that case, the gray level value of each class k is assumed to follow a Gaussian distribution of mean μ_k and standard deviation σ_k. One can then derive the data term F^C by using the maximum likelihood method. We have $B_k(x) = \frac{(u_0(x)-\mu_k)^2}{\sigma_k^2}$, and

$$F^C(\Phi_1, \dots, \Phi_K) = \sum_{k=1}^{K} e_k \int_\Omega H_\alpha(\Phi_k) \frac{(u_0 - \mu_k)^2}{\sigma_k^2} \, dx \tag{8}$$

Classification of Textured Images: Before presenting the model of [4], we first emphasize that other data terms for texture classification could be used. A very popular choice is to use Gabor filters as in [20,25]. Another interesting choice is to use a structure tensor as in [21,17,10]. The advantage of this last approach is that it needs much fewer parameters than Gabor filters or wavelets approach. We have decided to use the model introduced in [4] which was specifically designed to be used with the classification framework that we have introduced in Section 3. In this work, F^C is designed to carry out the classification of textured images. In that case, a class k is a texture and is characterized through an undecimated wavelet packet transform. To each pixel s, one can associate a vector $U(s) = (U_1(s), \dots, U_I(s))$, where $U_i(s)$ is the square of the wavelet coefficient in the sub-band i at pixel s. Assuming that U_i has a density probability of the kind (this fact has been checked empirically in [4])

$$p_{U_i}(y) = \frac{A_i}{2\sqrt{y}} \exp\left(- \left(\frac{\sqrt{y}}{\alpha_i} \right)^{\beta_i} \right) \mathbf{1}_{y \geq 0} \tag{9}$$

one can derive the data term thanks to the maximum likelihood method:

$$B_k(x) = \sum_{i=1}^{I} \left(-\log A_i^k + \frac{1}{2} \log u_i(x) + \left(\frac{\sqrt{u_i(x)}}{\alpha_i^k} \right)^{\beta_i^k} \right) \tag{10}$$

5 Combining Textured and non Textured Features for Classification

In the preceding section, we have presented two different data terms. We now aim at combining them to carry out image classification out of both textured and non textured features. A natural way to achieve this goal is to use the logic framework introduced in [24].

5.1 Logical Model

In [24], the authors propose a framework for object detection using logic operations as a structure for defining multi-channel segmentation. The model combines object information from the different channels into *any* logic combination. In this paper, we consider the logical OR and the logical AND models, although this method extend to the other logic models as well. A comprehensive description of various logic models can be found in [24]. The basic idea is that when we have available several channels corresponding to the same original image, each one contain some interesting information. The difficulty is then to know how to combine all the information coming from the different channels. The simplest idea might be to form an energy by adding the different pieces of information coming from each channels: this is the underlying idea in [24,4,20,29,15,33] for instance. But restricting the way to combine the information to this single possibility is obviously too limited. That is why the authors of [24] have developed a logic framework in which *any* logical operation between the different channels can be performed. This new way of considering multi-channels data has considerably increased the possible applications of active contours methods in image segmentation and classification. An application of the logical framework to the "active contours without edges" model [13] has been developed in [24]. One of the main advantage of this approach is its flexibility (as illustrated on Figures 3, 4, and 5). For the same image, there can be different meaningful classifications. We can adapt the logical framework to get any of these meaningful classifications.

We propose here a way to adapt it to our classification framework. Assume that $\tilde{B_1}$ and $\tilde{B_2}$ are logical variables (whose values are in $[0,1]$). The logical OR and logical AND models are built from the two logical elements:

$$\begin{aligned} &\text{OR element} && \sqrt{\tilde{B_1}(x)\tilde{B_2}(x)} \\ &\text{AND element} && 1-\sqrt{(1-\tilde{B_1}(x))(1-\tilde{B_2}(x))} \end{aligned} \tag{11}$$

These logical elements are continuous analogues of the OR and AND operations of binary logic. The arguments $\tilde{B_1}$ and $\tilde{B_2}$ take on any real value between 0 and 1, where a value of 0 corresponds to no error and a value of 1 corresponds to the maximum error. By construction, the logical elements also take on values between 0 and 1. To apply this logical model to our classification framework, we need to calibrate the different data terms. Indeed, the functions B_k which we use for non textured images is positive, but it can be larger than 1; and there is no

reason why the B_k we use for textured images should range in $[0,1]$. We know that the larger $B_k(x)$ is, the less likely x belongs to the class k. Therefore, we just need to linearly modify B_k so that it ranges in $[0,1]$ (and so that $B_k(x)=0$ means that x is very likely to belong to the class k, whereas $B_k(x)=1$ means that x is very unlikely to belong to the class k). To linearly modify the functions B_k, we compute $\max_k = \sup_{x\in\Omega} B_k(x)$, as well as $\min_k = \inf_{x\in\Omega} B_k(x)$. We set $\tilde{B}_k = \frac{B_k - \min_k}{\max_k - \min_k}$. We then use these new functions $\tilde{B}_k$ in the logical framework (11).

At this point of the paper, we have introduced all the tools we need in our classification algorithm. We can now present some numerical results to illustrate our approach in the following section.

5.2 Numerical Results

Initialization: To get an automatic initialization, and to make it independent from the user, we have used "seeds" (as in [35,23,4]): we split the initial image into small sub-images (in practice 5*5 images). In each sub-image, for each class k, we compute the data term by assuming that all the pixels of the sub-image belong to the same class k. We set all the pixels in the sub-image to the class k for which the data term of the whole sub-image is the smallest.

Parameters: To use our model, we need to tune many parameters. But this task is not as complicated as it might first appear:

1) The users give the number of classes, the texture and non texture features, and the logical expression. Instead of giving the texture and non texture parameters, the user can give a patch of each class (this is what we have done here in practice). The texture and non texture parameters are then estimated from each patch (see Section 4 for more details). In the case of real images, the patch is a part of the original image. In the case of synthetic images, the patch is the texture or non texture image used to create the original image. In [4], the authors had noticed that the results of the classification of the algoritm are not very sensitive to the choice of the patch from which the texture parameters are estimated.

2) We first set $e_1 = \cdots = e_K = 1$. If we have some additional information about the fact that a class is very likely or not, we can then modify these values (decreasing e_k favors the class k). But in all the examples presented in this paper, we have simply used $e_1 = \cdots = e_K = 1$.

3) In our experiments, we always choose a common value for the regularization term: $\gamma_1 = \cdots = \gamma_K = \gamma$ ($\gamma \in \mathbb{R}_+$), which is related to the scale of the detected objects. There remain only two parameters to tune: the partition term coefficient λ, and the common value of the contour regularization terms γ. The parameter λ is first determined with a value large enough in order to ensure at the end of the algorithm that the partition constrainst is satisfied. The results are not sensitive to variations of λ, provided it is large enough. Second, the regularization parameter γ. Variations of γ give more or less regular solutions. This parameter is tuned by trial and error.

Finding a Cheetah: In this example, we aim at segmenting the cheetah in Figure 1. we therefore have two classes. The first one corresponds to the cheetah. It is characterized by the intersection of the textured pattern (the dots) of the cheetah with the average gray level value of its skin. The second class corresponds to the background of the image. It is characterized by the intersection of the textured pattern (the vegetation) of the landscape with the average gray level value of the background (which is darker than the cheetah). We show the decomposition obtained with the $u + v$ algorithm on Figure 1, and the obtained classification result on Figure 2. Thanks to the combination of both textured and non textured patterns, we get an almost perfect segmentation of the cheetah.

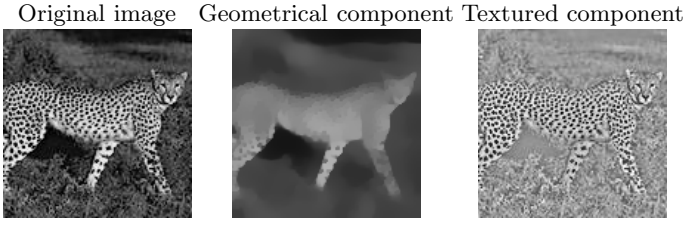


Fig. 1. Original image and $u + v$ decomposition

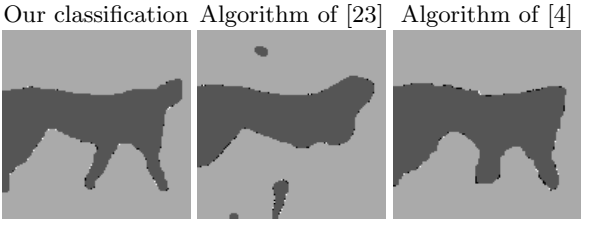


Fig. 2. Final classification of Figure 1 (For our model: $\lambda = 250, \gamma_1 = \gamma_2 = 110$, $e_1 = e_2 = 1$, for the algorithm of [23]: $\lambda = 20, \gamma_1 = \gamma_2 = 15$, $e_1 = e_2 = 1$, and for the algorithm of [4]: $\lambda = 250, \gamma_1 = \gamma_2 = 130$, $e_1 = e_2 = 1$))

For comparisons, we have also used the classification algorithms of [23,4] on the cheetah image of Figure 1. We display the result with the algorithms of [23] and [4]) on Figure 1. The algorithm of [23] is designed for non textured images. It therefore perfoms well on the structures of the image, but fails with the textures: it can not get the legs of the cheetah for instance. The algorithm of [4] is suited for texured images. But it fails at recovering the head of the cheetah wich is a non textured part of the image. This example clearly illustrates the advantage of our approach. By using both textured and non texured features, our algorithm gives a very good result, whereas the two previous ones fail. Notice that, in the case when we aim at finding a given object in the image, we could improve the quality of our segmentation result by using shape priors, as introduced in [15].

Multiple Questions: In this last example, we illustrate the flexibility of our approach. As we have explained in the introduction, depending on the application, there can be many meaningful classification results for the same image. One of the main advantage of our framework is that the user can specify which classification result he is interested in, as we illustrate on Figures 4 and 5. We consider the image of Figure 3. In this case, there are three classes: the one of the zebra, the one of the gnus, and the one of the grass. We characterize each of the three classes by its average gray level intensity and its texture pattern (the texture parameters are computed from patches of the original image as explained in Section 4). We display the classification result on Figure 4. For comparison, we also show the result we get with the algorithm of [23] (for non textured images) and the one of [4] (designed for textured images). Both algorithms fail at recovering the three classes, contrary to our new approach. In both case, the output contains only two classes. With the algorithm of [23], the zebra and the gnus are finally put in the same class: this is due to the fact that the average gray level value cannot discriminate between the zebra and the gnus. With the algorithm of [4], the grass and the gnus are finally put in the same class: this is due to the fact that neither the grass nor the gnus have strong texture patterns, and therefore we cannot discriminate them just from the texture information. In this last case, the problem may also come from the fact that the texture parameters of the gnus class are estimated on a very small patch. In fact, the three classification results of Figure 4 are the answers to three different questions.

1) The classification result of our algorithm answer to the question: *Where is the zebra, where are the gnus, and where is the grass?*
2) The classification result of the algorithm of [23] answers the question: *Where are the animals?*
3) The classification result of the algorithm of [4] answers the question: *Where is the zebra?*

But thanks to our using the logical framework of [24], we can also get the results of the algorithms of [23] and [4] with our approach. We just need to change the logical expressions. Let us denote by G_{zebra} (resp T_{zebra}) the geometrical (resp texture) feature of the zebra class, G_{gnus} (resp T_{gnus}) the geometrical (resp texture) feature of the gnus class, and G_{grass} (resp T_{grass}) the geomet-

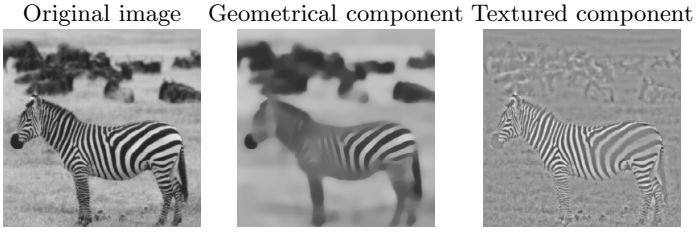


Fig. 3. Original image and $u + v$ decomposition

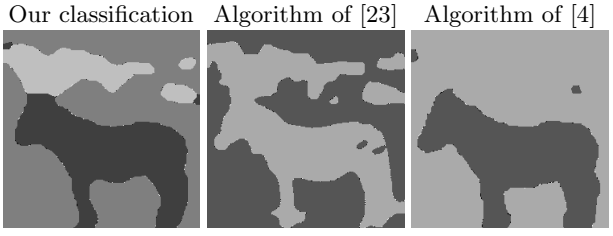


Fig. 4. Final classification of Figure 3 (for our model: $\lambda = 250, \gamma_1 = \gamma_2 = \gamma_3 = 200$, $e_1 = e_2 = e_3 = 1$, for the algorithm of [23]: $\lambda = 15, \gamma_1 = \gamma_2 = \gamma_3 = 30$, $e_1 = e_2 = e_3 = 1$, and for the algorithm of [4]: $\lambda = 300, \gamma_1 = \gamma_2 = \gamma_3 = 150$, $e_1 = e_2 = e_3 = 1$)

rical (resp texture) feature of the grass class. To get the classification result of Figure 4, we have used the following three classes: $Cl_1 = G_{zebra} \bigcap T_{zebra}$, $Cl_2 = G_{gnus} \bigcap T_{gnus}$, and $Cl_3 = G_{grass} \bigcap T_{grass}$. If we are interested in the question *Where is the zebra?*, then we can use: $Cl_1 = G_{zebra} \bigcap T_{zebra}$, and $Cl_2 = (G_{gnus} \bigcup G_{grass}) \bigcap (T_{gnus} \bigcup T_{grass})$. If we are interested in the question *Where are the animals?*, then we can use: $Cl_1 = (G_{gnus} \bigcup G_{zebra}) \bigcap (T_{gnus} \bigcup T_{zebra})$, and $Cl_2 = G_{grass} \bigcap T_{grass}$. We show the classification results for these two questions on Figure 5. This clearly illustrates the flexibility of the logical framework approach [24].

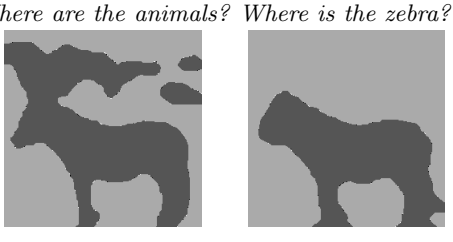


Fig. 5. Classification of Figure 3 with different questions (on the left: $\lambda = 250, \gamma_1 = \gamma_2 = 150, e_1 = e_2 = 1$, on the right: $\lambda = 350, \gamma_1 = \gamma_2 = 250, e_1 = e_2 = 1$)

6 Conclusion

In this paper, we have presented a new supervised classification algorithm for images with both textured and non textured areas. This algorithm is based on a variational approach. Thanks to a decomposition algorithm, we split the image into a geometrical part and a textured part [3]. And then, thanks to the logic framework introduced in [24], we can combine the information from both channels to get the classification. In our framework, a class can be characterized both by textured and non-textured features. This generalizes the approaches of [23,34,13] which are designed to work with non-textures images, as well as the approaches of [20,25,4] which are designed to work with textures images. Since

natural images are combinations of both textured and non textured patterns, our new algorithm enlarges the scope of possible applications for active contours-based classification algorithms. We refer the interested reader to [6] for more numerical results.

Acknowledgment

The NSF under contracts DMS-9973341, ACI-0072112, INT-0072863, the ONR under contract N00014-03-1-0888, the NIH under contract P20 MH65166, and the NIH Roadmap Initiative for Bioinformatics and Computational Biology U54 RR021813 funded by the NCRR, NCBC, and NIGMS.

References

1. G. Aubert and J.F. Aujol. Optimal partitions, regularized solutions, and application to image classification. *Applicable Analysis*, 84(1):15–35, January 2005.
2. G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.
3. J.F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *JMIV*, 22(1):71–88, January 2005.
4. J.F. Aujol, G. Aubert, and L. Blanc-Fraud. Wavelet-based level set evolution for classification of textured images.
IEEE Transactions on Image Processing, 12(12):1634–1641, 2003.
5. J.F. Aujol and A. Chambolle. Dual norms and image decomposition models. *International Journal on Computer Vision*, 63(1):85–104, June 2005.
6. J.F. Aujol and T. F. Chan. Combining geometrical and textured information to perform image classification , 2004. CAM Report 04-65.
7. J.F. Aujol and B. Matei. Simultaneous structure and texture compact representation. In *ACIVS 2004*, September 2004.
8. M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.
9. M. Berthod, Z. Kato, S. Yu, and J. Zerubia. Bayesian image classification using markov random fields. *Image and Vision Computing*, 14(4):285–293, 1996.
10. T. Brox and J. Weickert. A TV flow based local scale measure for texture discrimination. In *ECCV 04*, volume 2, pages 578–590, May 2004.
11. V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.
12. A. Chambolle. An algorithm for total variation minimization and applications. *JMIV*, 20:89–97, 2004.
13. T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–77, February 2001.
14. C. Chesnaud, P. Refregier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Transactions PAMI*, 21(11):1145–1157, 1999.
15. D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *IJCV*, 50(3):295–313, December 2002.

16. M. A. T. Figueiredo, J. M. N. Leitao, and A. K. Jain. Unsupervised contour representation and estimation unsing B-splines and a minimum description length criterion. *IEEE Transactions IP*, 9(6):1075–1087, 2000.
17. M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *ICCV 03*, 2003.
18. B. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random fields models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:478–482, May 1991.
19. Yves Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations, March 2001. The Fifteenth Dean Jacquelines B. Lewis Memorial Lectures.
20. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3), 2002.
21. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *CVPR 03*, volume 2, pages 699–704, June 2003.
22. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
23. C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia. A level set method for image classification. *IJCV*, 40(3):187–197, 2000.
24. B. Y. Sandberg and T. F. Chan. A logic framework for active contours on multi-channel images. *JVCIR*, 16(3):333-358, 2005.
25. B. Y. Sandberg, T. F. Chan, and L. A. Vese. A level-set Gabor-based active contours algorithm for segmenting textured images, 2002. CAM Report 02-39.
26. T. F. Chan B. Y. Sandberg and L. A. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11(2):130–141, 2000.
27. J.L. Starck, M. Elad, and D.L. Donoho. Image decomposition: separation of texture from piecewise smooth content, 2003. To appear in IEEE Transactions on Image Processing.
28. M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.
29. Z. Tu and S-C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions PAMI*, 24(5):657–673, 2002.
30. M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, November 1995.
31. L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
32. L.A. Vese and S.J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19:553–572, 2003.
33. A. Yezzi and S. Soatto. Stereoscopic segmentation. *Intl. J. of Computer Vision*, 53(1):31–43, 2003.
34. Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, July 1996.
35. Song Chu Zhu and A.Yuille. Region competition: unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9), 1996.

Heuristically Driven Front Propagation for Geodesic Paths Extraction

Gabriel Peyré[1] and Laurent Cohen[2]

[1] CMAP, UMR CNRS 7641,
École Polytechnique, 91128 Palaiseau, France
gabriel.peyre@polytechnique.fr
[2] CEREMADE, UMR CNRS 7534, Universite Paris Dauphine, 75775 Paris, France
cohen@ceremade.dauphine.fr

Abstract. In this paper we present a simple modification of the Fast Marching algorithm to speed up the computation using a heuristic. This modification leads to an algorithm that is similar in spirit to the A* algorithm used in artificial intelligence. Using a heuristic allows to extract geodesics from a single source to a single goal very quickly and with a low memory requirement. Any application that needs to compute a lot of geodesic paths can gain benefits from our algorithm. The computational saving is even more important for 3D medical images with tubular structures and for higher dimensional data.

1 Shortest Path: Continuous and Discrete Algorithms

A large class of problems can be formulated as the extraction of a shortest path for a given discrete or continuous metric. The applications of shortest paths thus range from tubular structures extraction in 3D medical images [1] to path finding in video games [2]. The ability to quickly extract a geodesic path is highly desirable. In this paper we propose a modification of front propagation methods. We introduce a heuristic that drives the computations, and greatly reduces the computation time the Fast Marching algorithm.

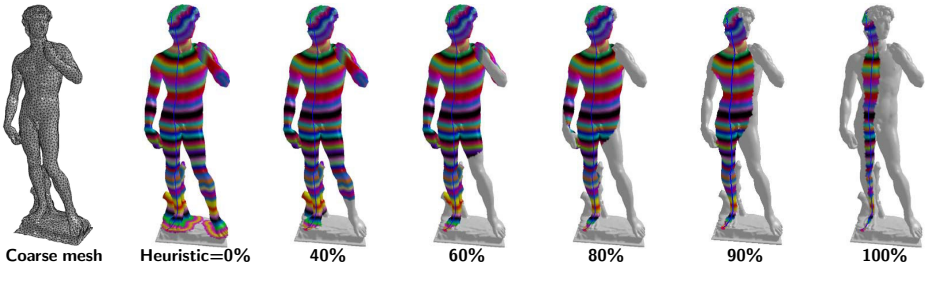


Fig. 1. Heuristically-driven front propagation on a 3D mesh shown with increasing values of heuristic proportion. The various colors indicate the level sets of the geodesic distance function. Only the colored region is explored by our front propagation algorithm.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 173–185, 2005.

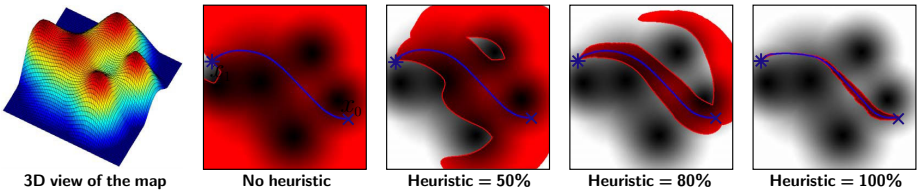


Fig. 2. An example of 2D path planning. The set of alive points according to increasing heuristic is shown in gray.

In this section we briefly review some basic facts about the Fast Marching algorithm and geodesics extraction and we also present a common framework that will allow to introduce our new approach.

1.1 Fast Marching Algorithm

The classical Fast Marching algorithm is presented in [3], and a similar algorithm was also proposed in [4]. This algorithm performs a front propagation that can be used to extract geodesic paths. The minimal length properties of geodesic has been applied in computer vision, for example to solve global minimization problems for deformable models [5]. We will illustrate various applications of geodesic paths in section 3.

In the continuous setting, a geodesic curve minimizes the weighted length of the curve for a given metric. In $\mathbb{R}^d$, we are given a potential function $g(x) > 0$, and the weighted geodesic distance between two points $x_0,\, x_1 \in \mathbb{R}^d$, is defined as

$$d(x_0,\, x_1) \overset{\text{def.}}{=} \min_{\gamma} \left(\int_0^1 \|\gamma'(t)\|\, g(\gamma(t))\mathrm{d}t \right), \tag{1}$$

where γ is a piecewise regular curve with $\gamma(0) = x_0$ and $\gamma(1) = x_1$. When $g = 1$, the integral in (1) corresponds to the length of the curve γ and d is the classical Euclidean distance.

To compute the distance function $U(x) \overset{\text{def.}}{=} d(x_0,\, x)$ with an accurate and fast algorithm, this minimization can be reformulated as follows. The level set curve $\mathcal{C}_t \overset{\text{def.}}{=} \{x \setminus U(x) = t\}$ propagates following the evolution equation $\frac{\partial \mathcal{C}_t}{\partial t}(x) = \frac{1}{g(x)}\overrightarrow{n_x}$ and the function U satisfies the nonlinear *Eikonal* equation:

$$\|\nabla U(x)\| = g(x). \tag{2}$$

The function $F = 1/g > 0$ can be interpreted as the propagation speed of the front $\mathcal{C}_t$.

The Fast Marching algorithm on an orthogonal grid makes use of an upwind finite difference scheme to compute the value u of U at a given point $x_{i,j}$ of a grid (the equation is written in $\mathbb{R}^2$ for simplicity), solving:

$$\begin{aligned} &\max(u - U(x_{i-1,j}),\, u - U(x_{i+1,j}), 0)^2 + \\ &\max(u - U(x_{i,j-1}),\, u - U(x_{i,j+1}), 0)^2 \,=\, h^2 g(x_{i,j})^2. \end{aligned} \tag{3}$$

This is a second order equation that is solved as detailed for example in [6]. An optimal ordering of the grid points is chosen so that the whole computation only takes $O(N \log(N))$, where N is the number of points.

1.2 Front Propagation Methods for Shortest Path

We now give a common framework for front propagation algorithms, which includes the Fast Marching procedure [3], the Dijkstra algorithm [7] to compute shortest paths on graphs, as well as our heuristical front propagation procedure. These methods can be described using the following formalism:

- *Alive* is the set of grid points at which the distance value U has been computed and will not change;
- *Trial* is the set of next grid points to be examined and for which an estimate of U has been computed;
- *Far* is the set of all other grid points, for which there is not yet an estimate for U.

Table 1 shows the main steps of the algorithms. Each algorithm must implement the following implementation-dependant sub-functions

- A way to update the value $U(y)$ at a given Trial point y. In the Fast Marching and our Heuristical front propagation, $U(y)$ is computed by solving equation 3.
- A priority map $\mathcal{P}$ orders the set of Alive points according to some computational criterion. In the Fast Marching and Dijkstra algorithm, $\mathcal{P}(x) = U(x)$ is the current distance to the starting point. In our heuristical front propagation, $\mathcal{P}(x)$ is chosen to minimize the number of visited points.

We will explain in section 2 how to actually construct a priority function $\mathcal{P}$ that makes use of a heuristic.

Table 1. Pseudo-code for the common framework for front propagation

Initialization:

- Alive set: the starting point x_0;
- Trial set: the neighbors of x_0;
- Far: the set of all other grid points.

Loop:

- Let x be the Trial point with the smallest priority $\mathcal{P}(x)$;
- Move it from the Trial to the Alive set;
- For each neighbor y of the current point x:
  - if y is Far, then add it to Alive and compute a new value for $U(y)$,
  - if y is Alive, recompute the value $U(y)$, and update it if the new value is smaller,
  - recompute the priority $\mathcal{P}(y)$.
- If the end point $x = x_1$ is reached, stop the algorithm.

1.3 Geodesic Extraction

For the applications we have in mind, the computation of the geodesic distance U to a point x_0 is only a tool needed before the extraction of a geodesic that links this point to another point x_1. This curve can be computed by extracting the parametric curve $C(t)$ that solves the back propagation equation:

$$\frac{dC}{dt} = -\overrightarrow{\nabla U} \qquad \text{with} \qquad C(0) = x_1.$$

This gradient descent is a very local computation, and it only uses the value of U for a small fraction of the visited grid points. Note that these grid points are those located in the Alive set at the end of the front propagation procedure.

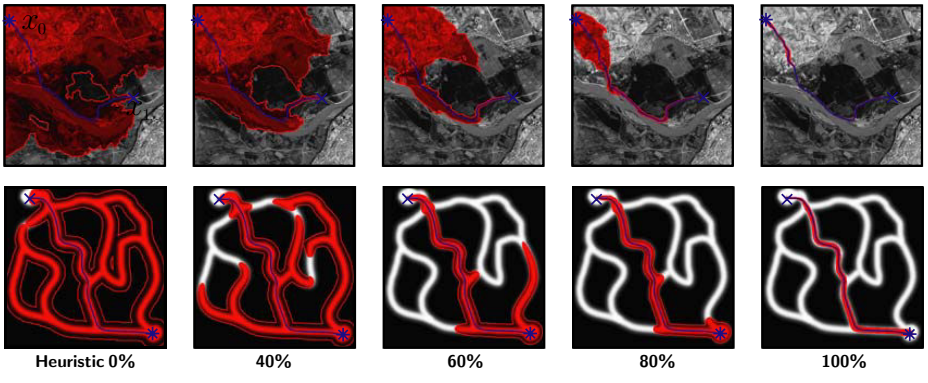


Fig. 3. Path planning using a multiresolution heuristic for various heuristic strength λ

2 Heuristically Driven Front Propagation

In this section we explain our algorithm in the 2D setting, and show some numerical results that illustrate the main features of this method. We also give some insights about how to choose the parameters of our method.

2.1 Fast Marching with a Heuristic

In order to minimize the number of Alive points at the end of the front propagation procedure, one should use a priority function $\mathcal{P}$ that will try to advance the front toward the goal point x_1, and not isotropically. In order to do so, assume that together with the the current weighted distance to the start point $U(x) = d(x_0, x)$, we are able to have an estimate of the weighted distance that remains to be marched $V(x) \approx d(x_1, x)$. Then our heuristical front propagation algorithm follows the implementation of the table 1 with a priority map

$$\mathcal{P}(x) = U(x) + \lambda\, V(x). \tag{4}$$

This introduces the first parameter of our algorithm: the weight λ we use for the heuristic, which typically range from 0% to 100%.

The rational behind the definition of $\mathcal{P}$ is that $d(x_0, x) + d(x_1, x)$ is minimal and constant along the geodesic path joining x_0 and x_1, see [8]. On figure 2, one can see the effect of various choices for the parameter λ. The value $\lambda = 0$ corresponds to the classical Fast Marching propagation, which result in a very large region of Alive points (colored in red). However, as we increase the value of λ toward 100%, the explored region shrinks around the geodesic path that links x_0 to x_1. There are however two important issues with this ordering of the Trial set:

- This ordering can break the monotone condition that is required by the Fast Marching algorithm to produce a valid approximation of the continuous underlying distance function. We show in the numerical results presented in section 2.4 that although these numerical errors can accumulate during the propagation, the Hausdorff error on the extracted geodesic remains low both in synthetic case studies and in real applications.
- We do not have an immediate access to the remaining distance $d(x, x_1)$, since it would involve performing another front propagation from x_1. We explain in the next section how to overcome this problem.

Our heuristic ordering strategy is a reminiscence of the A* algorithm [9], which was first introduced to solve problems in artificial intelligence. The use of heuristics and fast exploration strategies (such as IDA* [10]) is crucial in the artificial intelligence field to avoid the complexity of exploring the whole configuration space. These techniques are also heavily used for path finding, for example in video games [2]. We show in section 3 that a large class of applications in computer vision can also gain benefit from this paradigm.

2.2 Multiresolution Heuristic

In order to compute the remaining distance $V(x) \approx d(x, x_1)$ with a fast algorithm, we perform a Fast Marching front propagation starting from the point x_1, but on a coarser grid. We thus have introduced a second parameter for our heuristical front propagation: the resolution $R \in]0, 1[$ we use for the coarse grid. If the original potential map g is of size $n \times n$, the query of $\mathcal{P}(y)$ thus requires:

- The pre-computation of a coarse potential map g_R of size $(Rn) \times (Rn)$. This is done by first a pre-filtering of g (to avoid aliasing of high frequencies) and then a cubic spline re-interpolation on a coarser grid.
- The pre-computation of the approximate distance map V of size $(Rn) \times (Rn)$. This is done by performing a full Fast Marching on a coarse grid, using potential g_R, and starting from point x_1.
- During the heuristical front propagation starting from point x_0, when $\mathcal{P}(y)$ is queried, we interpolate with cubic splines the value of V on the coarse grid to retrieve a value on the original grid.

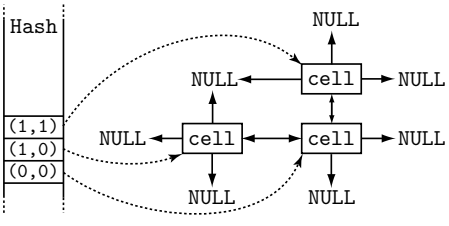
There is clearly a tradeoff between choosing a low R to reduce the computation time, and a high R so that $V(x)$ approximates well $d(x, x_1)$. We show in section 2.4 some insight about the correct value for this parameter R.

The new algorithm we propose allows to use multiresolution computation for the extraction of geodesic curve. Using a multiresolution framework for solving the point-to-point geodesic problem is not so easy because it is a boundary problem, and for instance, multigrid methods are not suitable. Adaptive mesh [11] and multigrid [12] have been used in conjunction with geodesic active contours for segmentation purpose. However the problem of segmentation is in some sense dual to the geodesic extraction, in which we are not directly interested in the level sets of the distance function.

2.3 Reducing Memory Usage

Classical methods, such as using an octree data structure, can be used to reduce the memory usage of level set algorithms, for example in order to perform image segmentation [11].

We chose to implement a simple data structure to reduce the memory usage by allocating the grid cell on the fly during the propagation. A typical cell data structure, for 2D computation, is:




```
struct cell {
  double distance;        // current geodesic distance
  char state;             // either far, open or close
  cell* neighbors[4]; };  // pointers to the 4 neighbors
```

To be able to retrieve a given cell in constant time, we also store the list of allocated cells in a hash table. This is important because when a new cell is allocated, we need to connect it to the existing cells. This pointer-based representation of the neighboring relation is very convenient to extract the geodesic with a gradient descent. There is some memory overhead due to the fact that we explicitly store pointer to neighbors, but the fact that our scheme can explore significantly less cells that the classical fast marching allows to save much more memory, as shown in next section. The computing time overhead due to the use of a hash table is about 40% in all our tests.

2.4 Numerical Validation

A Matlab implementation of our algorithm, together with the scripts needed to reproduce the figures of this paper, is freely available on Matlab Central [13].

In order to estimate the precision of the results, we use the Hausdorff error between the found paths and the paths obtained by fast marching without heuristic. On figure 6 one can see the geodesics extracted for different values of λ. Figure 4 shows the result of our algorithm for various settings on (a) a synthetic map and (b) a satellite image. We have depicted:

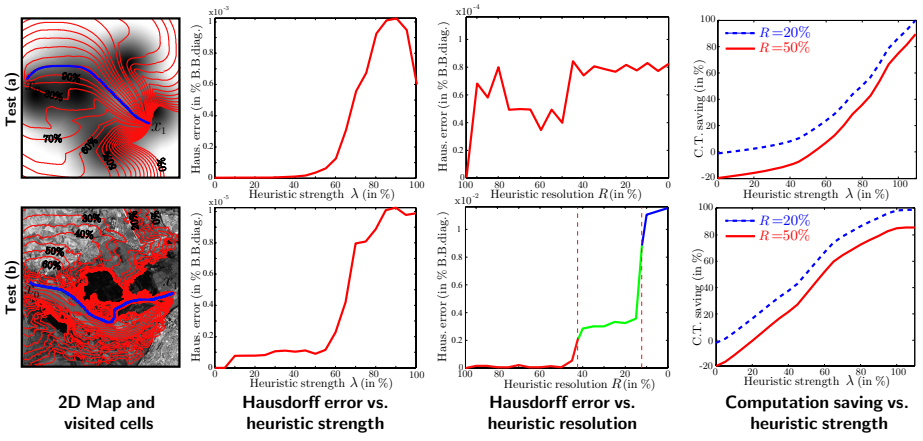


Fig. 4. Influence of heuristic strength and resolution on number of visited cells, Hausdorff error and computation time reduction

Fig. 5. Influence of the resolution of the heuristic on the shape of the geodesic

- The 2D map: the red curves indicate the boundary of the visited region. One can see that these curves shrink toward the geodesic (central blue curve) as one increase the strength of the heuristic from 0% to 100%.
- Hausdorff error vs. heuristic strength λ: we have set the heuristic resolution R to 50%. One can see that the error is higher for the synthetic map (a). This is due to the fact that this map contains large flat areas, where a small error in the computed geodesic distance leads to deviation of the extracted geodesic. In contrast, the geodesic in the satellite image (b) contains very anisotropic areas, which stabilize the extracted geodesic.
- Hausdorff error vs. heuristic resolution R: we have set the heuristic strength λ to 50%. One can see that the synthetic map (a) is nearly insensitive to the resolution of the coarse map used to compute the heuristic. This is because the underlying function is very smooth, so one can reduce a lot the resolution without too much impact on the accuracy of the Heuristic. In contrast, one can see that the satellite image suffers from too huge variation when the resolution parameter R becomes smaller than 60% and then again for 90%. This is due to strong topological change in the path, as depicted in figure 5.

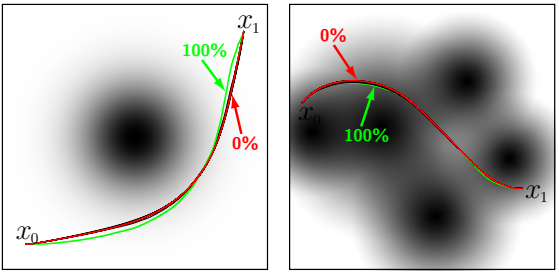


Fig. 6. Graphical display of extracted geodesics for various heuristic strengths

- Computation time saving vs. heuristic strength λ: the saving is computed relatively to the time spent by the classical Fast Marching. In 2D the computation times decrease roughly linearly with the strength of the heuristic.
- Computation time saving vs. heuristic resolution R (not shown): there is a constant overhead due to the coarse resolution computation (which results in an offset between the curves for $R = 50\%$ and $R = 20\%$). For $R = 20\%$, this overhead is balanced by the heuristic saving as soon as $\lambda \geqslant 5\%$.

These tests clearly show that our algorithm can bring a large computational speed up, but the parameters should be finely tuned to adapt to the characteristic of each map. For instance, these experiments show that the user must have some prior knowledge about the typical width of the tubular structures he wants to extract, and set the resolution R so that the coarse map V_R still contains these structures.

3 Applications

In this section, we show the versatility of our method by proposing various applications where the extraction of geodesics is a central issue. We explain why our algorithm can bring a speed improvement and allows us to use simpler methods.

3.1 Volumetric Geodesics Extraction

3D geodesic extraction is very useful in medical volumetric data analysis. It can be applied to perform tubular structures extraction, and it is extended to virtual endoscopy in [1]. On figure 7 one can see the extraction of 3D geodesics on synthetic data (top and middle rows) and on real medical data (bottom row) for $R = 20\%$. The red surface shows the boundary of the explored regions of alive cells. The computation time gain (Comp. gain) is also indicated.

3.2 2D Path Planning

Direct application of AI techniques has been intensively used for path finding, and A* is the method of choice for video games [2]. For the Euclidean case, faster

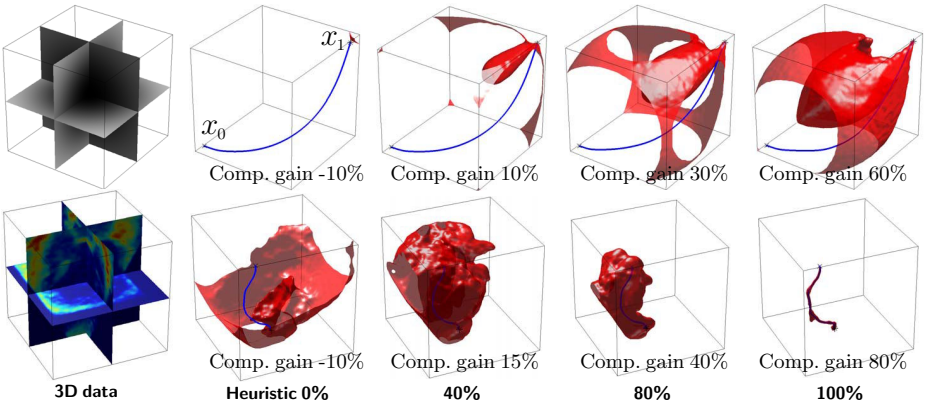


Fig. 7. Extraction of geodesics in 3D

and more complex graph-based methods have been proposed, such as visibility graphs [14].

The Fast Marching method can be used to produce continuous paths with a sub pixel precision [3], even when a non Euclidean metric is used. This is very powerful since it allows the modeling of smooth obstacles (such as forest, enemy repulsion, etc). In [15], the authors compare the Fast Marching and the A* algorithms for path planning. However they do not go one step further and connect these two powerful methods. On figure 3 one can see various paths extracted from a 2D map for a resolution $R = 20\%$ and a varying λ. Although the computed path can deviate from the real geodesic, our algorithm always produces locally acceptable path (smooth and avoiding obstacles).

3.3 Constrained Path Planning

Geodesics can be used to compute the path of a robot with various shape and motion constraints [16]. Basically, each additional degree of freedom add a new dimension to the domain in which the front propagation should be performed. Solving such high dimensional problems is time and memory consuming, so the use of a heuristic is highly desirable. In our experiment, the most important issue is the memory used by the full-grid classical Fast Marching, and the memory management strategy exposed in subsection 2.3 is crucial to scale to complex problems. The resolution R of the heuristic should be chosen carefully as a function of the typical width of the corridors and rooms the robot must pass through. In our tests we set $R = 20\%$ and we use a varying heuristic strength λ. Note however that although our algorithm can produce a wrong path for agressive heuristic usage, it never produces non-admissible moves.

On figure 8, one can see two examples of path extractions in 2D with one rotational additional degree of freedom. This results in 3D front propagation, and the corresponding speed function is depicted on the left. Figure 9 shows the influence of the heuristic strength λ on the cells explored by the front propagation.

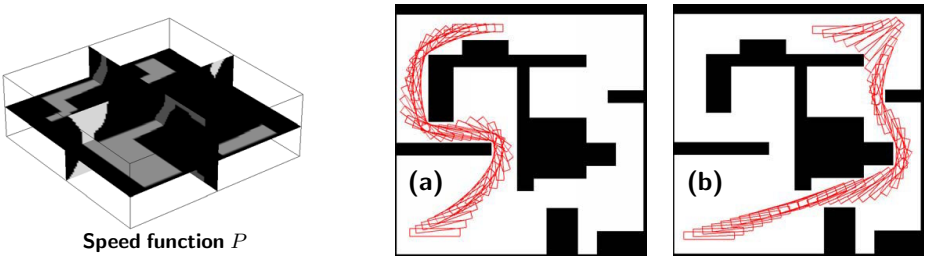


Fig. 8. Examples of constrained path planning

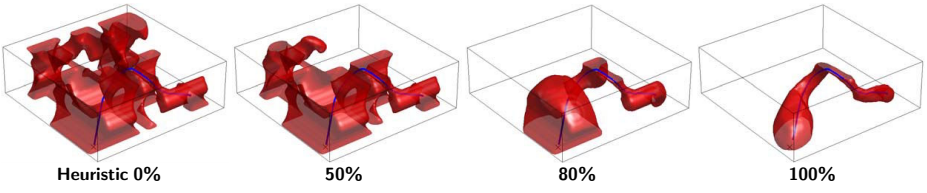


Fig. 9. Explored area for constrained path planning

3.4 Globally Optimal Geodesic Active Contours

The concept of circular geodesics was first introduced in [17]. The authors of [18] proposed a simple way to compute circular geodesics around a point, in order to compute a globally optimal geodesic, with an application to object segmentation. The user simply select a point C inside the object to segment and then the algorithm virtually "cuts" the image along a horizontal line that links C to the boundary of the image. This way, one can force a geodesic path to go around C by running a classical Fast Marching from a point S to itself, but forbidding the front to pass through the segment CD.

For an underlying image I, the globally optimal geodesic around C is defined as the closed geodesic curve with minimum length, where the metric is defined as

$$g(x) = \frac{1}{\|C - x\|} \frac{1}{1 + \|\nabla I(x)\|^2} + \varepsilon, \tag{5}$$

where $\|C - x\|$ is the distance from the curve point x to the center C.

The authors of [18] proposed a powerful algorithm based on the branch-and-bound paradigm, which is a dichotomic search that avoids computing the closed geodesic for each point S on the segment CD. However, with our heuristic front propagation, we have tested a simpler algorithm that works well in practice. We simply compute the circular geodesics that pass though a given fixed number of points along the cut segment CD. These extractions can be performed quickly using our heuristically driven front propagation, with the restriction that the front should not pass though the cut segment.

In figure 10, we have shown a globally optimal circular geodesic, computed with various heuristic strength λ.

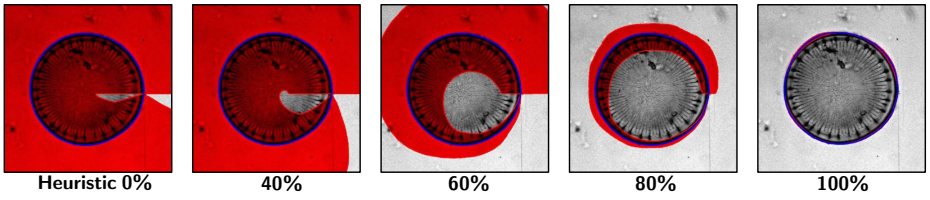


Fig. 10. Globally optimal circular path extraction with increasing heuristic

3.5 Geodesic Extraction on 3D Meshes

The Fast Marching algorithm has been extended to 3D meshes in [19]. Our heuristic algorithm also extends to 3D meshes, with the following modifications with respect to the Euclidean setting:

- We must construct a coarse mesh approximation of the original 3D mesh. Mesh simplification is a large topic, and several greedy methods exist, see for example [20]. In our tests, we use the farthest point strategy proposed in [21] for remeshing, since it uses the Fast Marching as a building block.
- Once the heuristic function has been computed on the coarse mesh, it must be interpolated on the original dense mesh. Several methods for data interpolation on 3D meshes exist, and we have used a method derived from harmonic mesh parameterization [22]. This involves the resolution of a sparse linear system that searches a harmonic function that fits the values computed on the coarse mesh.

These two steps are quite computationally intensive, but note that:

- The coarse mesh can be pre-computed, and can be re-used for multiple geodesic extraction.
- To avoid the computational overhead of computing once for all the interpolation on the whole mesh, we use the local parameterization strategy of [23]. We compute the interpolation only on a small set of overlapping disk-like charts that covers the region of alive vertices.

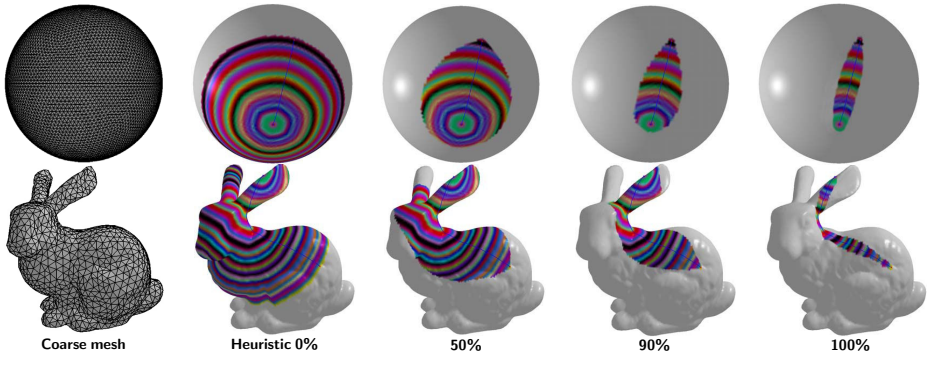


Fig. 11. Heuristically driven front propagation on 3D meshes

On figure 1 and 11, one can see the algorithm in action on various meshes, and for various values of the parameter λ.

4 Conclusion

In this paper we have presented a simple modification of the Fast Marching to speed up the computation time and the memory requirement of the algorithm. This modification is straightforward to implement, since it only involves the computation of a multiresolution heuristic to propagate the front toward the correct direction. Numerical tests on synthetic and real data show that this modification does not result in large distortion of the extracted curve. We examine some potential applications of this algorithm and show that it fits nicely into various existing computational frameworks.

References

1. Deschamps, T., Cohen, L.: Fast Extraction of Minimal Paths in 3D Images and Applications to Virtual Endoscopy. Medical Image Analysis **5** (2001)
2. Stout, W.B.: Smart moves: Intelligent path-finding. Game Developer (1996)
3. Sethian, J.: Level Sets Methods and Fast Marching Methods. 2nd edn. Cambridge University Press (1999)
4. Tsitsiklis, J.: Efficient Algorithms for Globally Optimal Trajectories. IEEE Trans. on Automatic Control **40** (1995) 1528–1538
5. Cohen, L.D., Kimmel, R.: Global Minimum for Active Contour models: A Minimal Path Approach. International Journal of Computer Vision **24** (1997) 57–78
6. Cohen, L.: Multiple Contour Finding and Perceptual Grouping Using Minimal Paths. Journal of Mathematical Imaging and Vision **14** (2001) 225–236
7. Cormen, T.H., Leiserson, C.E., Rivest, R.R.: Introduction to Algorithms. MIT Press, Cambridge, Massachusetts (1990)
8. Kimmel, R., Amir, A., Bruckstein, A.M.: Finding shortest paths on surfaces using level sets propagation. IEEE Trans. on PAMI **17** (1995) 635–640
9. Nilsson, N.: Problem-solving Methods in Artificial Intelligence. McGraw-Hill, New York (1971)
10. Korf, R.E.: Depth-first iterative-deepening: an optimal admissible tree search. Artif. Intell. **27** (1985) 97–109
11. Droske, M., Meyer, M., Rumpf, M., Schaller, C.: An adaptive level set method for interactive segmentation of intracranial tumors. Neurosurgical Research **27** (2005)
12. Papandreou, G., Maragos, P.: A fast multigrid implicit algorithm for the evolution of geodesic active contours. In: CVPR04. (2004) II: 689–694
13. Peyré, G.: Fast marching toolbox, available on Matlab Central. `http://www.mathworks.com/matlabcentral/` (2005)
14. Rohnert, H.: Shortest paths in the plane with convex polygonal obstacles. Inf. Process. Lett. **23** (1986) 71–76
15. Melchior, P., Orsoni, B., Lavialle, O., Poty, A., Oustaloup, A.: Consideration of obstacle danger level in path planning using a* and fast-marching optimisation: comparative study. Signal Processing **11** (2003) 2387–2396

16. Kimmel, R., Sethian, J.A.: Optimal algorithm for shape from shading and path planning. Journal of Mathematical Imaging and Vision **14** (2001) 237–244
17. Sun, C., Pallottino, S.: Circular shortest path in images. Pattern Recognition **36** (2003) 711–721
18. Appleton, B., Talbot, H.: Globally optimal geodesic active contours. Journal of Mathematical Imaging and Vision, To Appear (2005)
19. Sethian, J., Kimmel, R.: Computing Geodesic Paths on Manifolds. Proc. Natl. Acad. Sci. **95** (1998) 8431–8435
20. Hoppe, H.: Progressive meshes. Proc. of SIGGRAPH 1996 (1996) 99–108
21. Peyré, G., Cohen, L.D.: Geodesic Remeshing Using Front Propagation. Proc. IEEE Variational, Geometric and Level Set Methods 2003 (2003) 33–40
22. Floater, M.S., Hormann, K., Reimers, M.: Parameterization of Manifold Triangulations. Approximation Theory X: Abstract and Classical Analysis (2002) 197–209
23. Surazhsky, V., Alliez, P., Gotsman, C.: Isotropic remeshing of surfaces: a local parameterization approach. In: Proceedings of 12th International Meshing Roundtable. (2003)

Trimap Segmentation for Fast and User-Friendly Alpha Matting

Olivier Juan and Renaud Keriven

CERTIS, ENPC,
77455 Marne-la-Vallée, France
{juan, keriven}@certis.enpc.fr

Abstract. Given an image, digital matting consists in extracting a foreground element from the background. Standard methods are initialized with a *trimap*, a partition of the image into three regions: a definite foreground, a definite background, and a *blended region* where pixels are considered as a mixture of foreground and background colors. Recovering these colors and the proportion of mixture between both is an under-constrained inverse problem, sensitive to its initialization: one has to specify an accurate trimap, leaving undetermined as few pixels as possible.

First, we propose a new segmentation scheme to extract an accurate trimap from just a coarse indication of some background and/or foreground pixels. Standard statistical models are used for the foreground and the background, while a specific one is designed for the blended region. The segmentation of the three regions is conducted simultaneously by an iterative Graph Cut based optimization scheme. This user-friendly trimap is similar to carefully hand specified ones.

As a second step, we take advantage of our blended region model to design an improved matting method coherent. Based on global statistics rather than on local ones, our method is much faster than standard Bayesian matting, without quality loss, and also usable with manual trimaps.

1 Introduction

The commonly used model of digital or alpha matting is the following. An image I is considered as a mixture between a foreground I_F and a background I_B, mixture quantified by an *alpha mask* $\alpha \in [0, 1]$. For each pixel x, this writes

$$I(x) = \alpha(x)I_F(x) + (1 - \alpha(x))I_B(x) \tag{1}$$

Such a blending has multiple reasons: transparent objects, aliasing, blur or motion blur. The problem is to recover I_F, I_B and α from I.

This inverse problem is under-constrained and can not be solved without priors. Historically, a solution was proposed in the case of a known constant background, e.g. a blue screen [1]. Recently, inspired by computer vision techniques, methods based on a model of the foreground and of the background were

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 186–197, 2005.

proposed that greatly improve the matte quality, even without a blue screen. Since the pioneering work of Ruzon and Tomasi [2], several methods have been proposed [3,4,5,6].

As a prerequisite of any method, the user has to specify a so-called *trimap*, partitioning the image into three regions: a set Ω_F of definitely foreground pixels (where α will always be 1), a set Ω_B of definitely background pixels ($\alpha = 0$), and a blended region Ω_M where α, I_F and I_B are unknown. Ω_M has to be an intermediate region, separating Ω_F from Ω_B. Matting methods suffer from sensitivity to this initial condition and one has to specify it accurately, leaving undetermined as few pixels as possible. Moreover, when too small Ω_F and Ω_B are given, the matting process generally does not work at all.

Digital matting was primitively developed for movie production. For a specialist, carefully specifying a trimap is a long but feasible process (actually faster and easier than alpha masking). Today, extracting a subject from a picture for editing purpose becomes a standard in a non professional context. Speed becomes also an issue, particularly with the ever increasing resolution of digital cameras.

This paper addresses both user-friendly trimap design and speed. First, we propose a trimap segmentation scheme from just a small subset of the background and/or foreground, that can be for instance specified by the user with a brush-like tool. Standard Gaussian Mixture Models (GMMs) are used for foreground and background modeling, while a specific statistical model is proposed for the blended region. To save the user from specifying some obscure number of components, the GMMs parameters are determined with a coupled Expectation Maximization (EM) / Minimum Description Length (MDL) scheme. For the sake of speed, the segmentation of the three regions is conducted simultaneously by an iterative Graph Cut based optimization. The resulting trimap proves to be similar to carefully hand specified ones.

As a second step, we take advantage of our blended region model to design an improved matting method. Based on global statistics rather than on local ones, our method is much faster than the original Bayesian matting, although without quality loss. It can also be used with manually designed trimaps.

2 Related Work

The original work of Ruzon and Tomasi [2] laid the foundations of most of the actual methods, for which the key point consist in modeling the background and the foreground with some statistical model. In their famous *Bayesian Matting*, Chuang et al. [3] improved both the statistical model and the way to use it to recover the alpha mask and the original background and foreground colors. Since then, Rother et al. proposed *GrabCut* [7,8], a method inspired by Boykov and Jolly work [9], where the image is actually segmented into two regions using an iterated Graph Cuts [10] scheme. A smooth alpha mask is then modeled as a ramp of variable width to be estimated. As a result, it is unadapted to non smooth objects like hairs or trees. The GrabCut method does not need

a trimap. It can be seen more as a two regions segmentation with a smooth transition between the two regions, than as a strictly speaking digital matting method. However, as another member of iterated Graph Cuts methods [9], our trimap segmentation has similarities with the segmentation step of GrabCut.

In their *Poisson Matting*, Sun et al. propose another prior on α, based on its gradient and Poisson equations, already used in image editing [5]. Their method supply different modes, refinements and filters, manually invoked by the user. Again, priors on α or its gradient can be questionable as the blending might have different origins and the blended objects different scales with respect to pixels size. Moreover, manual decisions might limit the usability of this technique for non specialists. In conclusion, Bayesian matting can be considered as the less *ad-hoc* method so far. Its weak points are the need of an accurate trimap (a problem common with other matting techniques) and its slowness due to many local statistics estimations. Figure 1 demonstrates how a coarse trimap affects digital matting.

To our knowledge, the only works addressing trimap design are video oriented. Following the original work by Mitsunaga et al. [11], one can specify trimaps for some *key frames* and interpolate them in the intermediate frames. In their recent work, Xiao and Shah [12] proposed an occlusion based trimap extraction couple with motion layer segmentation. However, it is unusable not only for still pictures but also in real film production where motion is often fast and/or heavily blurred.

This paper is organized as follows. First, section 3 exposes the background-foreground model and our parameters estimation method. Then, section 4 introduces the trimap segmentation, details its implementation, and compares it with manual segmentations. Finally, section 5 proposes an improved fast, global and accurate matting method, and shows results.

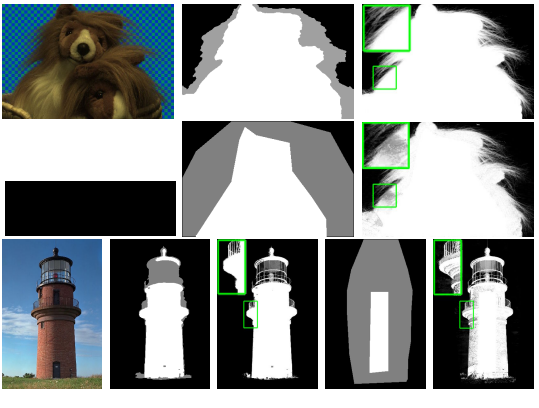


Fig. 1. Sensitivity of Bayesian matting [3] to the trimap. First row: the original image, an accurate trimap and its corresponding alpha. Second row: a coarse trimap and its alpha. Third row: same as rows 1 and 2 for another image.

3 Unsupervised Two-Regions Segmentation

As a first step toward our trimap segmentation, we first focus on segmenting an image into two regions, each of them having its own characteristics, a-priori unknown. This often called *Unsupervised Segmentation* has recently received a lot of attention from the Computer Vision community. Many approaches have been proposed, among which some Level Set [13,14] based methods (e.g. [15,16]). More recently, using the Graph Cuts framework, Boykov and Jolly initiated an iterated method [9], further developed by Rother et al. in [7] and in their GrabCut scheme [8].

In this section, we briefly describe the segmentation part of the GrabCut scheme. Already known to the GrabCut aware reader, the content of this section introduces definitions and notations. The slight difference with the original work is that we plead for a more sophisticated parameter estimation method, EM + MDL based, mathematically more justified, more user-friendly, and yielding somehow better results.

Let I be a color image defined over a domain Ω. For all $x \in \Omega$, $I(x)$ is a pixel defined in a color space (e.g. RGB or CieLab). Let Ω_U be a part of Ω specified by the user. Our goal here is to segment Ω into two *"coherent"* regions that we will abusively still call the background and the foreground, respectively still denoted by Ω_B Ω_F, such that $\Omega_U \subset \Omega_B$.

3.1 Region Modeling

Following previous work and using a statistical approach, each region Ω_X ($X = F$ or B) is modeled by a Probability Density Function (PDF) approximated by a Gaussian Mixture Model (GMM):

$$p_X(I) = \sum_{i=1}^{N_X} \pi_i^X G_{\mu_i^X, \Sigma_i^X}(I) \text{ with } \sum_{i=1}^{N_X} \pi_i^X = 1 \text{ and } \pi_i^X \in [0,1]$$

Each component is represented by a Gaussian of mean μ_i^X and covariance Σ_i^X: $G_{\mu,\Sigma}(I) = \frac{|\Sigma|^{1/2}}{(2\pi)^{3/2}} e^{-(I-\mu)^T \Sigma^{-1}(I-\mu)/2}$ and π_i^X is the prior of the i^{th} component with respect to all components, i.e. its proportion in the mixture.

Estimating the parameters $\Theta_X = \{N_X, (\pi_i^X, \mu_i^X, \Sigma_i^X)_{i=1..N_X}\}$ is a widely studied problem. For a given N_X, one can use the K-Means algorithm (see [17]), a fast but approximate method. This method is widely sensitive to its initialization. Moreover it does not provide a likelihood maximum, which is not appropriate for a segmentation based on likelihood maximization. Indeed the K-Means just solves:

$$(\mu_i^X, \Sigma_i^X) = \underset{(\mu_i, \Sigma_i)}{\arg\min} \sum_{x \in \Omega_X} \left\| I(x) - \mu_{k(I(x))} \right\|^2_{\Sigma_{k(I(x))}}$$

with $k(I) = \arg\min_k \|I - \mu_k\|^2_{\Sigma_k}$, $\|I - \mu\|_\Sigma$ being the Mahalanobis distance between I and μ with respect to Σ. Note that [7] suggests [18] as a variant and that [3] uses the method in [19]. We prefer the EM algorithm [20,21]. It is much

more robust with respect to the initial parameters and provides a likelihood maximum, solving:

$$(\pi_i^X, \mu_i^X, \Sigma_i^X) = \underset{(\pi_i,\mu_i,\Sigma_i)}{\arg\min} \sum_{x\in\Omega_X} p(I(x))$$

Finally, we combine the EM algorithm with a MDL [22] estimation of N^X, saving the user from manually adjusting the number of Gaussian components. Note that we have also tested more recent algorithms like Split and Merge EM [23], without any significant improvement.

3.2 Energy Design

Let γ be the partition function of Ω into Ω_F and Ω_B: $\gamma(x) = F$ if $x \in \Omega_F$, $\gamma(x) = B$ otherwise. Under the hypothesis that regions are independent with respect to their color distribution, it is natural to use the posterior probability of the pixels as a segmentation criterion, thus stating the problem as minimizing an energy:

$$E_{data}(\gamma) = \int_\Omega -log\ p_{\gamma(x)}(I(x))\ dx \tag{2}$$

An extra control term should be added to constrain the smoothness of the solution which is often addressed as a local smoothness constrain: neighbor pixels should belong to the same region. This yields an additional smoothness energy:

$$E_{smooth}(\gamma) = \int_\Omega \left(\int_{y\in\mathcal{N}(x)} \mathcal{V}(x,y)\ dy \right) dx \tag{3}$$

where $\mathcal{N}(x)$ is a local neighborhood of x and $\mathcal{V}(x,y) = \mathcal{V}^0(x,y)$ if $\gamma(x) \neq \gamma(y)$ with $\mathcal{V}(x,y) = 0$ otherwise.

Under the assumption that the frontier between the two regions corresponds to high image gradients, a frequent choice is $\mathcal{V}^0(x,y) = \kappa \exp(-\frac{\|I(x)-I(y)\|^2}{2\sigma^2})$ where κ is some positive constant controlling the degree of smoothness and σ is set as in [9]. The global energy to minimize ends to:

$$E(\gamma) = E_{data}(\gamma) + E_{smooth}(\gamma) \tag{4}$$

3.3 Implementation and Comparison

When minimizing E either with a Level Sets Method approach [24,25] or with a Graph Cuts one [9], one should be aware of the dependency of the PDFs upon γ. This leads to an iterated process that is usual in the Level Sets gradient descent, but is not in the case of Graph Cuts. As we do not need sub-pixel accuracy, we opt for a Graph Cuts approach, mainly for speed reasons. Using EM instead of K-means is theoretically important: the algorithm consists in alternately updating the PDFs according to the segmentation and in segmenting according to the PDFs. At each step, the energy decreases:

Table 1. Weights associated to node p

link	weight	for
$t_{s,p}$	0	$p \in \Omega_U$
$t_{p,t}$	∞	$p \in \Omega_U$
$t_{s,p}$	$D_B(p)$	$p \notin \Omega_U$
$t_{p,t}$	$D_F(p)$	$p \notin \Omega_U$
$t_{p,q}$	$\mathcal{V}(p,q)$	$q \in \mathcal{N}(p)$

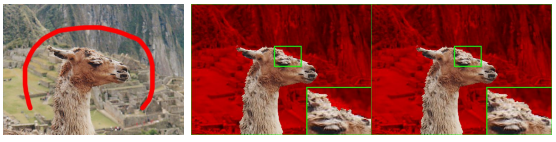


Fig. 2. Importance of then EM estimation and reliability of the MDL criterion. Original image with background specification in red (left) and the corresponding segmentations using the method in [7] with fixed $N_F = N_B = 5$ (middle) and our EM/MDL approach (right)

- Updating the PDF using EM ensures that E_{data} decreases, E_{smooth} being fixed.
- The Graph Cuts step ensures that E decreases.

Let us just recall useful notations [26]. We consider a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ that is a set of nodes $\mathcal{V}$ and directed edges $\mathcal{E}$ connecting them. Two special terminal nodes are present: the source s and the sink t. Each edge (p, q) connecting a node p to a node q is assigned a weight $t_{p,q}$. Edges are broken in two groups: n-links and t-links. A n-link is an edge connecting two non-terminal nodes. A t-link connects a non-terminal node to a terminal node. A cut C is a partitioning of the nodes of the graph into two disjoint subsets S et T such that the source $s \in S$ and the sink $t \in T$. Its cost is the sum of the weights of all edges (p, q) such that $p \in S$ and $q \in T$. A minimum cut is a cut with minimal cost and one minimum cut can be determined in polynomial time with a max-flow extraction algorithm.

Here, each pixel of the image is associated to a node and to edges for each of its neighbors. Each node is also connected to the sink and the source. The weights on the t-links deal with data constrain and those on the n-links account for smoothness. For a pixel x associated to node p, let D_X be the negative logarithm of the probability density function associated to region Ω_X: $D_X(p) = -\log\ p_X\left(I\left(x\right)\right)$. The Graph is built according to table 3.3. After the cut, the nodes that are still connected to the source, are assigned to Ω_F, the others to Ω_B. Figure 2 shows the result of this segmentation process on some test image using both the method in [7] and a method using an EM/MDL estimation. Note that some details misclassified by the original method are correctly handled by the EM/MDL approach. Yet, these improvements are not decisive. More important is the fact that the MDL based estimation of N_X proves to be reliable and masks one annoying parameter from the user.

4 Trimap Segmentation

With these notations in hand, let us go back to our main goal of segmenting a trimap. Assuming that the blended region will also be modeled by a PDF $p_M(I)$, still to be modeled, the data driven part of the energy is unchanged and given by equation (2) with a new partition function that reflects the 3 regions.

However, keeping the same smoothing term is a nonsense. A high image gradient does not indicate a frontier between two regions anymore. Instead, we use the length of the frontiers separating the regions as a smoothing energy. Classical in the Level Set framework and inducing mean curvature motion, this can also be handled rigorously in an Markov Random Field framework (see [27]). Here, we will restrict ourselves to an approximation, just replacing the previous $\mathcal{V}^0(x,y)$ to a decreasing function of the distance between x and y (e.g. $\mathcal{V}^0(x,y) = \kappa/(1+d(x,y))$).

Keeping the same GMM models for p_B and p_F, we still have to design a model for p_M in order to define the energy to minimize.

4.1 A PDF for Ω_M

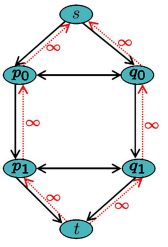
A straightforward solution would be to take a third GMM for p_M and to estimate its parameters $\Theta_M = \{N_M, (\pi_i^M, \mu_i^M, \Sigma_i^M)_{i=1..N_M}\}$ via the same EM/MDL scheme as for p_B and p_F. It would be a mistake. Indeed, p_M is not independent from p_B and p_F: in Ω_M, I, I_F and I_B are related by equation (1). Despite this, one could willingly ignore this dependency and try to segment $(\Omega_B, \Omega_M, \Omega_F)$ as three regions with each one its own independent GMM. Unfortunately, it is not obvious that the resulting iterated minimizing process will converge to the desired regions without a very accurate initialization, specifying pixels of the three regions. On the contrary, making p_M depend on p_B and p_F will turn out to be sufficient to keep a coarse initialization Ω_U.

Let us examine equation (1). We will assume for simplicity that both I_F and I_B come from one single Gaussian of the respective GMMs p_F and p_B. In their Bayesian estimation of layers from multiple images, Wexler et al. [4] assume that α follows a Beta law. Yet, they choose the parameters of the Beta distribution by estimating them on some reference image. Thus, although Kitamoto gives in [28,29] a Gaussian approximation of a mixture of two Gaussian distributions when the mixture coefficient follows a Beta Law, we prefer to simply consider that α follows a uniform law. In that case, if I_F comes from $G_{\mu_i^F, \Sigma_i^F}$ and I_B from $G_{\mu_j^B, \Sigma_j^B}$, the distribution of I can be approximated by another Gaussian $G_{\mu_{ij}^M, \Sigma_{ij}^M}$, given also by Kitamoto in [28,29] as:

$$\mu_{ij}^M = \frac{\mu_i^F + \mu_j^B}{2} \quad \text{and} \quad \Sigma_{ij}^M = \frac{1}{3}\left(\Sigma_i^F + \Sigma_j^B\right) + \frac{1}{12}\left(\mu_j^B - \mu_i^F\right)\left(\mu_j^B - \mu_i^F\right)^T \quad (5)$$

Note that this is, again, an approximation and that more sophisticated models could be investigated. Actually, our simple assumption of a uniform α, and of a Gaussian approximation for I, will turn out to give good results. With this choice, it is natural to model p_M with another GMM, whose $N_M = N_F N_B$ components are now fixed and dependent on p_F and p_B:

$$p_M(I) = \sum_{i=1}^{N_F} \sum_{j=1}^{N_B} \pi_{ij}^M G_{\mu_{ij}^M, \Sigma_{ij}^M}(I) \quad (6)$$



link	weight	for
t_{s,p_0}	∞	$p \in \Omega_U$
t_{p_0,p_1}	∞	$p \in \Omega_U$
$t_{p_1,t}$	0	$p \in \Omega_U$
t_{s,p_0}	$D_F(p)$	$p \notin \Omega_U$
t_{p_0,p_1}	$D_M(p)$	$p \notin \Omega_U$
$t_{p_1,t}$	$D_B(p)$	$p \notin \Omega_U$
t_{p_i,q_i}	$\mathcal{V}(p,q)$	$q \in \mathcal{N}(p)$

Fig. 3. Trimap segmentation. Graph representation for two nodes p and q and associated weights.

where $\sum_{ij} \pi_{ij}^M = 1$ and where the $(\mu_{ij}^M, \Sigma_{ij}^M)$ are given by equation (5). The only free parameters are the (π_{ij}^M), and we estimate them with an EM algorithm on Ω_M.

4.2 Graph Cuts Implementation

As we assume that the blended region Ω_M separates Ω_F from Ω_B, we can use the Graph Cuts implementation described in [30] which is simpler than the usual α-expansion based algorithm and provides a global minimum. Each pixel x is represented by two nodes p_0 and p_1. The graph is built according to figure 3. After the cut, each node is labeled according to the following rule:

- If the link between $\{s, p_0\}$ is cut, the node is assigned to the foreground.
- If the link between $\{p_0, p_1\}$ is cut, the node is assigned to the blended region.
- If the link between $\{p_1, t\}$ is cut, the node is assigned to the background.

Here we use the method described by Kolmogorov and Zabih in [10] to force the algorithm to cut one and only one of the three links $\{s, p_0\}$, $\{p_0, p_1\}$ and $\{p_1, t\}$. It consists in adding infinite reverse edges on the graph (see red links on figure 3). Like in the two regions case, we use an iterative scheme. However we found that using the two regions segmentation as a first step gives good initial estimates for p_B and p_F and speeds up the convergence.

4.3 Results

Figure 4 shows the trimap obtained for the reference image in [3] from just a coarse indication of the background. It is similar to the hand designed one used in the original work. For comparison purposes, we show also the trimap obtained when naively modeling the blended region with an independent GMM, even when starting from a more accurate initialization. Figure 6**a** and 6**b** in next section show many other automatic trimaps. Table 4.3 gives the running times of the trimap extraction (and of the first step of two regions segmentation) for some of our test images, on a standard 2.4GHz PC without any specific optimization. These are the times for a complete convergence and the process might be stopped

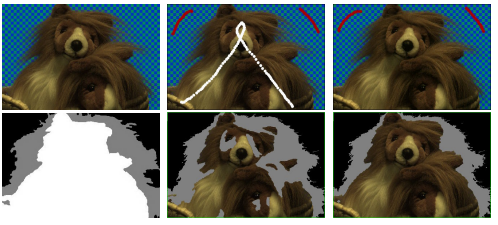


Table 2. Running times for trimap segmentation on some test images

Image	First step	Total time
Teddy Bear	36s	94s
Butterfly	14s	28s
Light	48s	133s

Fig. 4. Automatic trimaps. First column: the original image and the hand designed trimap used in [3]. Second column: background/foreground initialization (in red/white) and the obtained trimap, naively considering p_M as an independent GMM. Third column: background only initialization (in red) and the trimap obtained with our method

before. A multi-scale approach would also improve speed significantly. Anyway, these are to be compared with the times needed for a cautious manual segmentation, depending on the user's ability and/or equipment. Note that the more complex a manual segmentation would be, the more the automatic segmentation seems to require time to converge (see images on figures 6**a** and 6**b**).

5 An Improved Matting Method

In this section, we propose a new matting algorithm taking advantage of our blended region model. Based on global statistics rather than on local ones, it is faster than the original Bayesian matting, although without quality loss.

Chuang et al.'s Bayesian matting algorithm is based on estimating local statistics of the foreground and of the background. For each pixel in the blended region, a neighborhood is considered, where the foreground and the background are respectively modeled by two Gaussian distributions $G_{\mu^F_{loc},\Sigma^F_{loc}}$ and $G_{\mu^B_{loc},\Sigma^B_{loc}}$.

Estimating a local mean and covariance for each pixel is inefficient from a computational point of view. Moreover, limiting the distribution of the neighborhood of a pixel to one single Gaussian may sometimes be a too coarse approximation. We propose to take advantage of our global GMM analysis of the foreground and the background carried out during the segmentation process. Keeping the assumption that I_F and I_B come from one Gaussian each, we choose these two Gaussian distributions respectively among the components of p_F and p_B. We use $\pi^M_{ij} G_{\mu^M_{ij},\Sigma^M_{ij}}(I)$ to measure which Gaussian distributions most probably explain I. Thus, we simply:

1. choose the pair (i_0, j_0) that maximizes $\pi^M_{ij} G_{\mu^M_{ij},\Sigma^M_{ij}}(I)$
2. use Chuang et al.'s solving scheme with $G_{\mu^F_{i_0},\Sigma^F_{i_0}}$ and $G_{\mu^B_{j_0},\Sigma^B_{j_0}}$ as priors for I_F and I_B instead of the local estimations $G_{\mu^F_{loc},\Sigma^F_{loc}}$ and $G_{\mu^B_{loc},\Sigma^B_{loc}}$.

The resulting process turns out to be faster than the original method and the results are similar. Note that it is essential that the GMMs have enough com-

Table 3. Running times for the standard Bayesian matting and for our method on some test images

Image	Bayesian matting	Our matting
Teddy Bear	47s	0.36s
Butterfly	2.7s	0.027s
Light	37s	0.27s

Fig. 5. From top to bottom, left to right: three alpha masks (ground truth, Bayesian matting using our trimap, our method using our trimap), a recompositing using our mask and foreground estimations

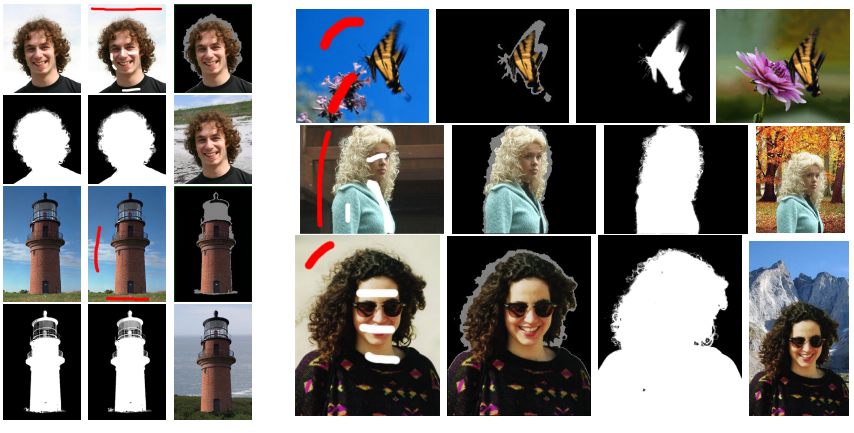


Fig. 6. a. For each image, in reading order: original image, user's initialization, automatic trimap, Bayesian matting, our matting, recompositing. **b.** On each line, from left to right: user's initialization, automatic trimap, our matting, recompositing.

ponents to explain all the colors/textures locally present in the image. Our EM/MDL estimation ensures this.

6 Results

In their original work on Bayesian matting, Chuang et al. proposed a real bench image, supplying a ground truth for the alpha mask (see [3]). Figure 5 shows this *true* mask, compared those obtained with their and our matting algorithm, using our automatic trimap in both cases. The result are similar and the relative errors, in L^2 norm in region Ω_M, respectively gives 1.5% and 1.4% errors.

As expected, the main advantage of our method is its computational efficiency. Table 5 gives the running times of both methods for some of our test images, under the same conditions as previously (standard 2.4GHz PC, no specific optimization). We observed a speedup of about 100. Please note that this

would also stand when starting from a manual trimap. The only overhead for our matting would be to estimate the global statistics from this trimap before running, which is actually negligible with respect to the matting process.

Finally, figures 6**a** and 6**b** show the complete process of our method on several test images: the original images (figure 6**a** only), the user's initialization, the segmented trimap, the mask obtained with Bayesian matting (fig. 6**a** only), the one obtained with our method, and a recompositing from our (α, I_F, I_B) estimation. It demonstrates how a simple initialization without any additional parameter (e.g. number of Gaussian distributions) is enough to get accurate trimaps, and how our fast matting method gives results similar to the ones obtained with the original but slower Bayesian matting.

7 Conclusion

In this paper, we propose a segmentation method aimed at extracting an accurate trimap for the digital matting problem. A statistical model is specifically designed for the blended region and an iterative Graph Cut based optimization scheme allows this trimap segmentation from just a coarse specification of some background and/or foreground pixels. This trimap is similar with those obtained by a meticulous hand drawing. Finally, taking advantage of this blended region model, we describe a improved digital matting method, based on global statistics, much faster than the original Bayesian matting, although without quality loss. This method is also usable starting from a manual trimap.

References

1. Smith, A.R., Blinn, J.F.: Blue screen matting. In: 23rd annual conference on Computer graphics and interactive techniques, ACM Press (1996) 259–268
2. Ruzon, M., Tomasi, C.: Alpha estimation in natural images. In: CVPR. (2000)
3. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: CVPR. Volume 2. (2001) 264–271
4. Wexler, Y., Fitzgibbon, A., Zisserman, A.: Bayesian estimation of layers from multiple images. In: ECCV. (2002)
5. Perez, P., Gangnet, M., Blake, A.: Poisson image editing. In: SIGGRAPH. (2003)
6. Sun, J., Jia, J., Tang, C., shum, H.: Poisson matting. In: SIGGRAPH. (2004)
7. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive gmmrf model. In: ECCV. (2004)
8. Rother, C., Kolmogorov, V., A., B.: Grabcut - interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004)
9. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV. (2001) 105–112
10. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. on Pattern Analysis and Machine Intelligence (2004) 65–81
11. Mitsunaga, T., Yokoyama, T., Totsuka, T.: Autokey: Human assisted key extraction. In: SIGGRAPH. (1995)
12. Xiao, J., Shah, M.: Accurate motion layer segmentation and matting. In: CVPR. (2005)

13. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on the Hamilton–Jacobi formulation. Journal of Computational Physics **79** (1988) 12–49
14. Osher, S., Paragios, N., eds.: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer Verlag (2003)
15. Brox, T., Rousson, M., Deriche, R., Weickert, J.: Unsupervised segmentation incorporating colour, texture, and motion. In: 10th International Computer Analysis of Images and Patterns. LNCS 2756, Springer Verlag (2003) 353–360
16. Kadir, T., Brady, M.: Unsupervised non-parametric region segmentation using level sets. In: Proceedings of ICCV 2003. (2003)
17. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In Cam, L.M.L., Neyman, J., eds.: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1. (1967) 281–297
18. X. Descombes, M. Sigelle, F.P.: Estimating gaussian markov random field parameters in a nonstationary framework: Application to remote sensing imaging. IEEE Trans. on Image Processing **8** (1999) 490–503
19. Orchard, M.T., Bouman, C.A.: Color Quantization of Images. IEEE Trans. on Signal Processing **39** (1991) 2677–2690
20. McLachlan, G., Krishnan, T.: The EM algorithm and extensions. Wiley, New York (1997)
21. McLachlan, G., David, P.: Finite Mixture Models. Wiley, New York (2000)
22. Figueiredo, M., Leitao, J.M.N., Jain, A.K.: On fitting mixture models. In: Energy Minimization Methods in Computer Vision and Pattern Recognition. (1999) 54–69
23. Ueda, N., Nakano, R., Ghahramani, Z., Hinton, G.E.: Smem algorithm for mixture models. Neural Computation **12** (2000) 2109–2128
24. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based space. In: International Conference on Computer Vision and Pattern Recognition. Volume 2., Madison, Wisconsin, USA (2003) 699–704
25. Juan, O., Keriven, R., Postelnicu, G.: Stochastic Motion and the Level Set Method in Computer Vision: Stochastics Active Contours. International Journal of Computer Vision (in press)
26. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence **23** (2001) 1222–1239
27. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: ICCV. (2003)
28. Kitamoto, A.: The moments of the mixel distribution and its application to statistical image classification. In Amin, A., Ferri, F., Inesta, J., Pudil, P., eds.: Advances in Pattern Recognition (SPR'00), LNCS 1876. (2000) 521–531
29. Kitamoto, A., Takagi, M.: Area proportion distribution – relationship with the internal structure of mixels and its application to image classification. Systems and Computers in Japan **31** (2000) 57–76
30. Ishikawa, H.: Exact optimization for markov random fields with convex priors. IEEE Trans. on Pattern Analysis and Machine Intelligence **25** (2003) 1333–1336

Uncertainty-Driven Non-parametric Knowledge-Based Segmentation: The Corpus Callosum Case

Maxime Taron[1], Nikos Paragios[1], and Marie-Pierre Jolly[2]

[1] CERTIS - Ecole Nationale des Ponts et Chaussees,
6 - 8 Avenue Blaise Pascal, 77455 Champs-sur-Marne, France
{taron, paragios}@certis.enpc.fr
[2] Imaging & Visualization Department, Siemens Corporate Research,
755 College Road East, Princeton, NJ 08540, USA
marie-pierre.jolly@siemens.com

Abstract. In this paper we propose a novel variational technique for the knowledge based segmentation of two dimensional objects. One of the elements of our approach is the use of higher order implicit polynomials to represent shapes. The most important contribution is the estimation of uncertainties on the registered shapes, which can be used with a variable bandwidth kernel-based non-parametric density estimation process to model prior knowledge about the object of interest. Such a non-linear model with uncertainty measures is integrated with an adaptive visual-driven data term that aims to separate the object of interest from the background. Promising results obtained for the segmentation of the corpus callosum in MR mid-sagittal brain slices demonstrate the potential of such a framework.

1 Introduction

Over the last decade, shape-based segmentation methods have become more and more common. First introduced in 1995, active shape models (ASM) and active appearance models (AAM) [3] have been very popular tools for the segmentation of anatomical structures in medical images [2,5,1,10]. More recently, principal component analysis (PCA) has also been applied to distance transforms for an implicit representation of shapes [9]. Shape-based segmentation is usually equivalent to recovering a geometric structure which is both highly probable in the model space and well aligned with strong features in the image. The advantage of the shape based methods over classical deformable templates [11] is that they allow the deformation process to be constrained to remain within the space of allowable shapes. These methods have proven to be a good compromise between complexity and shape generalization. However, since modeling is performed after registration, errors in the registration can be propagated into the model space. Furthermore, the assumption of Gaussian shape models might be a little restrictive.

In this paper, shapes are represented implicitly using the distance transform. To generate a model of the structure of interest, we register shape examples using a spline based free form deformation. The main contribution of this paper is the derivation of

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 198–209, 2005.

a measure representing the uncertainty of the registration at the zero iso-surface. After dimensionality reduction, these measures are combined with a variable bandwidth kernel-based approach to derive a density function that models the family of shapes under consideration. Given a new image, the segmentation process is expressed in a variational level set framework [14] where the energy function makes use of the uncertainties of the registration between the deformed shape which aligns to the image features and the model.

We apply our novel modeling and segmentation technique to the case of the corpus callosum. The corpus callosum is a thick bundle of nerve fibers that connect the left and right hemispheres in the brain. It is believed to be responsible for balancing the load of learning tasks across each hemisphere, making each specialized in certain tasks. While not learning, it is responsible for routing most of the communication between the two hemispheres. This is the reason why a surgical procedure has been developed to cut the corpus callosum in patients with severe epilepsy for which drug treatment is ineffective. In addition, several studies indicate that the size and shape of the corpus callosum is related to various types of brain dysfunction such as dyslexia [4] or schizophrenia [6]. Therefore, neurologists are interested in looking at the corpus callosum and analyzing its shape. Magnetic resonance imaging (MRI) is a safe and non-invasive tool to image the corpus callosum. Since manual delineation can be very time consuming, we demonstrate how our algorithm can be used to segment the corpus callosum on mid-sagittal MR slices.

The remainder of this paper is organized as follows. In Section 2, we introduce registration with uncertainties and probabilistic modeling to describe the corpus callosum structure. The segmentation component combining data and shape terms is described in Section 3. Experimental results are presented in Section 4. Finally, conclusions and future directions are discussed in Section 5.

2 Shape Representation Through Implicit Polynomials

Let us consider a training set $\{C_1, C_2, ..., C_{\mathrm{N}}\}$ of shapes representing the structure of interest. The model building task consists of recovering a probabilistic representation of this set. In order to remove all the pose variation from the training set, all shapes have to be registered to a common pose with respect to an affine transformation. Then a reference model $C_{\mathcal{M}}$ is locally registered to every sample of the training set C_i using implicit polynomials. We will first describe the registration process and the calculation of uncertainties on the registered model. The uncertainty measures represent the allowable range of variations in the deformations of the model that still match C_i. Then we describe the way these uncertainties are used in the estimation of probability density function of the deformations.

2.1 Registration Through Implicit Polynomials

In the classical ASM the initial step is used to recover explicit correspondence between the discretized contour of the model shape and the training examples. In the present

framework, the model shape is non rigidly registered to every sample from the training, and the statistical shape model is actually built on the parameters of the recovered transformation.

Shapes C_i are represented in an implicit fashion using the Euclidean distance transform [9,15] . In the 2D case, we consider the function defined on the image domain Ω :

$$\phi_{C_i}(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in C_i \\ +\mathcal{D}(\mathbf{x}, C_i), & \mathbf{x} \in \mathcal{R}_{C_i} \\ -\mathcal{D}(\mathbf{x}, C_i), & \mathbf{x} \notin \mathcal{R}_{C_i} \end{cases}$$

where $\mathcal{R}_{C_i}$ is the region enclosed by C_i. Such a space is invariant to translation, rotation and can also be modified to account for scale variations. This representation has already been used along with simple criteria like sum of squared differences to address similarity registration [15] or mutual information for affine transformations [7].

The retained framework for density estimation does not put any constraint on the reference model used for registration. In practice we choose a shape characteristic of the object to segment. Without loss of generality, we can choose for $C_{\mathcal{M}}$ a smoothed version of C_1. All contours of the training set are now registered to $C_{\mathcal{M}}$ with respect to an affine transform and from now on, we will denote $\{C_1, C_2, ..., C_{\mathrm{N}}\}$ as the globally registered training set.

Local registration is crucial to model building. To this end one would like to recover an invertible transformation (diffeomorphism) $\mathcal{L}_{\mathbf{\Theta}_i}$ parameterized by a vector $\mathbf{\Theta}_i$ that creates a one to one mapping between each contour of the training set C_i and the model $C_{\mathcal{M}}$:

$$\mathcal{L}_{\mathbf{\Theta}_i} : \mathcal{R}^2 \rightarrow \mathcal{R}^2 \text{ and } \mathcal{L}_{\mathbf{\Theta}_{\mathrm{i}}}(\mathrm{C}_{\mathcal{M}}) \approx \mathrm{C_i}$$

When $\mathcal{L}_{\mathbf{\Theta}}$ is chosen as a 2D polynomial with coefficients $\mathbf{\Theta}$ in an appropriate basis, the expression $\phi \circ \mathcal{L}_{\mathbf{\Theta}}$ inherits the invariance properties of implicit polynomials, i.e. linear transformations applied to $\mathbf{\Theta}$ are related to linear transformations applied to the data space. In the present paper, we used a simple polynomial warping technique to address the demand of local registration: the free form deformations method (FFD) [16]. The essence of FFD is to deform an object by manipulating a regular control lattice overlaid on its embedding space. We use a cubic B-spline FFD to model the local transformation $\mathcal{L}$. Consider the $M \times N$ square lattice of points, $[\{\mathbf{P}^0_{m,n}\}; (m,n) \in [1;M] \times [1;N]]$. In this case the vector of parameters $\mathbf{\Theta}$ defining the transformation $\mathcal{L}$ is the displacement coordinates of the control lattice. $\mathbf{\Theta}$ has size $2MN$:

$$\mathbf{\Theta} = \{\delta\mathbf{P}^x_{m,n}, \delta\mathbf{P}^y_{m,n}\}; (m,n) \in [1;M] \times [1;N]$$

The motion of a pixel $\mathbf{x}$ given the deformation of the control lattice, is defined in terms of a tensor product of Cubic B-splines [17]. As FFD is linear in the parameter $\mathbf{\Theta} = \delta\mathbf{P}$, it can be expressed in a compact form by introducing $\mathcal{X}(\mathbf{x})$ a $[2 \times 2MN]$ matrix:

$$\mathcal{L}(\mathbf{\Theta}; \mathbf{x}) = \sum\sum B_i(u)B_j(v)(\mathbf{P}^0_{i,j} + \delta\mathbf{P}_{i,j}) = \mathbf{x} + \mathcal{X}(\mathbf{x})\mathbf{\Theta}$$

where (u, v) are the coordinates of $\mathbf{x}$, and (B_i, B_j) the cubic B-spline basis functions.

Local registration now is equivalent to finding the best lattice configuration such that the overlaid structures coincide. Since structures correspond to distance transforms

of globally aligned shapes, the sum of squared differences (SSD) can be considered as the data-driven term to recover the deformation field $\mathcal{L}(\mathbf{\Theta};\mathbf{x})$ between the element C_i of the training set and the model $C_{\mathcal{M}}$ (corresponding respectively to the distance transform ϕ_i and $\phi_{\mathcal{M}}$)

$$E_{\text{data}}(\mathbf{\Theta}) = \iint_{\Omega} \chi_\alpha(\phi_i(\mathbf{x}))\left[\phi_i(\mathcal{L}(\mathbf{\Theta};\mathbf{x})) - \phi_{\mathcal{M}}(\mathbf{x})\right]^2 d\mathbf{x} \tag{1}$$

with $\chi_\alpha(\phi_i(\mathbf{x}))$ being an indicator function that defines a band of width α around the contour. In order to further preserve the regularity of the recovered registration, one can consider an additional smoothness term on the deformation field $\delta\mathcal{L}$. We consider a computationally efficient smoothness term :

$$E_{\text{smooth}}(\mathbf{\Theta}) = \iint_{\Omega} \left(\left|\mathcal{L}_{xx}(\mathbf{\Theta};\mathbf{x})\right|^2 + 2\left|\mathcal{L}_{xy}(\mathbf{\Theta};\mathbf{x})\right|^2 + \left|\mathcal{L}_{yy}(\mathbf{\Theta};\mathbf{x})\right|^2\right) d\mathbf{x}.$$

The data-driven term and the smoothness constraint component can now be integrated to recover the local deformation component through the calculus of variations. We denote as $\mathbf{\Theta}_i$ the reached minimum.

However, one can claim that the local deformation field is not sufficient to characterize the registration between two shapes. Data is often corrupted by noise so that the registration retrieved using a deformable model may be imprecise. Therefore, recovering uncertainty measurements [8] that do allow the characterization of an allowable range of variation for the registration process is an eminent condition of accurate shape modeling.

2.2 Uncertainty Estimation on Registered Shapes

We aim to recover uncertainties on the vector $\mathbf{\Theta}$ in the form of a $[2MN \times 2MN]$ covariance matrix by adapting a method initially introduced in [18]. We are considering the quality of the local registration on shapes, that is the zero levelset of the distance transform. Therefore, E_{data} is formulated in the limit case where α the size of the limited band around the model shape tends to 0. The data term of the energy function (1) can now be expressed as:

$$E_{\text{data}}(\mathbf{\Theta}) = \oint_{\mathcal{C}_{\mathcal{M}}} \phi_i^2(\mathcal{L}(\mathbf{\Theta};\mathbf{x}))d\mathbf{x} = \oint_{\mathcal{C}_{\mathcal{M}}} \phi_i^2(\mathbf{x}')d\mathbf{x},$$

where we denote $\mathbf{x}' = \mathcal{L}(\mathbf{\Theta}_i;\mathbf{x})$. Let us consider $\mathbf{q}$ to be the closest point from $\mathbf{x}'$ located on C_i. As ϕ_i is assumed to be a Euclidean distance transform, it also satisfies the condition $\|\nabla\phi_i(\mathbf{x}')\| = 1$. Therefore one can express the values of ϕ_i at the first order in the neighborhood of $\mathbf{x}'$ in the following manner :

$$\begin{aligned}\phi_i(\mathbf{x}' + \delta\mathbf{x}') &= \phi_i(\mathbf{x}') + \delta\mathbf{x}' \cdot \nabla\phi_i(\mathbf{x}') + \circ(\delta\mathbf{x}') \\ &= (\mathbf{x}' + \delta\mathbf{x}' - \mathbf{q}) \cdot \nabla\phi_i(\mathbf{x}') + \circ(\delta\mathbf{x}')\end{aligned}$$

This local expression of ϕ_i with a dot product reflects the condition that a point to curve distance was adopted. Under the assumption that E_{data} is small when reaching the

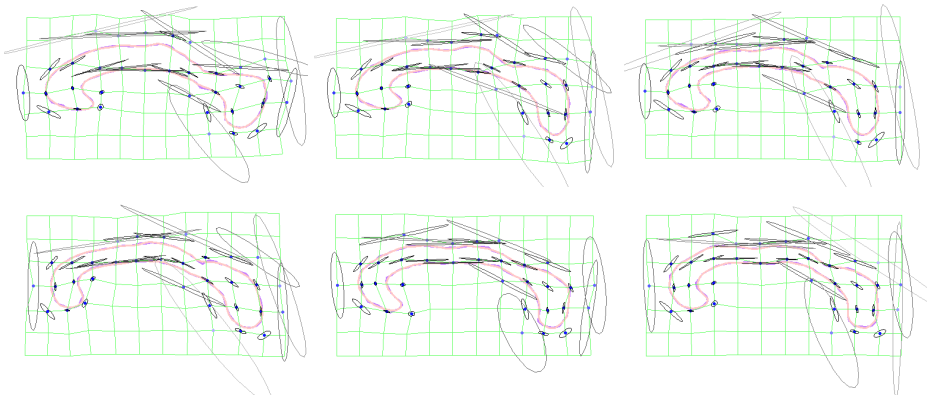


Fig. 1. Implicit higher order polynomials and registration of corpus callosum with uncertainty estimates (this figure should be seen in color)

optimum, we can write the classical second order approximation of quadratic energy in the form:

$$E_{\text{data}}(\mathbf{\Theta}) = \oint_{\mathcal{C}_\mathcal{M}} \left[(\mathbf{x}' - \mathbf{q}) \cdot \nabla\phi_i(\mathbf{x}')\right]^2 = \oint_{\mathcal{C}_\mathcal{M}} \left[(\mathbf{x} + \mathcal{X}(\mathbf{x})\mathbf{\Theta} - \mathbf{q}) \cdot \nabla\phi_i(\mathbf{x}')\right]^2$$

Localizing the global minimum of an objective function E is equivalent to finding the major mode of a random variable with density $\exp(-E/\beta)$. The coefficient β corresponds to the allowable variation in the energy value around the minimum. In the present case of a quadratic energy (and therefore Gaussian random variable), the covariance and the Hessian of the energy are directly related by $\Sigma_{\mathbf{\Theta}_i}^{-1} = H_{\mathbf{\Theta}_i}/\beta$. This leads to the following expression for the covariance :

$$\Sigma_{\mathbf{\Theta}_i}^{-1} = \frac{1}{\beta} \oint_{\mathcal{C}_\mathcal{M}} \mathcal{X}(\mathbf{x})^T . \nabla\phi_i(\mathbf{x}') . \nabla\phi_i(\mathbf{x}')^T . \mathcal{X}(\mathbf{x}) d\mathbf{x}$$

In the most general case one can claim that the matrix $H_{\mathbf{\Theta}}$ is not invertible because the registration problem is under-constrained. Then, additional constraints have to be introduced towards the estimation of the covariance matrix of $\mathbf{\Theta}_i$ through the use of an arbitrarily small positive parameter γ :

$$E(\mathbf{\Theta}) = \oint_{\mathcal{C}_\mathcal{M}} \left[(\mathbf{x} + \mathcal{X}(\mathbf{x})\mathbf{\Theta} - \mathbf{q}) \cdot \nabla\phi_i(\mathbf{x}')\right]^2 d\mathbf{x} + \gamma\, \mathbf{\Theta}^T\mathbf{\Theta}$$

This leads to the covariance matrix for the parameter estimate :

$$\Sigma_{\mathbf{\Theta}_i} = \beta \left(\oint_{\mathcal{C}_\mathcal{M}} \mathcal{X}(\mathbf{x})^T . \nabla\phi_i(\mathbf{x}') \nabla\phi_i(\mathbf{x}')^T \mathcal{X}(\mathbf{x}) d\mathbf{x} + \gamma \mathbf{I} \right)^{-1} \tag{2}$$

2.3 Hybrid Kernel Based Density Function and Kernel Selection

Now that all shapes of the training set have been aligned, standard statistical techniques like PCA or ICA could be applied to recover linear Gaussian models. But in the most general case shapes that refer to objects of particular interest vary non-linearly and therefore the assumption of simple parametric models likes Gaussian is rather unrealistic. Therefore within our approach we propose a non-parametric form of the probability density function.

Let $\{\mathbf{\Theta}_1 ... \mathbf{\Theta}_N\}$ be the N vectors of parameters associated with the registration of the N sample of the training set. Considering that this set of vectors is a random sample drawn from the density function f describing the shapes, the fixed bandwidth kernel density estimator consists of:

$$\hat{f}(\mathbf{\Theta}) = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \frac{1}{\|\mathbf{H}\|^{1/2}} \mathrm{K}\left(\mathbf{H}^{-1/2}(\mathbf{\Theta} - \mathbf{\Theta}_i)\right)$$

where $\mathbf{H}$ is a symmetric definite positive (bandwidth matrix) and K denote the centered Gaussian kernel with identity covariance. Fixed bandwidth approaches often produce under-smoothing in areas with sparse observations and over-smoothing in the opposite case.

Kernels of variable bandwidth can be used to encode such a condition and provide a structured way for utilizing the variable uncertainties associated with the sample points. In the literature, kernel density estimation methods that do rely on varying bandwidths are generally referred to as adaptive kernels. Density estimation is performed with kernels whose bandwidth adapts to the sparseness of the data [19].

In the present case, the vectors $\{\mathbf{\Theta}_i\}$ come along with associated uncertainties $\{\Sigma_i\}$. Furthermore, the point $\mathbf{\Theta}$ where the density function is evaluated corresponds to a deformed model, and therefore is also associated to a measure of uncertainty Σ. In order to account for the uncertainty estimates both on the sample points themselves as well as on the estimation point, we adopt a hybrid estimator [12].

$$\begin{aligned}\hat{f}_H(\mathbf{\Theta}, \Sigma) &= \frac{1}{N} \sum_{i=1}^{N} \mathcal{K}(\mathbf{\Theta}, \Sigma, \mathbf{\Theta}_i, \Sigma_i) \\ &= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\|\mathbf{H}(\Sigma_{\mathbf{\Theta}}, \Sigma_{\mathbf{\Theta}_i})\|^{1/2}} \mathrm{K}(\mathbf{H}(\Sigma_{\mathbf{\Theta}}, \Sigma_{\mathbf{\Theta}_i})^{-1/2}(\mathbf{\Theta} - \mathbf{\Theta}_i)\end{aligned}$$

where we choose for the bandwidth function: $\mathbf{H}(\Sigma_{\mathbf{\Theta}}, \Sigma_{\mathbf{\Theta}_i}) = \Sigma_{\mathbf{\Theta}} + \Sigma_{\mathbf{\Theta}_i}$ as proposed in [12]. Using this estimator, the density decreases more slowly in directions of large uncertainties when compared to the other directions.

This metric can now be used to assess the probability of a new sample being part of the training set and account for the non-parametric form of the observed density. However, the computation is time consuming because it leads to the calculation of large matrix inverses. Since the cost is linear in the number of samples in the training set, there is an eminent need to decrease its cardinality by selecting the most representative kernels.

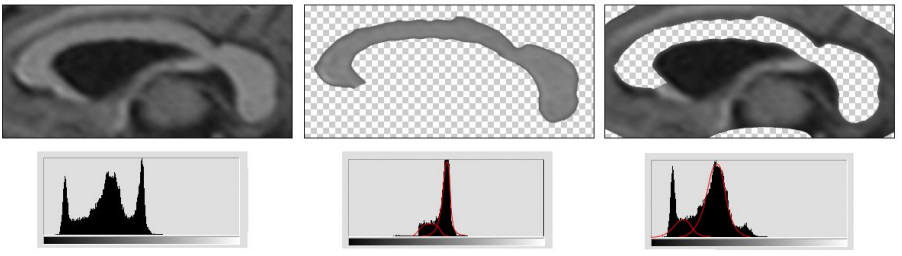


Fig. 2. Histograms of the corpus callosum and the background area. The of a use gaussian mixture to model the corpus callosum and background intensity distribution in MR is appropriate (this figure should be seen in color).

The maximum likelihood criterion expresses the quality of approximation from the model to the data. We use a recursive sub-optimal algorithm to select kernels and therefore build a compact model that maximizes the likelihood of the whole training set.

Consider a set $\mathcal{Z}_{\mathrm{K}} = \{X_1, X_2, \ldots, X_{\mathrm{K}}\}$ of K kernels extracted from the training set with mean and uncertainties estimates $\{X_i = (\mathbf{\Theta}_i \Sigma_i)\}_{i=1}^{K}$. The log likelihood of the entire training set according to this model is:

$$C_{\mathrm{K}} = \sum_{i=1}^{N} \log \left(\frac{1}{K} \sum_{(\mathbf{\Theta}_j, \sigma_j) \in \mathcal{Z}_{\mathrm{K}}} \mathcal{K}(\mathbf{\Theta}_j, \Sigma_i, \mathbf{\Theta}_i, \Sigma_i) \right)$$

A new kernel X_{K+1} is extracted from the training set as the one maximizing the quantity $C_{\mathrm{K}+1}$ associated with $\mathcal{Z}_{\mathrm{K}+1} = \mathcal{Z}_{\mathrm{K}} \bigcup X_{K+1}$. The same kernel may be chosen several times in order to preserve a increasing sequence C_K. Consequently the selected kernels X_i in $\mathcal{Z}_{\mathrm{K}}$ are also associated with a weight factor w_i. Once such a selection has been completed, the hybrid estimator is evaluated over $\mathcal{Z}_{\mathrm{K}}$:

$$\hat{f}_H(\mathbf{\Theta}, \Sigma) = \frac{1}{N} \sum_{(\mathbf{\Theta}_i, \sigma_i, w_i) \in \mathcal{Z}_{\mathrm{K}}} w_i \mathcal{K}(\mathbf{\Theta}, \Sigma, \mathbf{\Theta}_i, \Sigma_i) \tag{3}$$

3 Shape Based Segmentation Applied to the Corpus Callosum

Let us consider an image $\mathcal{I}$ where the corpus callosum structure is present and is to be recovered. Recall that we now have a model of the corpus callosum: a shape that can be transformed using an affine transformation and a FFD, and a measure of how well the deformed shape belongs to the family of trained shapes.

Let $\phi_{\mathcal{M}}$ be the distance transform of the reference model. Segmentation consists of globally and locally deforming $\phi_{\mathcal{M}}$ towards delineating the corpus callosum in $\mathcal{I}$. Let $\mathcal{A}$ be an affine transformation of the model and $\mathcal{L}(\mathbf{\Theta})$ its local deformation using FFD as previously introduced.

For now, we assume that the visual properties of the corpus callosum $\pi_{\mathrm{cor}}()$ as well as the ones of the local surrounding area $\pi_{\mathrm{bck}}()$ are known. Then segmentation of the

corpus callosum is equivalent to the minimization of the following energy with respect to the parameters $\mathbf{\Theta}$ and $\mathcal{A}$:

$$E_{\text{image}}(\mathcal{A}, \mathbf{\Theta}) = -\iint_{\mathcal{R}_{\mathcal{M}}} \log\left[\pi_{\text{cor}}\left(\mathcal{I}\left(\mathcal{A}(\mathcal{L}(\mathbf{\Theta}; \mathbf{x}))\right)\right)\right] d\mathbf{x} - \iint_{\Omega - \mathcal{R}_{\mathcal{M}}} \log\left[\pi_{\text{bkg}}\left(\mathcal{I}\left(\mathcal{A}(\mathcal{L}(\mathbf{\Theta}; \mathbf{x}))\right)\right)\right] d\mathbf{x}$$

where $\mathcal{R}_{\mathcal{M}}$ denotes the inside of $C_{\mathcal{M}}$. However, the direct calculation of variations involves image gradient and often converges to erroneous solutions due to the discretization of the model domain. In that case, we change the integration domain to the image by implicitly introducing the inverse transformation (see Appendix). A bimodal partition in the image space is now to be recovered. The definition of this domain $\mathcal{R}_{\text{cor}}$ depends upon the parameters of the transformation $[\mathcal{A}, \mathbf{\Theta}]$ as :

$$\mathcal{R}_{\text{cor}} = \mathcal{A}(\mathcal{L}(\mathbf{\Theta}, \mathcal{R}_{\mathcal{M}})) \text{ and } \mathbf{y} = \mathcal{A}(\mathcal{L}(\mathbf{\Theta}, \mathbf{x}))$$

The actual image term of the energy to be minimized then becomes:

$$E_{\text{image}}(\mathcal{A}, \mathbf{\Theta}) = -\iint_{\mathcal{R}_{\text{cor}}} \log\left[\pi_{\text{cor}}\left(\mathcal{I}\left(\mathbf{y}\right)\right)\right] d\mathbf{y} - \iint_{\Omega - \mathcal{R}_{\text{cor}}} \log\left[\pi_{\text{bkg}}\left(\mathcal{I}\left(\mathbf{y}\right)\right)\right] d\mathbf{y} \tag{4}$$

where statistical independence is considered at the pixel as well as hypotheses level. In practice the distributions of the corpus callosum as well as the ones of the surrounding region $[\pi_{\text{cor}}, \pi_{\text{bkg}}]$ can be recovered in an incremental fashion using the Mumford-Shah principle [13]. In the present case, each distribution is estimated by fitting a mixture of Gaussians to the image histogram using an Expectation-Maximization algorithm (Fig. 2).

The shape based energy term, making use of the non parametric framework introduced earlier is also locally influenced by a covariance matrix of uncertainty calculated on the transformed model. This covariance matrix is computed in a fashion similar to (2) with the difference that it may only account for the linear structure of the transformed model and therefore allow variations of $\mathbf{\Theta}$ that creates tangential displacements of the contour:

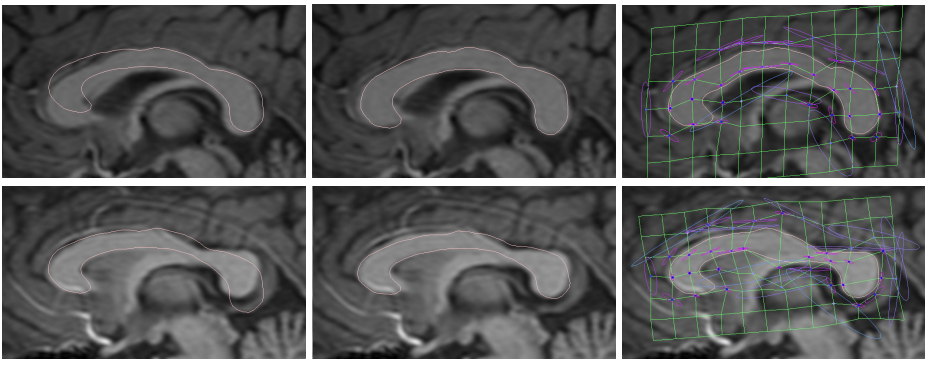
$$\Sigma_{\mathbf{\Theta}}^{-1} = \frac{1}{\beta} \oint_{\mathcal{C}_{\mathcal{M}}} \mathcal{X}(\mathbf{x})^T \nabla\tilde{\phi}_{\mathcal{M}}(\mathbf{x}') \nabla\tilde{\phi}_{\mathcal{M}}(\mathbf{x}')^T \mathcal{X}(\mathbf{x}) d\mathbf{x}$$

where $\tilde{\phi}_{\mathcal{M}}$ is the transformation of $\phi_{\mathcal{M}}$ under the deformation $\mathcal{A}(\mathcal{L}(\mathbf{\Theta}))$. Direct computation leads to:

$$\nabla\tilde{\phi}_{\mathcal{M}}(\mathbf{x}') = \text{com}\left[\frac{d}{d\mathbf{x}}(\mathcal{L}(\mathbf{\Theta}, \mathbf{x}))\right] . \nabla\phi_{\mathcal{M}}(\mathbf{x})$$

where 'com' denotes the matrix of cofactors. Then we introduce the shape based energy term using the same notations as in (3) as:

$$E_{\text{shape}}(\mathbf{\Theta}, \Sigma_{\mathbf{\Theta}}) = -\log(\hat{f}_H(\mathbf{\Theta}, \Sigma))$$



(a) (b) (c)

Fig. 3. Segmentation with uncertainties estimates of the corpus callosum; (a) Automatic rough positioning of the model, (b) segmentation through affine transformation of the model (c) segmentation using the local deformation of the FFD grid and uncertainties estimates on the registration/segmentation process (this figure should be seen in color)

The global energy is minimized with respect to the parameters of $\mathcal{A}$ and $\mathbf{\Theta}$ through the computation of variations on $E = E_{\text{image}} + E_{\text{shape}}$ and implemented using a standard gradient descent.

4 Experimental Results

We have applied our method to the segmentation of the corpus callosum in MR mid-sagittal brain slices.

The first step was to build a model of the corpus callosum. Minimization of the registration energy is performed using gradient descent. In parallel, we successively refine the size of the band α around the contour (from .3 to .05 times the size of the shape), while we increase the complexity of the diffeomorphism (from an affine transformation to an FFD with a regular $[7 \times 12]$ lattice).

Fig. 1 shows examples of FFD deformations along with uncertainty ellipses. These ellipses are the representation of the 2D conic obtained when projecting the covariance matrix $\Sigma_{\mathbf{\Theta}}$ (of size 168×168) on the control points. It therefore does not allow us to represent the correlations between control points.

The segmentation process is initialized by positioning the initial contour according to the method proposed in [10]. Energy minimization is performed through gradient descent, while the PDF π_{cor} and π_{bkg} are estimated by mixtures of Gaussians. Fig. 2 shows the histogram of a typical image of the corpus callosum. The figure illustrates how well mixtures of two Gaussian distributions can represent the individual histograms for the corpus callosum and the background, respectively. Segmentation results are presented in (Fig. 3 and Fig. 4) along with the associated uncertainties. In Fig. 3, we demonstrate the individual steps of the segmentation process: the left most image shows the automatic initialization of the contour, the middle image shows the contour after the affine

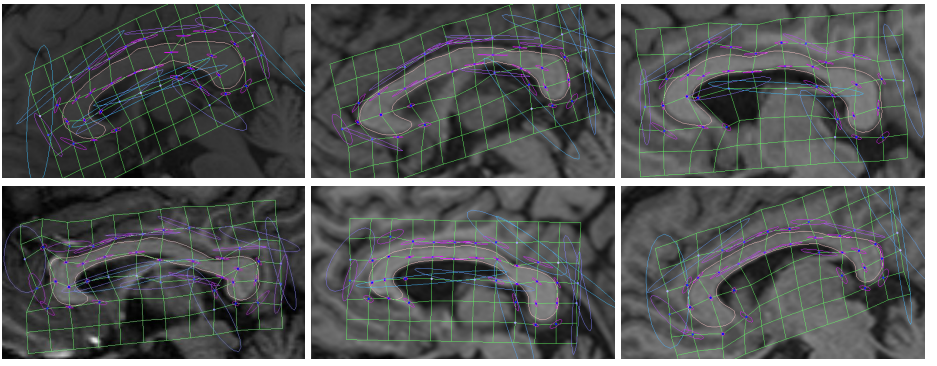


Fig. 4. Additional segmentation results with uncertainty measures

transformation has been recovered, and the right image shows the local deformations. Fig. 4 shows additional results and illustrates that our method can handle a wide variety of shapes for the corpus callosum as well as large variations in image contrast. It can be seen that the results in the bottom left image is not perfect. In general, failures may be due to the fact that the shape constraint is not strong enough and the contrast in the image dominates the deformation. Also, it might be that the shape of this particular corpus callosum cannot be captured with the current PDF because it has been reduced to only 10 kernels.

5 Conclusions

In this paper we have introduced a novel method to account for prior knowledge in the segmentation process using non-parametric variable bandwidth kernels that are able to account for errors in the registration and the segmentation process. We have shown that the method can generate a very good model of the object of interest and produce very good segmentation results.

However the method of kernel selection presented in Section 3 has shown some limitation in practice. Therefore there is a strong need to build more efficient and compact estimators of the shape variation PDF which account for these uncertainty measures. It is also important to note that this method can be extended to higher dimensions. Building models in 3D and segmenting objects of large variability is the next step of our research work.

The covariance matrices of uncertainty Σ_{Θ} are very sparse. Indeed, while using regular FFD, the influence of every grid point is local and therefore many cross correlation coefficients are null. Different types of B-spline deformations using an irregular positioning of control points (but dependent on the model) will be tried to address this issue and therefore reduce the dimensionality of the problem.

Last, but not least, introduction of uncertainties directly measured in the image as part of the segmentation process will provide local measures of confidence and could be considered as a major breakthrough in the area of knowledge-based object extraction.

References

1. J. G. Bosch, S. C. Mitchell, B.P.F. Lelieveldt, F. Nikland, O. Kamp, M. Sonka, and J. H. Reiber. Automatic segmentation of echocardiographic sequences by active appearance motion models. *IEEE Trans. Medical Imaging*, 21(11):1374–1383, 2002.
2. T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):255–266, 1994.
3. T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. *Technical Report, University of Manchester*, 2004.
4. J. Duncan, A. Wang, A. Amini, R. Greene, L. Kier, J. Gore, J. Holahan, S. Shaywitz, J. Fletcher, R. Bronen, and B. Shaywitz. A MRI-based study of the corpus callosum in dyslexic and normal children. *Neurology*, 1996.
5. N. Duta and M. Sonka. Segmentation and interpretation of MR brain images: An improved active shape model. *IEEE Trans. Medical Imaging*, 17(6):1049–1062, 1998.
6. M. Frumin, P. Golland, R. Kikinis, Y. Hirayasu, D. F. Salisbury, J. Hennen, C. C. Dickey, M. Anderson, F. A. Jolesz, W. E. L. Grimson, R. W. McCarley, and M. R. Shenton. Shape differences in the corpus callosum in first-episode schizophrenia and first-episode psychotic affective disorder. *American Journal of Psychiatry*, 159:866–868, 2002.
7. X. Huang, N. Paragios, and D. Metaxas. Registration of Structures in Arbitrary Dimensions: Implicit Representations, Mutual Information & Free-Form Deformations. Technical Report DCS-TR-0520, Division of Computer & Information Science, Rutgers University, 2003.
8. K. Kanatani. Uncertainty modeling and model selection for geometric inference. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10):1307–1319, 2004.
9. M. Leventon, E. Grimson, and O. Faugeras. Statistical Shape Influence in Geodesic Active Controus. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages I:316–322, 2000.
10. A. Lundervold, N. Duta, T. Taxt, and A. Jain. Model-guided segmentation of corpus callosum in MR images. In *CVPR*, pages 1231–1238, 1999.
11. T. McInerney, G. Hamarneh, M. Shenton, and D. Terzopoulos. Deformable organisms for automatic medical image analysis. *Medical Image Analysis*, 6:251–266, 2002.
12. A. Mittal and N. Paragios. Motion-based background substraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition*, volume 2, pages 302–309, 2004.
13. D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 22–26, 1985.
14. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
15. N. Paragios, M. Rousson, and V. Ramesh. Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration. In *European Conference on Computer Vision*, pages II:775–790, 2002.
16. D. Rueckert, L.I. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999.
17. T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *Proceedings SIGGRAPH '86*, 20:151–160, 1986.
18. C. Stewart, C.-L. Tsai, and B. Roysam. The dual bootstrap iterative closest point algorithm with application to retinal image registration. *IEEE Trans. Med. Img.*, 22:1379–1394, 2003.
19. M. Wand and M. Jones. *Kernel Smoothing*. Chapman & Hall, 1995.

Appendix

In this section we give some further exploration of the calculus of the derivative on the energy term E_{image}. We need first to introduce the Heaviside distribution which we note H and the inverse diffeomorphism of $\mathcal{A} \circ \mathcal{L}(\mathbf{\Theta})$ which we note $\mathcal{G}(\mathbf{\Theta})$. This diffeomorphism therefore verifies:

$$\mathcal{A}(\mathcal{L}(\mathbf{\Theta}, \mathcal{G}(\mathbf{\Theta}, \mathbf{y}))) = \mathbf{y} \tag{5}$$

For simpler notation purpose we also pose:

$$D(\mathbf{x}, \mathbf{y}) = -H(\phi_{\mathcal{M}}(\mathbf{x}))\log(\pi_{\text{cor}}(\mathcal{I}(\mathbf{y}))) - (1 - H(\phi_{\mathcal{M}}(\mathbf{x})))\log(\pi_{\text{bkg}}(\mathcal{I}(\mathbf{y})))$$

Then the image term of the energy (eq. 4) can be rewritten as:

$$E_{\text{image}}(\mathbf{\Theta}) = \int_{\Omega} D(\mathcal{G}(\mathbf{\Theta}, \mathbf{y}), \mathbf{y}) d\mathbf{y}$$

When differentiating Eq. (5) with respect to $\mathbf{\Theta}$ and substituting the expression obtained for $d\mathcal{G}/d\mathbf{\Theta}$ into the expression of $dE_{\text{image}}(\mathbf{\Theta})/d\mathbf{\Theta}$, we get the following:

$$\frac{dE_{\text{image}}(\mathbf{\Theta})}{d\mathbf{\Theta}} = \\ -\int_{\Omega} \frac{\partial D}{\partial \mathbf{x}^T}(\mathcal{G}(\mathbf{\Theta}, \mathbf{y}), \mathbf{y}) \left[\frac{\partial(\mathcal{A} \circ \mathcal{L})}{\partial \mathbf{x}^T}(\mathcal{G}(\mathbf{\Theta}, \mathbf{y}), \mathbf{\Theta})\right]^{-1} \frac{\partial(\mathcal{A} \circ \mathcal{L})}{\partial \mathbf{\Theta}^T}(\mathcal{G}(\mathbf{\Theta}, \mathbf{y}), \mathbf{\Theta}) d\mathbf{y}$$

Now changing the integration variable according to the diffeomorphism $\mathbf{x} = \mathcal{G}(\mathbf{\Theta}, \mathbf{y})$

$$\frac{dE_{\text{image}}(\mathbf{\Theta})}{d\mathbf{\Theta}} = -\int_{\Omega} \frac{\partial D}{\partial \mathbf{x}^T}(\mathbf{x}, \mathcal{A}(\mathcal{L}(\mathbf{\Theta}, \mathbf{x})))\text{com}\left(\frac{\partial(\mathcal{A} \circ \mathcal{L})}{\partial \mathbf{x}^T}(\mathbf{x}, \mathbf{\Theta})\right)^T \frac{\partial(\mathcal{A} \circ \mathcal{L})}{\partial \mathbf{\Theta}^T}(\mathbf{x}, \mathbf{\Theta}) d\mathbf{x}$$

where 'com' denotes the matrix of cofactors. When calculating explicitly the partial derivative of D with respect to its first variable, this integral further simplifies into a curve integral along the reference model:

$$\frac{dE_{\text{image}}(\mathbf{\Theta})}{d\mathbf{\Theta}} = \\ -\oint_{C_{\mathcal{M}}} \tilde{D}(\mathcal{A}(\mathcal{L}(\mathbf{\Theta}, \mathbf{x}))) \left[\text{com}\left(\frac{\partial(\mathcal{A} \circ \mathcal{L})}{\partial \mathbf{x}^T}(\mathbf{x}, \mathbf{\Theta})\right) . \nabla \phi_{\mathcal{M}}(\mathbf{x})\right]^T \frac{\partial(\mathcal{A} \circ \mathcal{L})}{\partial \mathbf{\Theta}^T}(\mathbf{x}, \mathbf{\Theta}) d\mathbf{x}$$

with $\tilde{D}$ defined as:

$$\tilde{D}(\mathbf{y}) = -\log(\pi_{\text{cor}}(\mathcal{I}(\mathbf{y}))) + \log(\pi_{\text{bkg}}(\mathcal{I}(\mathbf{y})))$$

This expression of the derivative refers only to the contour in the model space. Therefore there is no need to parse the entire image domain at every iteration of the gradient descent used in our implementation. Instead, we only scan the model contour at every iterations. Parsing of the images is only necessary when we reevaluate the parameters of the gaussian mixtures for π_{cor} and π_{bkg} (every 20 iteration).

Dynamical Statistical Shape Priors for Level Set Based Sequence Segmentation

Daniel Cremers and Gareth Funka-Lea

Department of Imaging and Visualization,
Siemens Corporate Research, Princeton, NJ

Abstract. In recent years, researchers have proposed to introduce statistical shape knowledge into the level set method in order to cope with insufficient low-level information. While these priors were shown to drastically improve the segmentation of images or image sequences, so far the focus has been on statistical shape priors that are time-invariant. Yet, in the context of tracking deformable objects, it is clear that certain silhouettes may become more or less likely over time. In this paper, we tackle the challenge of learning dynamical statistical models for implicitly represented shapes. We show how these can be integrated into a segmentation process in a Bayesian framework for image sequence segmentation. Experiments demonstrate that such shape priors with memory can drastically improve the segmentation of image sequences.

1 Level Set Based Image Segmentation

In 1988, Osher and Sethian [16] introduced the level set method[1] as a means to implicitly propagate boundaries $C(t)$ in the image plane $\Omega \subset \mathbb{R}^2$ by evolving an appropriate embedding function $\phi : \Omega \times [0, T] \rightarrow \mathbb{R}$, where:

$$C(t) = \{x \in \Omega \mid \phi(x, t) = 0\}. \tag{1}$$

The ordinary differential equation propagating explicit contour points is thus replaced by a partial differential equation modeling the evolution of a higher-dimensional embedding function. The key advantages of this approach are well-known. First, the implicit contour representation does not depend on a specific parameterization and during the propagation no control point regridding mechanisms need to be introduced. Second, evolving the embedding function allows topological changes such as splitting and merging of the embedded contour to be elegantly modeled. In the context of shape modeling and statistical learning of shapes, the latter property allows for the construction of shape dissimilarity measures defined on the embedding functions which can handle shapes of varying topology. Third, the implicit representation (1) naturally generalizes to hypersurfaces in three or more dimensions. To impose a unique correspondence between a contour and its embedding function one can constrain ϕ to be a signed distance function, i.e. $|\nabla\phi| = 1$ almost everywhere.

[1] A precursor of the level set method was proposed by Dervieux and Thomasset [8].

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 210–221, 2005.

Starting in the early 90's researchers proposed to apply the level set method to image segmentation (cf. [12,3,10,17]). Level set implementations of the Mumford-Shah functional [14] were independently proposed in [4,24].

In recent years, researchers have successfully introduced prior shape information into level set based segmentation schemes [11,25,21,5,19,7,22,20]. Statistically learned shape information was shown to cope for missing or misleading information in the input images due to noise, clutter and occlusion. These shape priors were developed to segment objects of familiar shape in a given image. Although they can be applied to tracking objects in image sequences [6,13,7], they are not well-suited for this task, because they neglect the temporal coherence of silhouettes which characterizes the motion of many deforming shapes.

When tracking a three-dimensional deformable object over time, clearly not all shapes are equally likely at a given time instance. Regularly sampled images of a walking person, for example, exhibit a typical pattern of consecutive silhouettes. Similarly, the projections of a rigid 3D object rotating at constant speed are generally not independent samples from a statistical shape distribution. Instead, the resulting set of silhouettes can be expected to contain strong temporal correlations. In this paper, we will develop statistical shape models for which the shape probability at a given time will depend on the shapes observed at previous time steps. The integration of such dynamical shape models into the segmentation process can be elegantly formulated within a Bayesian framework for level set segmentation of image sequences as follows.

2 Level Set Based Tracking as Bayesian Inference

In this section, we will introduce a Bayesian formulation for the problem of level set based image sequence segmentation. We will first treat the general formulation in the space of embedding functions and subsequently for propose a computationally more efficient formulation in a low-dimensional subspace.

2.1 General Formulation

In the following, we define as *shape* a set of closed 2D contours modulo a certain transformation group the elements of which are denoted by T_θ with a parameter vector θ. Depending on the application, these may be rigid-body transformations, similarity or affine transformations or larger transformation groups. The shape is represented implicitly by an embedding function ϕ according to equation (1). Thus objects of interest will be given by $\phi(T_\theta\, x)$, where the transformation T_θ acts on the grid, leading to corresponding transformations of the implicitly represented contour. We purposely separate shape ϕ and transformation parameters θ since one may want to use different models to represent and learn their respective temporal evolution.

Assume we are given consecutive images $I_t : \Omega \to \mathbb{R}$ from an image sequence, where $I_{1:t}$ denotes the set of images $\{I_1, I_1, \ldots, I_t\}$ at different time instances. Assume we have already segmented the images at previous times in terms of

embedding functions $\hat{\phi}_{1:t-1}$ and transformation parameters $\hat{\theta}_{1:t-1}$. The problem of segmenting the current frame I_t can then be addressed in the framework of Bayesian inference by maximizing the conditional probability

$$\mathcal{P}(\phi_t, \theta_t \,|\, I_{1:t},\, \hat{\phi}_{1:t-1},\, \hat{\theta}_{1:t-1}) = \frac{\mathcal{P}(I_{1:t} \,|\, \phi_t, \theta_t, \hat{\phi}_{1:t-1}, \hat{\theta}_{1:t-1})\; \mathcal{P}(\phi_t, \theta_t \,|\, \hat{\phi}_{1:t-1}, \hat{\theta}_{1:t-1})}{\mathcal{P}(I_{1:t} \,|\, \hat{\phi}_{1:t-1}, \hat{\theta}_{1:t-1})},$$

with respect to the embedding function ϕ_t and the transformation parameters θ_t.[2] The denominator in the above expression does not depend on the estimated quantities and can therefore be neglected in the maximization.

In order to further reduce the complexity of the estimation problem, we will make the following assumptions:

- The images $I_{1:t}$ are mutually independent and their probability only depends on the current shape and transformation. Therefore, the first term in the numerator reduces to:

$$\mathcal{P}(I_{1:t} \,|\, \phi_t, \theta_t, \phi_{1:t-1}, \theta_{1:t-1}) = \prod_{i=1}^{t} \mathcal{P}(I_i \,|\, \phi_i, \theta_i) = \mathcal{P}(I_t \,|\, \phi_t, \theta_t) \cdot \text{const.} \qquad (2)$$

- We assume that the intensities of the shape of interest and of the background are independent samples from two Gaussian distributions $K_{\mu,\sigma}(I) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(I-\mu)^2}{2\sigma^2}\right)$ with unknown means μ_1, μ_2 and variances σ_1, σ_2. As a consequence, the data term can be written as:

$$\begin{aligned} \mathcal{P}(I_t \,|\, \phi_t, \theta_t) &= \prod_{\substack{x \\ \phi(T_{\theta_t} x) \geq 0}} K_{\mu_1,\sigma_1}\big(I_t(x)\big) \prod_{\substack{x \\ \phi(T_{\theta_t} x) < 0}} K_{\mu_2,\sigma_2}\big(I_t(x)\big) \\ &\propto \exp\left(-\int_{\Omega} \left(\frac{(I_t - \mu_1)^2}{2\sigma_1^2} + \log \sigma_1 \right) H\phi_t(T_{\theta_t} x) \right. \qquad (3) \\ &\qquad \left. + \left(\frac{(I_t - \mu_2)^2}{2\sigma_2^2} + \log \sigma_2 \right) \Big(1 - H\phi_t(T_{\theta_t} x)\Big) dx \right), \end{aligned}$$

  where we have introduced the Heaviside step function $H\phi \equiv H(\phi)$ to denote the areas where ϕ is positive ($H\phi = 1$) or negative ($H\phi = 0$). Related intensity models for segmentation have been proposed among others in [26,4]. The intensity model parameters μ_i and σ_i are estimated jointly with the shape ϕ_t and the transformation θ_t. Their optimal values are simply given by the means and variances of the intensity I_t inside and outside the current shape. To keep the notation simple, we do not display them as part of the dynamic variables.

[2] Since the modeling of probability distributions on infinite-dimensional spaces is in general an open problem including issues of defining appropriate measures and of integrability, the functions ϕ in this paper may be thought of as finite-dimensional approximations obtained by sampling the embedding functions on a regular grid.

– The prior probability of the current shape and transformation are mutually independent and only depend on their previous estimates. The second term in the numerator therefore simplifies as follows:

$$\mathcal{P}(\phi_t, \theta_t \,|\, \phi_{1:t-1}, \theta_{1:t-1}) = \mathcal{P}(\phi_t \,|\, \phi_{1:t-1})\; \mathcal{P}(\theta_t \,|\, \theta_{1:t-1})$$

By this assumption, we therefore neglect couplings between shape and transformation. Since the focus of the present paper is on modeling temporally correlated shape deformations, we will simply assume a uniform prior on the transformation parameters, i.e. $\mathcal{P}(\theta_t \,|\, \theta_{1:t-1}) = \text{const}$. Rathi et al. [18] recently proposed a temporal model of these transformation parameters while not imposing any specific model on the shape. In this sense, our work is complimentary to theirs. In the following, we will develop appropriate models for the conditional probability $\mathcal{P}(\phi_t \,|\, \phi_{1:t-1})$.

2.2 A Finite-Dimensional Formulation

When estimating the conditional probability $\mathcal{P}(\phi_t \,|\, \phi_{1:t-1})$ from sample data, one needs to revert to finite-dimensional approximations of the embedding function. It is well-known that statistical models can be estimated more reliably if the dimensionality of the data is low. We will therefore recast the Bayesian inference in a low-dimensional formulation given within the subspace spanned by the largest principal eigenmodes of a set of sample shapes.

Let $\{\phi_1, \ldots, \phi_N\}$ be a temporal sequence of training shapes.[3] Let ϕ_0 denote the mean shape and $\psi_1, \ldots, \psi_n$ the n largest eigenmodes with $n << N$. We will then approximate each training shape as:

$$\phi_i(x) = \phi_0(x) \;+\; \sum_{j=1}^{n} \alpha_{ij}\, \psi_j(x), \tag{4}$$

where

$$\alpha_{ij} = \langle \phi_i - \phi_0, \psi_j \rangle \equiv \int (\phi_i - \phi_0)\psi_j \, dx. \tag{5}$$

Such PCA based representations of level set functions have been successfully applied for the construction of statistical shape priors in [11,24,22]. In the following, we will denote the vector of the first n eigenmodes as $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_n)$.

[3] We assume that all training shapes ϕ_i are signed distance functions, yet an arbitrary linear combination of eigenmodes will in general not generate a signed distance function. While the proposed statistical shape models favor shapes which are close to the training shapes (and therefore close to the set of signed distance functions), not all shapes sampled in the considered subspace will correspond to signed distance functions. In addition, it is quite possible that linear combinations result in empty shapes, i.e. the zero level set of a linear combination may be the empty set.

While level set based shape representations via harmonic embedding [9] do form a linear space, such representations are limited in practice, because not every shape can be represented by an appropriate harmonic function.

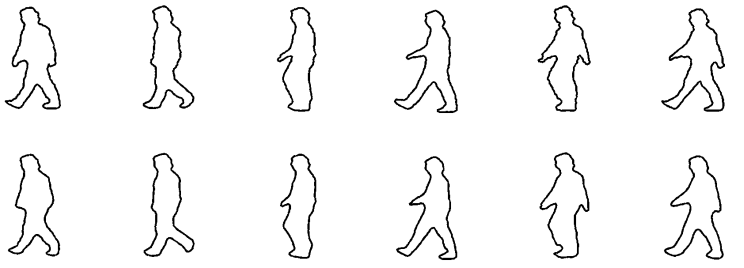


Fig. 1. Low-dimensional approximation of a set of training silhouettes. The silhouettes (above) are approximated by the first 6 principal components of their embedding functions (below) – see equation (4).

Each sample shape ϕ_i is therefore approximated by the n-dimensional shape vector $\boldsymbol{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{in})$. Similarly, an arbitrary shape ϕ can be approximated by a shape vector of the form

$$\boldsymbol{\alpha}_\phi = \langle \phi - \phi_0, \boldsymbol{\psi} \rangle. \tag{6}$$

Figure 1 shows a set of silhouettes from a sequence of a walking person and their approximation by the first 6 eigenmodes. While this approximation is certainly a rough approximation lacking some of the details of the shape, we found it sufficiently accurate for our purpose.

The goal of image sequence segmentation within this subspace can then be stated as follows: Given consecutive images $I_t : \Omega \to \mathbb{R}$ from an image sequence, and given the segmentations $\hat{\boldsymbol{\alpha}}_{1:t-1}$ and transformations $\hat{\theta}_{1:t-1}$ obtained on the previous images $I_{1:t-1}$, we need to maximize the conditional probability

$$\mathcal{P}\big(\boldsymbol{\alpha}_t, \theta_t | I_{1:t}, \{\hat{\boldsymbol{\alpha}}, \hat{\theta}\}_{1:t-1}\big) \propto \mathcal{P}\big(I_{1:t} | \boldsymbol{\alpha}_t, \theta_t, \{\hat{\boldsymbol{\alpha}}, \hat{\theta}\}_{1:t-1}\big)\, \mathcal{P}\big(\boldsymbol{\alpha}_t, \theta_t | \{\hat{\boldsymbol{\alpha}}, \hat{\theta}\}_{1:t-1}\big), \tag{7}$$

with respect to the shape parameters $\boldsymbol{\alpha}_t$ and the transformation parameters θ_t. One can introduce the same approximations as in the previous section. In all expressions the variables ϕ_i are simply replaced by their shape vectors $\boldsymbol{\alpha}_i$. Due to space limitations, we will not carry this out explicitly. The key contribution of this work, is to model the probability

$$\mathcal{P}(\boldsymbol{\alpha}_t \,|\, \hat{\boldsymbol{\alpha}}_{1:t-1}), \tag{8}$$

which constitutes the probability for observing a particular shape conditioned on the previously observed shapes.

Abundant theory has been developed to model temporally correlated time series data. Applications of dynamical systems to model deformable shapes were proposed among others in [2]. In our context, we intend to learn dynamical models for the implicitly represented shapes.

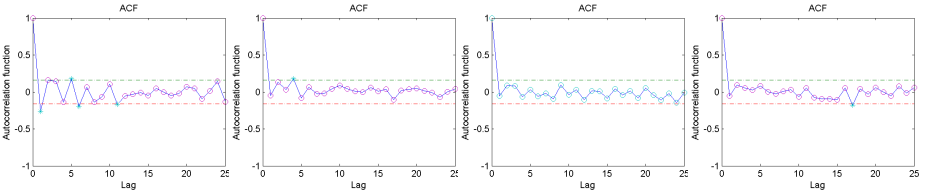


Fig. 2. Autocorrelation functions. To validate the accuracy of the fitted autoregressive model, we plotted the autocorrelation functions of the residuals associated with the first four shape modes. Except for the first mode, more than 95% of autocorrelations (for lag > 0) lie within the confidence limits of an IID process.

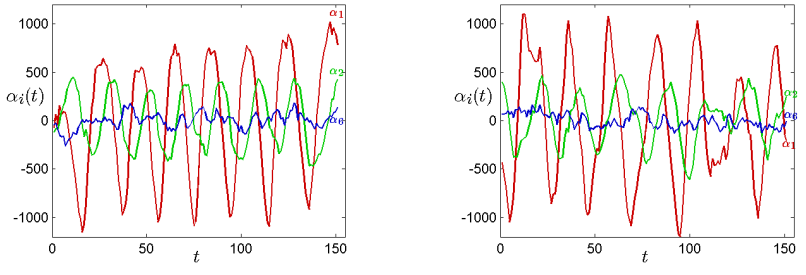


Fig. 3. Model comparison. The original shape sequence **(top)** and the sequence synthesized by a statistically learned second order Markov chain **(bottom)** exhibit similar oscillatory behaviour and amplitude modulation. The plots show the temporal evolution of the first, second and sixth shape eigenmode.

3 Dynamical Statistical Shape Models

In the following, we propose to learn the temporal dynamics of a deforming shape by approximating the shape vectors $\boldsymbol{\alpha}_t \equiv \boldsymbol{\alpha}_{\phi_t}$ of a sequence of silhouettes by a Markov chain (cf. [2,15]) of order k, i.e.:

$$\boldsymbol{\alpha}_t = \boldsymbol{\mu} + A_1 \boldsymbol{\alpha}_{t-1} + A_2 \boldsymbol{\alpha}_{t-2} + \ldots + A_k \boldsymbol{\alpha}_{t-k} + \boldsymbol{\eta}. \tag{9}$$

The state at time t is therefore given by a linear combination of previous states, modeled by a mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and transition matrices $A_1, \ldots, A_k \in \mathbb{R}^{n \times n}$, and zero-mean Gaussian noise $\eta \in \mathbb{R}^n$ with covariance $\Sigma \in \mathbb{R}^{n \times n}$ superimposed. The probability of a shape conditioned on the shapes observed in previous time steps is therefore given by the corresponding autoregressive model of order k:

$$\mathcal{P}(\boldsymbol{\alpha}_t \,|\, \boldsymbol{\alpha}_{1:t-1}) \propto \exp\left(-\frac{1}{2}\, \boldsymbol{v}^\top \Sigma^{-1}\, \boldsymbol{v}\right), \tag{10}$$

where

$$\boldsymbol{v} = \boldsymbol{\alpha}_t - \boldsymbol{\mu} - A_1 \boldsymbol{\alpha}_{t-1} - A_2 \boldsymbol{\alpha}_{t-2} \ldots - A_k \boldsymbol{\alpha}_{t-k} \tag{11}$$

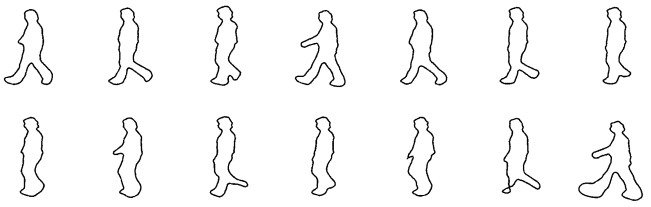


Fig. 4. Synthetically generated walking sequence. Sample silhouettes generated by a statistically learned second order Markov model on the embedding functions – see equation (9). While the Markov model captures much of the typical oscillatory behaviour of a walking person, not all generated samples correspond to permissible shapes – cf. the last two silhouettes on the bottom right. Yet, as we shall see in Section 5, the model is sufficiently accurate to constrain the segmentation process in a meaningful way.

Various methods have been proposed in the literature to estimate the model parameters given by the mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and the matrices $A_1, \ldots, A_k, \Sigma \in \mathbb{R}^{n \times n}$. We applied a stepwise least squares algorithm proposed in [15]. Different tests have been devised to quantify the accuracy of the model fit. Two established criteria for model accuracy are Akaike's Final Prediction Error [1] and Schwarz's Bayesian Criterion [23]. Using dynamical models up to an order of 8, we found that according to Schwarz's Bayesian Criterion, our training sequences were best approximated by an autoregressive model of second order.

From a sequence of 151 consecutive silhouettes, we estimated the parameters of a second order autoregressive model. We subsequently validated this model by plotting the autocorrelation functions of the residuals associated with each of the modeled eigenmodes – see Figure 2. These show that the residuals are essentially uncorrelated.

In addition, the estimated model parameters allow us to synthesize a walking sequence according to (9).[4] Figure 3 shows the temporal evolution of the first, second and sixth eigenmode in the input sequence (left) and in the synthesized sequence. Clearly, the second order model captures some of the key elements of the oscillatory behaviour.

While the synthesized sequence does capture the characteristic motion of a walking person, Figure 4 shows that the individual synthesized silhouettes do not in all instances mimic valid shapes. We believe that such limitations can be expected from a model which strongly compresses the represented input sequence: Instead of 151 shapes defined on a 256×256 grid, the model merely retains a mean shape ϕ_0, 6 eigenmodes $\boldsymbol{\psi}$ and the autoregressive model parameters given by a 6-dimensional mean and three 6×6 matrices. This amounts to 458851 instead of 9895936 parameters, corresponding to a compression to 4.6% of the original size.

[4] In order to remove the dependency on the initial conditions, the first several hundred samples were discarded from the synthesized sequence.

4 Dynamical Shape Priors in Variational Segmentation

Maximizing the conditional probability (7) under the assumptions introduced in Section 2 can be done by minimizing the negative logarithm of (7). Up to a constant, the latter is given by:

$$E(\boldsymbol{\alpha}_t, \theta_t) = E_{data}(\boldsymbol{\alpha}_t, \theta_t) + \nu\, E_{dynamics}(\boldsymbol{\alpha}_t). \tag{12}$$

According to equation (3), the data term is given by:

$$E_{data} = \int_{\Omega} \left(\frac{(I_t - \mu_1)^2}{2\sigma_1^2} + \log \sigma_1 \right) H\phi_{\boldsymbol{\alpha}_t, \theta_t} + \left(\frac{(I_t - \mu_2)^2}{2\sigma_2^2} + \log \sigma_2 \right) \left(1 - H\phi_{\boldsymbol{\alpha}_t, \theta_t} \right) dx,$$

where, for notational simplicity, we have introduced the expression $\phi_{\boldsymbol{\alpha}_t, \theta_t} \equiv \phi_0(T_{\theta_t} x) + \boldsymbol{\alpha}_t^\top \boldsymbol{\psi}(T_{\theta_t} x)$ to denote the embedding function of a shape generated with deformation parameters $\boldsymbol{\alpha}_t$ and transformed with parameters θ_t.

Using the autoregressive model (10), the dynamical shape energy is given by:

$$E_{dynamics}(\boldsymbol{\alpha}_t) = \frac{1}{2}\, \boldsymbol{v}^\top\, \Sigma^{-1}\, \boldsymbol{v} \tag{13}$$

with $\boldsymbol{v}$ defined in (11).

Tracking an object of interest over a sequence of images $I_{1:t}$ with a statistically learnt dynamical shape prior can be done by minimizing energy (12). In this work, we pursue a gradient descent strategy leading to the following differential equations to estimate the shape vector $\boldsymbol{\alpha}_t$ and θ_t:

$$\frac{d\boldsymbol{\alpha}_t(\tau)}{\partial \tau} = -\frac{\partial E_{data}(\boldsymbol{\alpha}_t, \theta_t)}{\partial \boldsymbol{\alpha}_t} - \nu\, \frac{dE_{dynamics}(\boldsymbol{\alpha}_t)}{d\boldsymbol{\alpha}_t} \tag{14}$$

where τ denotes the artificial evolution time, as opposed to the physical time t. The first term is given by:

$$\frac{\partial E_{data}}{\partial \boldsymbol{\alpha}_t} = \left\langle \boldsymbol{\psi}, \delta(\phi_{\boldsymbol{\alpha}_t}) \left(\frac{(I_t - \mu_1)^2}{2\sigma_1^2} - \frac{(I_t - \mu_2)^2}{2\sigma_2^2} + \log \frac{\sigma_1}{\sigma_2} \right) \right\rangle,$$

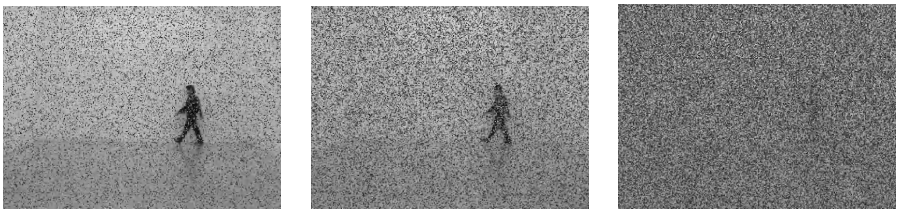
and the second one is given by:

$$\frac{dE_{dynamics}}{d\boldsymbol{\alpha}_t} = \Sigma^{-1}\, \boldsymbol{v}, \tag{15}$$

with $\boldsymbol{v}$ given in (11). These two terms affect the shape evolution in the following manner: The first term draws the shape to separate the image intensities according to the two Gaussian intensity models. Since the effect of variations in the shape vector $\boldsymbol{\alpha}_t$ are given by the eigenmodes $\boldsymbol{\psi}$, the data term is a projection onto these eigenmodes. The second term induces a relaxation of the shape vector $\boldsymbol{\alpha}_t$ toward the most likely shape, given the shapes obtained on previous time frames.

Minimization with respect to the transformation parameters θ_t is obtained by evolving the respective gradient descent equation given by:

$$\frac{d\theta_t(\tau)}{\partial \tau} = -\frac{\partial E_{data}}{\partial \theta_t} = -\left\langle \nabla\psi \frac{d(T_{\theta_t} x)}{d\theta_t}, \delta(\phi_{\boldsymbol{\alpha}_t}) \left(\frac{(I_t - \mu_1)^2}{2\sigma_1^2} - \frac{(I_t - \mu_2)^2}{2\sigma_2^2} + \log \frac{\sigma_1}{\sigma_2} \right) \right\rangle.$$



25% noise **50% noise** **90% noise**

Fig. 5. Images from a sequence with increasing amounts of noise[5]

Fig. 6. Sample segmentations with a static shape prior on a walking sequence with **25%** noise. Constraining the level set evolution to a low-dimensional subspace allows to cope with a certain amount of noise.

5 Segmentation and Tracking Results

In the following, we will apply the dynamical statistical shape prior introduced above for the purpose of level set based tracking.

To construct the shape prior, we hand-segmented a sequence of a walking person, centered and binarized each shape. Subsequently, we determined the set of signed distance functions $\{\phi_i\}_{i=1..N}$ associated with each shape and computed the dominant 6 eigenmodes. Projecting each training shape on these eigenmodes, we obtained a sequence of shape vectors $\{\boldsymbol{\alpha}_i \in \mathbb{R}^6\}_{i=1..N}$. We fitted a second order multivariate autoregressive model to this sequence by computing the mean vector μ, the transition matrices A_1, A_2 and the noise covariance Σ shown in equation (10). Subsequently, we compared segmentations of noisy sequences obtained by segmentation in the 6-dimensional subspace without and with the dynamical statistical prior.

Figure 5 shows a sample input frame from a sequence with 25%, 50%, and 90% noise.[5] Figure 6 shows a set of segmentations obtained without dynamical shape prior on a sequence with 25% noise. While the segmentation without dynamical prior is successful with little noise, Figure 7 shows that it eventually breaks down when the noise level is increased.

Figure 8 shows segmentations of the same sequence as in 7 obtained with a *dynamical* statistical shape prior derived from a second order autoregressive model. Figure 9 shows that the dynamical statistical shape prior provides for good segmentations, even with 90% noise. Clearly, exploiting the temporal sta-

[5] 90% noise means that 90% of all pixels were replaced by a random intensity sampled from a uniform distribution.

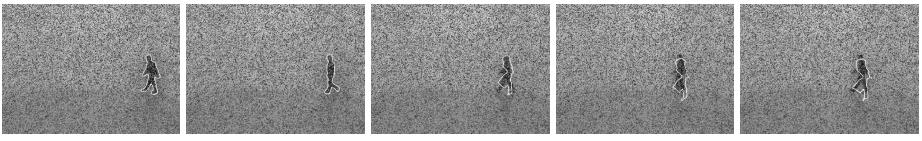


Fig. 7. Sample segmentations with a static shape prior on a walking sequence with **50%** noise. Using merely a static shape prior, the segmentation scheme cannot cope with larger amounts of noise.

Fig. 8. Segmentation using a dynamical statistical shape prior based on a second order autoregressive model. In contrast to the segmentation in Figure 7, the prior imposes statistically learned information about the *temporal dynamics* of the shape evolution to cope with misleading low-level information.

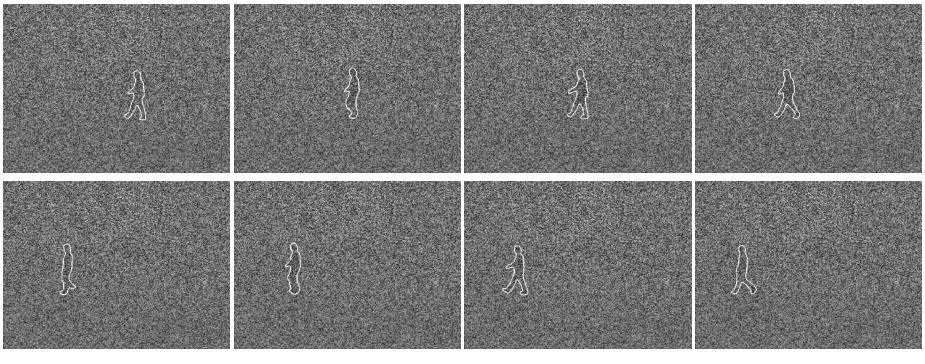


Fig. 9. Tracking with dynamical statistical shape prior to cope with larger amounts of noise. The input images were corrupted with 90% of noise. Yet, the statistically learned dynamical shape model allows to disambiguate the low-level information. These experiments confirm that our tracking schemes can indeed compete with the capacities of human observers.

tistics of dynamical shapes allows to make the segmentation process very robust to missing and misleading information.

6 Conclusion

In this work, we introduced *dynamical* statistical shape models for implicitly represented shapes. In contrast to existing statistical shape models for implicit shapes, these models capture the temporal correlations which characterize deforming shapes such as the consecutive silhouettes of a walking person or the 2D projections of a rotating 3D object. Therefore they account for the fact that the

probability of observing a particular shape at a given time instance may depend on the shapes observed at previous time instances.

For the construction of statistical shape models, we extended the concepts of Markov chains and autoregressive models to the domain of implicitly represented shapes. The resulting dynamical implicit shape models therefore support shapes of varying topology and are easily extended to higher-dimensional shapes (i.e. surfaces).

With the estimated dynamical models one can synthesize shape sequences of arbitrary length. In the context of a sequence of a walking person, we validated the accuracy of the estimated dynamical models, comparing the dynamical shape evolution of the input sequence to that of the synthesized sequence for various shape eigenmodes. In addition, we validated that the residuals are statistically uncorrelated. Although the synthesized shapes do not in all instances correspond to valid shapes, one can nevertheless use the dynamical model to constrain a segmentation process in a meaningful way.

To this end, we developed a Bayesian formulation for level set based image sequence segmentation, which allows to impose the statistically learnt dynamical models as shape priors in the segmentation process. In contrast to most existing approaches to tracking, autoregressive models are integrated as statistical priors in a *variational* approach which can be minimized by local gradient descent (rather than stochastic optimization methods).

Experimental results confirm that the resulting shape priors make it possible to reliably track familiar deformable objects despite large amounts of noise. Future work is focused on further quantitative performance analysis, on the development of statistical models which capture the joint dynamics of deformation and transformation modes, and on the optimization with stochastic methods.

Acknowledgments

We thank Alessandro Bissacco and Payam Saisan for providing the image sequence data. We thank Gianfranco Doretto and Paolo Favaro for helpful discussions on autoregressive models.

References

1. H. Akaike. Autoregressive model fitting for control. *Ann. Inst. Statist. Math.*, 23:163–180, 1971.
2. A. Blake and M. Isard. *Active Contours*. Springer, London, 1998.
3. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. IEEE Intl. Conf. on Comp. Vis.*, pages 694–699, Boston, USA, 1995.
4. T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, 2001.
5. Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. S. Gopinath, R. W. Briggs, and E. Geiser. Using shape priors in geometric active contours in a variational framework. *Int. J. of Computer Vision*, 50(3):315–328, 2002.
6. D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics in Mumford–Shah based segmentation. In A. Heyden et al., editors, *Europ. Conf. on Comp. Vis.*, volume 2351 of *LNCS*, pages 93–108, Copenhagen, May 2002. Springer.

7. D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk. In *Pattern Recognition*, volume 3175 of *LNCS*, pages 36–44. Springer, 2004.
8. A. Dervieux and F. Thomasset. A finite element method for the simulation of Raleigh-Taylor instability. *Springer Lect. Notes in Math.*, 771:145–158, 1979.
9. A. Duci, A. Yezzi, S. Mitter, and S. Soatto. Shape representation via harmonic embedding. In *ICCV*, pages 656–662, 2003.
10. S. Kichenassamy, A. Kumar, P. J. Olver, A. Tannenbaum, and A. J. Yezzi. Gradient flows and geometric active contour models. In *IEEE Intl. Conf. on Comp. Vis.*, pages 810–815, 1995.
11. M. Leventon, W. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *CVPR*, volume 1, pages 316–323, Hilton Head Island, SC, 2000.
12. R. Malladi, J. A. Sethian, and B. C. Vemuri. A topology independent shape modeling scheme. In *SPIE Conf. on Geometric Methods in Comp. Vision II*, volume 2031, pages 246–258, 1994.
13. M. Moelich and T. Chan. Tracking objects with the Chan-Vese algorithm. Technical Report 03-14, Computational Applied Mathematics, UCLA, Los Angeles, 2003.
14. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.
15. A. Neumaier and T. Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM T. on Mathematical Software*, 27(1):27–57, 2001.
16. S. J. Osher and J. A. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. of Comp. Phys.*, 79:12–49, 1988.
17. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. of Computer Vision*, 46(3):223–247, 2002.
18. Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle filtering for geometric active contours and application to tracking deforming objects. In *IEEE Int. Conf. on Comp. Vision and Patt. Recognition*, 2005. To appear.
19. T. Riklin-Raviv, N. Kiryati, and N. Sochen. Unlevel sets: Geometry and prior-based segmentation. In T. Pajdla and V. Hlavac, editors, *European Conf. on Computer Vision*, volume 3024 of *LNCS*, pages 50–61, Prague, 2004. Springer.
20. M. Rousson and D. Cremers. Efficient kernel density estimation of shape and intensity priors for level set segmentation. In *MICCAI*, 2005. To appear.
21. M. Rousson and N. Paragios. Shape priors for level set representations. In A. Heyden et al., editors, *Proc. of the Europ. Conf. on Comp. Vis.*, volume 2351 of *LNCS*, pages 78–92, Copenhagen, May 2002. Springer, Berlin.
22. M. Rousson, N. Paragios, and R. Deriche. Implicit active shape models for 3d segmentation in MRI imaging. In *MICCAI*, pages 209–216, 2004.
23. G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6:461–464, 1978.
24. A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky. Model–based curve evolution technique for image segmentation. In *Comp. Vision Patt. Recog.*, pages 463–468, Kauai, Hawaii, 2001.
25. A. Tsai, A. J. Yezzi, and A. S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. on Image Processing*, 10(8):1169–1186, 2001.
26. S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE PAMI*, 18(9):884–900, 1996.

Non-rigid Shape Comparison of Implicitly-Defined Curves

Sheshadri R. Thiruvenkadam[1], David Groisser[2], and Yunmei Chen[2]

[1] University of California, Los Angeles, CA-90024, USA
sheshad@math.ucla.edu
[2] University of Florida, Gainesville, FL-32611, USA
{groisser, yun}@math.ufl.edu

Abstract. We present a novel variational model to find shape-based correspondences between two sets of level curves. While the usual correspondence techniques work with parametrized curves, we use a level-set formulation that enables us to handle curves with arbitrary topology. Given the functions $\Phi_1 : (\Omega_1 \subseteq \mathbb{R}^2) \rightarrow \mathbb{R}$ and $\Phi_2 : (\Omega_2 \subseteq \mathbb{R}^2) \rightarrow \mathbb{R}$ whose 0-level curves we want to match, we search for a diffeomorphism that minimizes the rate of change of the difference in tangential orientation of the zero-level sets. To make the formulation symmetric and to simplify computations, we map the domains of the level-set functions Φ_i to a common domain Ω by initial diffeomorphisms that are chosen arbitrarily. We then search for diffeomorphisms from Ω to itself, generating them by flows of certain vector fields on Ω. The resulting correspondences are scale- and rotation-invariant with respect to the curves. We show how this model can be used as a basis to compare curves of different topology. The model was tested on synthetic and MRI cardiac data,with good results.

1 Introduction

Detecting non-rigid correspondences between curves is an important problem with many applications in computer vision and medical imaging, including motion analysis[19,5,15,14], shape analysis[16,6], feature-based registration[17] and knowledge-based segmentation[24,4]. Given a pair of curves C_1 and C_2, the problem is formulated as finding a meaningful transformation $\Im : C_1 \rightarrow C_2$ that minimizes a shape-dissimilarity measure between the curves. Methods like [20,26,25], have addressed the correspondence problem successfully for parametrized curves.

In this paper, we propose a framework to solve the correspondence problem for implicitly represented curves, to handle curves of arbitrary topology. We assume that the curves C_1 and C_2 are 0-level sets of higher dimensional functions.

Recently, some techniques have been proposed to match implicitly-defined curves [23,12]. In general, such methods find a suitable transformation $\top : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that minimizes the Euclidean distance between the curves. These spatial transformations $\top$ can be modeled using splines [2,3] or dense deformations [23]. However, the cost of accurately modeling local variations between

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 222–234, 2005.

the curves is high, since transformations of the embedding domain have to drive the curve deformations. Moreover, this approach does not guarantee the point-correspondence criterion $\top(C_1) = C_2$, which is essential in many applications, e.g. motion-tracking. These methods are more suited for curve-matching, whereas our work is applicable both to curve-matching and to motion-tracking of implicitly represented curves.

One of the novelties of our work is that we solve the correspondence problem for implicitly represented curves by *directly* computing transformations between the curves. To be more precise, we look for a diffeomorphism $\Im : C_1 \rightarrow C_2$. A worthy mention here is the work of Memoli et al.[21] to find maps between two general curved manifolds, represented implicitly. Recently there has been interests in modeling diffeomorphic transformations for landmark matching[13,9,18] and image matching[7] using flows of time-dependent vector fields. This approach has been found to be suitable for generating large non-rigid deformations of the plane. We use a similar idea to construct our search space of diffeomorphisms between the curves. To simplify computations, we map our curves initially to a topologically equivalent object, say C(e.g. in the 1-curve case, C is chosen to be a circle.). We then look for diffeomorphisms of C, that are generated by flows of tangential vector fields of C. A similar approach is taken in [9,1], but for matching landmarks on surfaces.

The similarity measure chosen for a shape-matching problem usually depends on the type of *variability* expected in that class of shapes. Euclidean distance [23,12], curvatures[25,15] and curve normal[14] are some of the criteria that can be used. For a survey on similarity measures, we refer the reader to [27]. Our method is geared towards curves that differ from each other primarily by local stretching and piecewise-rigid motions. For this type of variability, we propose to use curve normals as the shape-matching criterion. It is to be noted that use of normals, does allow non-rigid variability between the curves(e.g. all convex curves have the same unit normal field).

Our model is novel in three aspects. First, as mentioned above, we directly compute diffeomorphisms between the curves. Second, our approach hybridizes level-set and parametric techniques, using the former to compute normal vectors and to provide a convenient cartesian domain in which to perform calculations, irrespective of the topology of the curves. Third, the correspondences our method produces are symmetric under interchange of C_1 and C_2, when the curves have the same topology. Further, the resulting correspondences are scale-/rotation-invariant with respect to the curves.

2 Related Work

Our work can be seen as a modification and extension of Tagare's shape-matching technique [25] to implicitly-defined curves. Tagare uses a specific class of correspondences called *bimorphisms* to match curves. We briefly review his work here.

Let C_1 and C_2 be two curves with lengths L_{C_1} and L_{C_2}, parametrized by arc-length. A parametrized bimorphism is a map $\mu : [0,1] \rightarrow [0,1] \times [0,1]$, given by

$\mu(t) = [\frac{s_1(t)}{\mathrm{L}_{C_1}}, \frac{s_2(t)}{\mathrm{L}_{C_2}}]$,where $s_1(t) : [0,1] \rightarrow [0, \mathrm{L}_{C_1}]$ and $s_2(t) : [0,1] \rightarrow [0, \mathrm{L}_{C_2}]$ are surjective functions, satisfying the conditions $s_1'(t) \geq 0$, $s_2'(t) \geq 0$ and $|\dot{\mu}(t)| > 0$. The functions $s_1(t)$ and $s_2(t)$ thus defined intrinsically model continuous, non-rigid correspondences between the curves.

Tagare defines a *shape-based correspondence* between C_1 and C_2 to be a bimorphism $\mu(t) = [\frac{s_1(t)}{\mathrm{L}_{C_1}}, \frac{s_2(t)}{\mathrm{L}_{C_2}}]$ that minimizes the following energy:

$$J_{C_1,C_2}[s_1, s_2] = \int_{[\mu]} [\frac{d\Theta_1}{ds_\mu}(s_1(t)) - \frac{d\Theta_2}{ds_\mu}(s_2(t))]^2 ds_\mu \tag{1}$$

where $\Theta_i(s)$ is the angular orientation of the normal to the curve C_i at arc length s and $ds_\mu = \sqrt{\frac{s_1'(t)^2}{\mathrm{L}_{C_1}^2} + \frac{s_2'(t)^2}{\mathrm{L}_{C_2}^2}}$ dt is the arc-length element of $[\mu] :=$ Image(μ). The objective function J can be seen to be symmetric and scale/rotation-invariant with respect to C_1 and C_2.

Since what we want to minimize in (1) is $(\frac{d}{ds_\mu}[\Theta_1(s_1(t)) - \Theta_2(s_2(t))])^2$,we can use one derivative fewer in our formulation by seeking a constant θ such that $\Theta_1(s_1(t)) - \Theta_2(s_2(t)) \approx \theta$. To use a similar idea to match implicitly-defined curves, it is convenient to compare the curve-normal functions $\boldsymbol{n}_1(s_1(t))$ and $\boldsymbol{n}_2(s_2(t))$ instead. This motivates us to reformulate (1) as:

$$J_{C_1,C_2}[s_1, s_2, \theta] = \int_{[\mu]} |\boldsymbol{n}_1(s_1(t)) - \mathrm{R}_\theta \boldsymbol{n}_2(s_2(t))|^2 ds_\mu \tag{2}$$

where R_θ is the counter-clockwise rotation by the angle θ.

3 Bimorphism for Level Curves

Let Ω,Ω_1 and Ω_2 be bounded regions in $\mathbb{R}^2$. Given Lipschitz functions $\Phi_1 : \Omega_1 \rightarrow \mathbb{R}$ and $\Phi_2 : \Omega_2 \rightarrow \mathbb{R}$ with 0-level curves C_1 and C_2 , we extend the problem of computing shape-based bimorphisms for these embedded curves. We require Φ_i to have non-vanishing gradients on their 0-level sets. For E.g. Φ_i can be chosen as the signed distance function of C_i. For the sake of simplicity, we start with the case when C_1 and C_2 have the topology of a circle. Let C be a unit circle in $\mathbb{R}^2$ centered at the origin, and $\Phi : \Omega \rightarrow \mathbb{R}$ its signed distance function.

We define a *bimorphism* between the embedded curves C_1 and C_2 to be a map $\upsilon : C \rightarrow (C_1 \times C_2) \subseteq \mathbb{R}^4$ given by $\upsilon = [\frac{\hat{f}_1}{\mathrm{L}_{C_1}}, \frac{\hat{f}_2}{\mathrm{L}_{C_2}}]$, where $\hat{f}_i : C \rightarrow C_i$ is a diffeomorphism and $\mathrm{L}_{C_i} = \int_{\Omega_i} \delta(\Phi_i)|\nabla\Phi_i| d\boldsymbol{x}$ is the length of C_i. Here $\delta(\Phi(\boldsymbol{x}))$ is the *Dirac delta function*. In our implementation, we use a smooth approximation for the delta function, denoted $\delta_\varepsilon(t)$. Along the same lines of (2), in this paper we define a *shape-based correspondence* between C_1 and C_2 to be a bimorphism that minimizes the energy:

$$\widehat{E}_{C_1,C_2}[\hat{f}_1, \hat{f}_2, \theta] = \int_{[\upsilon]} |\boldsymbol{\upsilon}_1(\hat{f}_1) - \mathrm{R}_\theta \boldsymbol{\upsilon}_2(\hat{f}_2)|^2 ds_\upsilon \tag{3}$$

where $\boldsymbol{v}_i = \frac{\nabla \Phi_i}{|\nabla \Phi_i|}$, is the unit normal vector field to the level sets of Φ_i and ds_υ is the length element of υ.

Now, we shift our computations over to the domain Ω, by writing (3) in a level set formulation. Let $\mathrm{N}_\epsilon(C) = \{x \in \Omega | \Phi(x) < \epsilon\}$, $\mathrm{N}_{\epsilon_i}(C_i) = \{x \in \Omega_i | \Phi_i(x) < \epsilon_i\}$ be neighborhoods of C and C_i respectively, such that we can extend the diffeomorphism $f_i : C \to C_i$ to a diffeomorphism $\bar{f}_i : \mathrm{N}_\epsilon(C) \to \mathrm{N}_{\epsilon_i}(C_i)$, i.e. $\bar{f}_i|_C = \hat{f}_i$. Consider the map $F : \mathrm{N}_\epsilon(C) \to \mathbb{R}^4$ defined as $F = [\frac{\bar{f}_1}{\mathrm{L}_{C_1}}, \frac{\bar{f}_2}{\mathrm{L}_{C_2}}]$. The graph of the map F is a surface in $\mathbb{R}^4$, with surface-element $|F_x \wedge F_y| d\boldsymbol{x} = \sqrt{|F_x|^2 |F_y|^2 - |F_x \cdot F_y|^2} d\boldsymbol{x}$. Since $F|_C = \upsilon$, the length of the curve υ embedded in $\mathbb{R}^4$ can be computed by:

$$\int_\upsilon ds_\upsilon = \int_\Omega \delta(\Phi) |\nabla \Phi| |F_x \wedge F_y| d\boldsymbol{x} = \int_\Omega \delta(\Phi) |F_x \wedge F_y| d\boldsymbol{x},$$

since $|\nabla \Phi| = 1$. Hence, we can rewrite (3) in the form

$$\overline{E}(\bar{f}_1, \bar{f}_2, \theta) = \int_\Omega \delta(\Phi(\boldsymbol{x})) |\boldsymbol{v}_1(f_1(\boldsymbol{x})) - \mathrm{R}_\theta \boldsymbol{v}_2(f_2(\boldsymbol{x}))|^2 \; |F_x \wedge F_y| d\boldsymbol{x} \tag{4}$$

4 Decomposition of the Bimorphism

In the energy (4), the search-space is over diffeomorphisms $\bar{f}_1, \bar{f}_2 : \mathrm{N}_\epsilon(C) \to \mathbb{R}^2$,with $\bar{f}_i(C) = C_i$. Let $f_{0,i} : \mathrm{N}_\epsilon(C) \to \Omega_1$(i=1,2), be diffeomorphisms arbitrarily chosen such that $f_{0,i}(C) = C_i$. Then given $\bar{f}_i$ there exists a unique diffeomorphism $f_i : \mathrm{N}_\epsilon(C) \to \mathrm{N}_\epsilon(C)$ with $f_i(C) = C$, such that $\bar{f}_i = f_{0,i} \circ f_i$(Fig.1).

Thus we can replace our search-space by the space of diffeomorphisms $f_1, f_2 : \mathrm{N}_\epsilon(C) \to \mathrm{N}_\epsilon(C)$ such that $f_i(C) = C$. Now the energy functional (4) is written as:

$$\widetilde{E}(f_1, f_2, \theta) = \int_\Omega \delta(\Phi(\boldsymbol{x})) |\boldsymbol{v}_1(f_{0,1} \circ f_1) - \mathrm{R}_\theta \boldsymbol{v}_2(f_{0,2} \circ f_2)|^2 \; |F_x \wedge F_y| d\boldsymbol{x} \tag{5}$$

where $F = [\frac{f_{0,1} \circ f_1}{\mathrm{L}_{C_1}}, \frac{f_{0,2} \circ f_2}{\mathrm{L}_{C_2}}]$.
We can observe that the set of minimizers to energy functional (5), if nonempty, is invariant with respect to

1. Initial maps $f_{0,i}$,
2. Rotation and scaling transformations on the level curves C_1 and C_2.

We restrict our search for f_i to flows of vector fields, of the form $\boldsymbol{X_i} = a_i(\boldsymbol{x})\boldsymbol{T}$, where $\boldsymbol{T} = \nabla \Phi^\perp$ is the tangential vector field to the level-curves of Φ obtained by counterclockwise rotation of $\nabla \Phi$, and where $a_i : \Omega \to \mathbb{R}$ is a Lipschitz function with compact support in $\mathrm{N}_\epsilon(C)$.
For a fixed $\boldsymbol{x}$, Consider the flow $\chi_i(\boldsymbol{x}, t)$ of $\boldsymbol{X_i}$ given by the following differential equation:

$$\dot{\chi}_\mathrm{i}(\boldsymbol{x}, \mathrm{t}) = \boldsymbol{X_i}(\chi_\mathrm{i}(\boldsymbol{x}, \mathrm{t})), \; \chi_\mathrm{i}(\boldsymbol{x}, 0) = \boldsymbol{x}.$$

Then for a fixed $\tau > 0$, $\chi_i(\boldsymbol{x}, \tau) : \mathrm{N}_\epsilon(C) \to \mathrm{N}_\epsilon(C)$ is a diffeomorphism and $\chi_i(C, \tau) = C$. Naturally, we let $f_i(\boldsymbol{x}) = f_i(\boldsymbol{x}; a_i) = \chi_i(\boldsymbol{x}, \tau)$.

Hence our search for diffeomorphisms $f_i : \mathrm{N}_\epsilon(C) \to \mathrm{N}_\epsilon(C)$ satisfying $f_i(C) = C$, reduces to looking for Lipschitz functions a_i with compact support in $\mathrm{N}_\epsilon(C)$. The energy functional (5) now becomes:
$E_0(a_1, a_2, \theta)=$

$$\int_\Omega \delta(\Phi(\boldsymbol{x}))|\boldsymbol{v}_1(f_{0,1} \circ f_1(\boldsymbol{x}; a_1)) - \mathrm{R}_\theta \boldsymbol{v}_2(f_{0,2} \circ f_2(\boldsymbol{x}; a_2))|^2 \; |F_x \wedge F_y| d\boldsymbol{x} \quad (6)$$

5 Approximation

We approximate our minimization problem (6) by restricting our search space for a_1 and a_2 to be finite-dimensional. Let $\{\boldsymbol{x}_j\}_{j=1}^N$ be a finite set of regularly spaced points on C. We look at N basis-functions $\{\psi_j\}_{j=1}^N$, with ψ_j centered at the control-point $\boldsymbol{x}_j$, defined by $\psi_j(\boldsymbol{x}) = \eta_\epsilon(r)(1 - \frac{cos^{-1}(\varsigma_j)}{\pi})$,where $r = |\boldsymbol{x}| - 1, \varsigma_j = \frac{\boldsymbol{x} \cdot \boldsymbol{x}_j}{|\boldsymbol{x}|}$, $\eta_\epsilon \in C_c^\infty(\mathbb{R})$ is the standard mollifier of width ϵ. Let $\Psi(\boldsymbol{x}) = [\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), ..., \psi_N(\boldsymbol{x})]$. To solve our problem , we find constants $\boldsymbol{c}_1 = [c_{11}, c_{12}, ..., c_{1N}]$ and $\boldsymbol{c}_2 = [c_{21}, c_{22}, ..., c_{2N}]$, and such that
$a_i = \sum_{j=1}^N c_{ij}\psi_i = \boldsymbol{c}_i \cdot \Psi$, and θ minimize (6) over $\mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}/2\pi\mathbb{Z}$. Hence $a_i \in C_c^0(N_\epsilon(C))$, as needed. Note that we can vary the smoothness of a_i by appropriately choosing the basis functions ψ_i. To summarize, we minimize:
$E(\boldsymbol{c}_1, \boldsymbol{c}_2, \theta) =$

$$\int_\Omega \delta(\Phi(\boldsymbol{x}))|\boldsymbol{v}_1(f_{0,1} \circ f_1(\boldsymbol{x}; \boldsymbol{c}_1)) - \mathrm{R}_\theta \boldsymbol{v}_2(f_{0,2} \circ f_2(\boldsymbol{x}; \boldsymbol{c}_2))|^2 \; |F_x \wedge F_y| d\boldsymbol{x} \quad (7)$$

over $\mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}/2\pi\mathbb{Z}$.
Here $F = [\frac{f_{0,1} \circ f_1}{\mathrm{L}_{C_1}}, \frac{f_{0,2} \circ f_2}{\mathrm{L}_{C_2}}]$, $f_i(\boldsymbol{x}) = \chi_i(\boldsymbol{x}, \tau)$ where χ_i satisfy the flow equations:

$$\dot{\chi_\mathrm{i}}(\boldsymbol{x}, \mathrm{t}) = (\boldsymbol{c}_\mathrm{i} \cdot \Psi(\chi_\mathrm{i}(\boldsymbol{x}, \mathrm{t})))\boldsymbol{T}(\chi_\mathrm{i}(\boldsymbol{x}, \mathrm{t})), \; \chi_\mathrm{i}(\boldsymbol{x}, 0) = \boldsymbol{x},$$

For any map T, we denote its Jacobian matrix as DT. Denote $f_{i,c_{ik}} = \frac{\partial f_i}{\partial c_{ik}}$. Let $\hat{\boldsymbol{V}} = \boldsymbol{v}_1(f_{0,1} \circ f_1(\boldsymbol{x}; \boldsymbol{c}_1)) - \mathrm{R}_\theta \boldsymbol{v}_2(f_{0,2} \circ f_2(\boldsymbol{x}; \boldsymbol{c}_2))$. The Euler-Lagrange equations for this minimization problem are:

$$\frac{\partial \mathrm{E}}{\partial \mathrm{c_{1k}}} = \int_\Omega \delta(\Phi)\hat{\boldsymbol{V}}.\mathrm{D}\boldsymbol{v}_1.\mathrm{Df}_{0,1}.\mathrm{f}_{1,c_{1k}})|\mathrm{F_x} \wedge \mathrm{F_y}| + |\hat{\boldsymbol{V}}|^2 \frac{\partial}{\partial \mathrm{c_{1k}}}(|\mathrm{F_x} \wedge \mathrm{F_y}|)\mathrm{d}\boldsymbol{x} = 0, \quad (8)$$

$$\frac{\partial E}{\partial c_{2k}} = \int_\Omega \delta(\Phi)(-\hat{\boldsymbol{V}}.\mathrm{R}_\theta D\boldsymbol{v}_2.Df_{0,2}.f_{2,c_{2k}}|F_x \wedge F_y| + |\hat{\boldsymbol{V}}|^2 \frac{\partial}{\partial c_{2k}}(|F_x \wedge F_y|)d\boldsymbol{x} = 0, \quad (9)$$

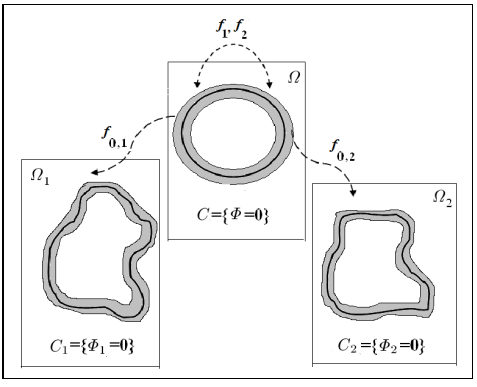


Fig. 1. The shaded grey bands are $\mathrm{N}_\epsilon(C)$, $\mathrm{N}_{\epsilon_i}(C_i)$. $f_{0,i}$ are initial diffeomorphisms. f_i are the minimization variables.

$$\frac{\partial E}{\partial \theta} = \int_\Omega \delta(\Phi)(-\hat{\boldsymbol{V}}.R'_\theta \boldsymbol{v}_2)|F_x \wedge F_y| = 0. \tag{10}$$

In (8) and (9),

$$\frac{\partial}{\partial c_{ik}}(|F_x \wedge F_y|) = \frac{1}{|F_x \wedge F_y|}[|F_y|^2 F_x \cdot F_{xc_{ik}} + |F_x|^2 F_y \cdot F_{yc_{ik}} - (F_x \cdot F_y)$$
$$(F_x \cdot F_{yc_{ik}} + F_y \cdot F_{xc_{ik}})].$$

To simplify further, we consider $f_{i,c_{ik}}(x; \boldsymbol{c}_i)$. Since $f_i(\boldsymbol{x}) = \chi_i(\boldsymbol{x}, \tau)$, we can compute $f_{i,c_{ik}}(\boldsymbol{x}) = \chi_{i,c_{ik}}(\boldsymbol{x}, \tau)$ from the differential equation:

$$\dot{\chi_{\mathrm{i,c_{ik}}}}(\boldsymbol{x}, \mathrm{t}) = \psi_{\mathrm{k}}(\chi_{\mathrm{i}})\boldsymbol{T}(\chi_{\mathrm{i}}) + (\boldsymbol{c}_{\mathrm{i}} \cdot \Psi(\chi_{\mathrm{i}}))\mathrm{D}\boldsymbol{T}(\chi_{\mathrm{i}}) \cdot \chi_{\mathrm{i,c_{ik}}},\ \chi_{\mathrm{i,c_{ik}}}(\boldsymbol{x}, 0) = \boldsymbol{0}.$$

Using this in $F(\boldsymbol{x}; \boldsymbol{c}_1, \boldsymbol{c}_2) = [\frac{f_{0,1} \circ f_1(\boldsymbol{x}; \boldsymbol{c}_1)}{\mathrm{L}_{C_1}}, \frac{f_{0,2} \circ f_2(\boldsymbol{x}; \boldsymbol{c}_2)}{\mathrm{L}_{C_2}}]$, we get

$F_{c_{1k}} = [\frac{Df_{0,1} \cdot f_{1,c_{1k}}}{\mathrm{L}_{C_1}}, 0]$ and $F_{c_{2k}} = [0, \frac{Df_{0,2} \cdot f_{2,c_{2k}}}{\mathrm{L}_{C_2}}]$. Since $F_{xc_{ik}} = F_{c_{ik}x}$ and $F_{yc_{ik}} = F_{c_{ik}y}$, we can write:

$$\frac{\partial}{\partial c_{ik}}(|F_x \wedge F_y|) = \frac{1}{|F_x \wedge F_y|}[|F_y|^2 F_x \cdot F_{c_{ik}x} + |F_x|^2 F_y \cdot F_{c_{ik}y} - (F_x \cdot F_y)$$
$$(F_x \cdot F_{c_{ik}y} + F_y \cdot F_{c_{ik}x})].$$

6 Numerical Implementation

To discretize Eqs.(8-10), we use a finite difference scheme. For any map T defined on Ω let T_{pq} denote its value at the pixel $\boldsymbol{x}_{pq} = [ph, qh]$, where h is the pixel size. We use gradient-descent to minimize the energy (7) over $\mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}/2\pi\mathbb{Z}$. Given initial values $\boldsymbol{c}_1^0, \boldsymbol{c}_2^0$ and θ^0, we iteratively solve for $\boldsymbol{c}_1^m, \boldsymbol{c}_2^m$ and θ^m, (m=1,2,3....).

For an iterate $\boldsymbol{c}_i^m$, θ^m, let us denote $f_{i,pq}^m = f_i(\boldsymbol{x}_{pq}) = \chi_i(\boldsymbol{x}_{pq}, \tau)$. We can compute $f_{i,pq}^m$ by numerically solving the equation:

$$\dot{\chi_{\mathrm{i}}}(\boldsymbol{x}_{\mathrm{pq}}, \mathrm{t}) = (\boldsymbol{c}_{\mathrm{i}}^{\mathrm{m}} \cdot \Psi(\chi_{\mathrm{i}}(\boldsymbol{x}_{\mathrm{pq}}, \mathrm{t})))\boldsymbol{T}(\chi_{\mathrm{i}}(\boldsymbol{x}_{\mathrm{pq}}, \mathrm{t})), \quad \chi_{\mathrm{i}}(\boldsymbol{x}_{\mathrm{pq}}, 0) = \boldsymbol{x}_{\mathrm{pq}}.$$

for $t \in [0, \tau]$.

Let $f_{i,c_{ik},pq}^m$ denote $\frac{\partial f_i}{\partial c_{ik}}(\boldsymbol{x}_{pq}; \boldsymbol{c}_i^m) = \frac{\partial \chi_i}{\partial c_{ik}}(\boldsymbol{x}_{pq}, \tau)$, $(k = 1, ..N)$. We can compute $f_{i,c_{ik},pq}^m$ by numerically solving the equation:

$$\dot{\chi_{\mathrm{i,c_{ik}}}}(\boldsymbol{x}_{\mathrm{pq}}, \mathrm{t}) = \psi_{\mathrm{k}}(\chi_{\mathrm{i}})\boldsymbol{T}(\chi_{\mathrm{i}}) + (\boldsymbol{c}_{\mathrm{i}}^{\mathrm{m}} \cdot \Psi(\chi_{\mathrm{i}}))\mathrm{D}\boldsymbol{T}(\chi_{\mathrm{i}}) \cdot \chi_{\mathrm{i,c_{ik}}}, \quad \chi_{\mathrm{i,c_{ik}}} = \mathbf{0}$$

for $t \in [0, \tau]$. At the iteration m, the bimorphism is given by $F^m = [\frac{f_{0,1} \circ f_1^m}{\mathrm{L}_{C_1}}, \frac{f_{0,2} \circ f_2^m}{\mathrm{L}_{C_2}}]$.

Hence, writing $\boldsymbol{u}^m = [\boldsymbol{c}_1^m, \boldsymbol{c}_2^m, \theta^m]$ we arrive at the discrete form for the Euler-Lagrange equations (8)-(10):

$$E_{c_{1k}}(\boldsymbol{u}^m) = \sum_{p,q} \delta_\varepsilon(\Phi_{pq}) \hat{\boldsymbol{V}}_{pq}^m . D\boldsymbol{v}_{1,pq}^m . Df_{0,1} . f_{1,c_{1k}}^m |F_x^m \wedge F_y^m| + |\hat{\boldsymbol{V}}_{pq}^m|^2 \frac{\partial}{\partial c_{1k}} (|F_x^m \wedge F_y^m|) = 0, \tag{11}$$

$$E_{c_{2k}}(\boldsymbol{u}^m) = \sum_{p,q} \delta_\varepsilon(\Phi_{pq}) (-\hat{\boldsymbol{V}}_{pq}^m . \mathrm{R}_{\theta^m} D\boldsymbol{v}_2^m . Df_{0,2} . f_{2,c_{2k}}^m) |F_x^m \wedge F_y^m| + |\hat{\boldsymbol{V}}_{pq}^m|^2 \frac{\partial}{\partial c_{2,k}} (|F_x^m \wedge F_y^m|) = 0, \tag{12}$$

$$E_\theta(\boldsymbol{u}^m) = \sum_{p,q} \delta_\varepsilon(\Phi_{pq}) (-\hat{\boldsymbol{V}}_{pq}^m . R'_{\theta^m} \boldsymbol{v}_{2,pq}^m) |F_x^m \wedge F_y^m| = 0 \tag{13}$$

$(k = 1, ..N)$, where $\delta_\varepsilon(t)$ is a regularized approximation for $\delta(t)$(see [9] for details). In (11)-(15), the derivatives are approximated using first-order difference schemes. We summarize the algorithm as follows:

Input initial $\boldsymbol{u}^0$
for $N = 1, 4, 8...$
 for $m = 0, 1, 2...$
  From $\boldsymbol{u}^m$ obtain $(f_1^m, f_2^m, f_{1,c_{1k}}^m, f_{2,c_{2k}}^m)$ using (11,12).
  Compute $\nabla E(\boldsymbol{u}^m)$ using (13)-(15).
  Update $\boldsymbol{u}^{m+1}$ using gradient descent.
 end
Re-initialize $\boldsymbol{u}^0$.
end

(14)

The step N=1,4,8... is necessary for the algorithm to avoid local minimizers of (6). We start with a given $\boldsymbol{u}^0$ for the N=1 step and re-initialize $\boldsymbol{u}^0$ for next value of N, by interpolating the minimizer for the current value of N. This is continued until the change in energy (6) goes below a threshold.

The diffeomorphisms $f_{0,1}$ and $f_{0,2}$ map the annular region $\mathrm{N}_\epsilon(C)$ to neighborhoods of C_1 and C_2(Fig.1). Note that the minimum value (if it exists) is independent of choices of $f_{0,i}$. Hence we are free to choose $f_{0,i}$ as simply and quickly as possible (e.g. by linear interpolation of some parameter).

7 Curves of Arbitrary Topology

So far we have restricted to the case when the curves being matched have the circle-topology. The model as it is can be used to match two arbitrary, closed, 0-level curves of the same topology. An interesting extension is to match closed, 0-level-curves with different topologies. Below, we briefly introduce shape-based bimorphisms between implicitly-defined curves C_1 and C_2 of arbitrary topology.

For simplicity, in this paper we consider only the case in which C_1 has the circle-topology and C_2 has the topology of two non-nested circles. Let $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ be any two distinct points on C. We denote $\widetilde{C} = C - \{\boldsymbol{p}_1, \boldsymbol{p}_2\}$.

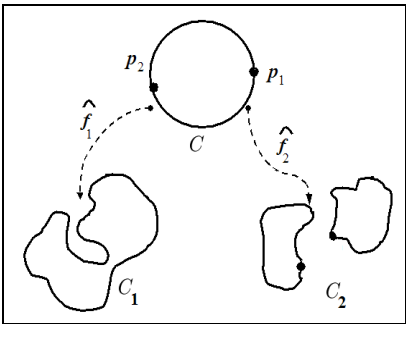


Fig. 2. Bimorphisms for multiple-curves of different topology

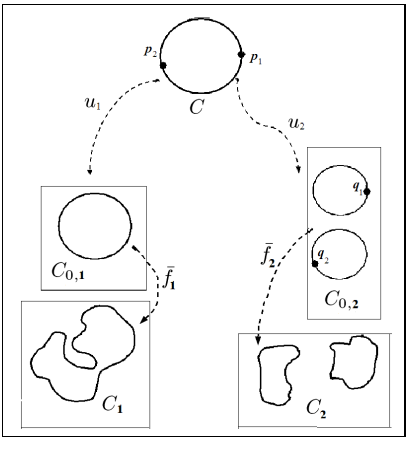


Fig. 3. Computing bimorphism search space for the different topology case

A bimorphism between the implicitly-defined curves C_1 and C_2 is a map $\upsilon : \widetilde{C} \to (C_1 \times C_2) \subseteq \mathbb{R}_4$ given by $\upsilon = [\frac{\hat{f}_1}{\mathrm{L}_{C_1}}, \frac{\hat{f}_2}{\mathrm{L}_{C_2}}]$. Here, $\hat{f}_1$ is the restriction to $\widetilde{C}$, of a diffeomorphism from C to C_1, and $\hat{f}_2 : \widetilde{C} \to C_2$ is a diffeomorphism(preserving C_2's orientation) whose image is C_2 minus two points, one on each of the connected components of C_2.(Fig.2). Denote the images of the maps f_i by $\widetilde{C}_i$.

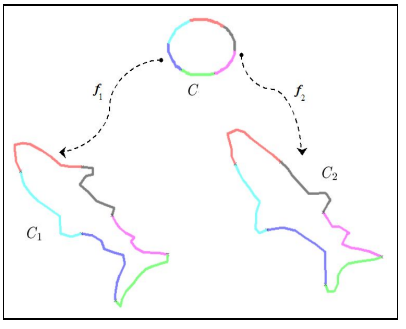


Fig. 4. Shape-based bimorphism between 2 shark curves. Colored segments on C are mapped to the same-colored segments on C_1 and C_2.

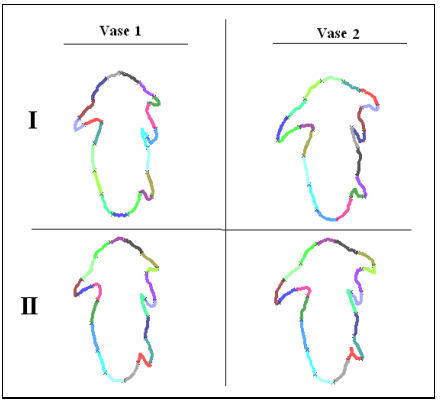


Fig. 5. I. Initial correspondence: colored segments not matched properly. **II**. Shape-based bimorphism computed: Colored segments in Vase 1 and Vase 2 are properly mapped.

A shape-based correspondence between C_1 and C_2 is a bimorphism that minimizes the energy:

$$\widehat{E}_{C_1,C_2}[\hat{f}_1, \hat{f}_2, \Theta] = \int_{[\upsilon]} |\boldsymbol{v}_1(\hat{f}_1) - \mathrm{R}_\Theta \boldsymbol{v}_2(\hat{f}_2)|^2 ds_\upsilon \tag{15}$$

where Θ is a locally constant function.

We now construct a search space for the diffeomorphisms $\hat{f}_i$. Each curve C_i is first mapped to a topological representative $C_{0,i}$(a disjoint union of circles). Let $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ be any two points arbitrarily chosen, one on each of the components of $C_{0,2}$. Denote $\widetilde{C}_{0,2} = C_{0,2} - \{\boldsymbol{q}_1, \boldsymbol{q}_2\}$. Let $u_1 : \widetilde{C} \to C_{0,1}$ be a diffeomorphism to its image and $u_2 : \widetilde{C} \to \widetilde{C}_{0,2}$, a diffeomorphism. The maps u_i are also chosen such that they preserve the orientation of $C_{0,i}$.

Now, any diffeomorphism $\hat{f}_i : \widetilde{C} \to \widetilde{C}_i$ can be computed as $\hat{f}_i = \bar{f}_i \circ u_i$, where $\bar{f}_i : C_{0,i} \to C_i$ is a diffeomorphism.(Fig.3). Further, the set of minimizing bimorphisms for (17) if nonempty, is invariant with respect to the choice of $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$. Hence, these points can be picked in any convenient manner.

Our search for maps $\hat{f}_i$ is simplified to the search for diffeomorphisms, $\bar{f}_i$. But, we already have a framework to construct the space of diffeomorphisms $\bar{f}_i : C_{0,i} \to C_i$, that is to decompose $\bar{f}_i$ as $\bar{f}_i = f_{0,i} \circ f_i$, where $f_{0,i} : C_{0,i} \to C_i$ are initial maps and $f_i : C_{0,i} \to C_{0,i}$ are time-τ flows of tangential vector-fields $X_i = a_i \widetilde{T}_i$ on $C_{0,i}$. The functions a_i, chosen to be at least Lipschitz, then model the search space for $\bar{f}_i$.

Similar to previous cases, let $\mathrm{N}_\epsilon(C), \mathrm{N}_{\epsilon_{0,i}}(C_{0,i})$, be the corresponding neighborhoods of C and $C_{0,i}$ respectively. Let $\Omega_{\epsilon_0} = \Omega - (\mathrm{N}_{\epsilon_0}(\boldsymbol{p}_1) \bigcup \mathrm{N}_{\epsilon_0}(\boldsymbol{p}_2))$, where $\mathrm{N}_{\epsilon_0}(\boldsymbol{p}_i)$ is a small neighborhood of $\boldsymbol{p}_i$. The energy (17) can be written as a mini-

mization problem over the space of Lipschitz functions a_i defined over $\mathrm{N}_{\epsilon_{0,i}}(C_{0,i})$ and locally constant Θ defined over $\mathrm{N}_{\epsilon_{0,2}}(C_{0,2})$:

$$E_2(a_1, a_2, \Theta) = \int_{\Omega_{\epsilon_0}} \delta(\Phi)|\boldsymbol{v}_1(f_{0,1} \circ f_1(u_1(\boldsymbol{x}); a_1)) - \mathrm{R}_{\Theta}\boldsymbol{v}_2(f_{0,2} \circ f_2(u_2(\boldsymbol{x}); a_2))|^2 \\ |F_x \wedge F_y| d\boldsymbol{x}$$

where, $F = [\frac{f_{0,1} \circ f_1 \circ u_1}{\mathrm{L}_{C_1}} \frac{f_{0,2} \circ f_2 \circ u_2}{\mathrm{L}_{C_2}}]$.

8 Results and Applications

The first experiment was on synthetic 'shark' data. Given the signed distance functions for the two 'shark' curves C_1 and C_2, we use algorithm (16) to compute a shape-based bimorphism $\upsilon : C \to C_1 \times C_2$, $\upsilon = [\hat{f}_1, \hat{f}_2]$. between the curves (see Eqn.(3)). To visualize the bimorphism υ, the circle C is partitioned into segments, each of a different color. Each segment's image under the bimorphism is indicated by segments of the same color, on the curves C_1 and C_2 Fig.4. In Fig.5, we see two 'vase' curves being matched. We have chosen more colored segments here to show the bimorphism. In the ideal case when the two curves differ only by rigid motions or local stretching, the shape-matching is accurate for any resolution of the segments chosen. In Fig.6, the model shows capability to recover large non-rigid deformations.

In Fig.7, preliminary results on synthetic data, for matching/tracking curves of arbitrary topology, are shown. In (I), the curve on the left splits into two

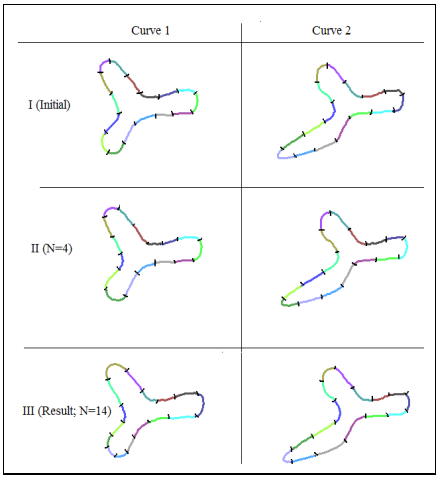


Fig. 6. *Large-deformation test.* **I**. Initial correspondence. **II**. Correspondence at N=4. **III**. Shape-based bimorphism computed.

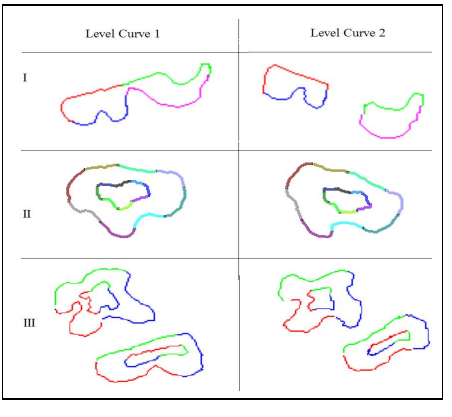


Fig. 7. *Matching level curves of arbitrary topology* **I**. Tracking splitting curves. **II** and **III**. Multiple Curve matching.

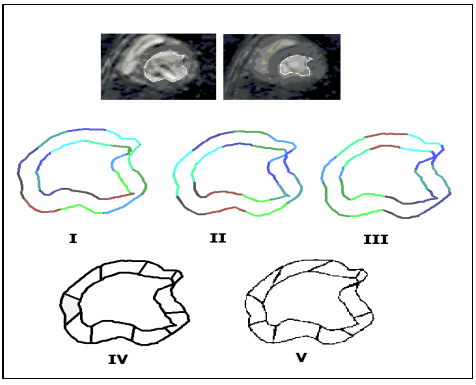


Fig. 8. *Tagare's MRI Heart Data* **I**. Initial correspondence. **II**. Correspondence computed by (16) after N=1 step. **III**. Final non-Rigid bimorphism computed by (16). **IV**. The same bimorphism as in (III), linking the sample points used on the outer curve by Tagare to the corresponding points on the inner curve. **V** Tagare's result.

curves. Colored segments of similar shapes are tracked before and after the splitting, using (17). (II) and (III) show matching of multiple curves of the same topology.

In Fig.8, we tested on Tagare's data (available in his website *http:// noodle.med.yale.edu /hdtag/*) of the endocardium of a dog's heart. The curves are the endocardial boundaries at end-systole and end-diastole. Picture (V) is the result computed by Tagare's algorithm for parametric bimorphisms. We used the same sample points as seen in (V) for the outer curve and find the corresponding points on the inner curve. (see IV). To minimize (6), we start with N=1 in (16) with c=0,d=0,$\theta = 0, \xi = .4, \epsilon_1 = 1$. We found stable convergence for N=16 in 35 iterations.

9 Comparison

Table.1 shows a comparison of the model's capabilities with some of the existing implicit matching methods.

Unlike existing methods that are suitable for curve-matching, our model is applicable to both shape-matching and to motion-tracking. By definition, the problem is to find diffeomorphic correspondences between given two 0-level curves. The input curves C_1 and C_2 are required to be at least Lipchitz continuous and we do not handle the case of data with outliers.

Except for Paragios' method, the other methods propose a large diffeomorphism setting using flows of suitable vector-fields on the entire image domain. However we search for diffeomorphisms just between the curves, thus resulting in a smaller search space for the vector fields. Further, our use of a shape-based matching criterion(i.e.normals) makes the energy invariant under scaling and rotation of the curves C_1 and C_2.

Table 1. Comparisons of our method with several current implicit matching methods

Features/Methods	**Our Method**	**Paragios**[23]	**Guo** [10]	**Glaunes**[8]
Applications	Mot. Trk, Shp. Mat.	Shp. Match.	Shp. Match.	Shp. Match.
Outlier Reject?	No	Yes	Yes	Yes
Shape Based?	Yes	No	No	No
Large defm.?	Yes	No	Yes	Yes
Search Space	curve Diffeos	2D Smooth trans.	2D Diffeos	2D Diffeos
Symmetry ?	Yes	No	No	No
Invariance	Scale/Rotation	Scale/Rotation	Rotation	Rotation
Topology	Arbitrary	Same	Same	Same

Also our energy functional is minimized over a "lifted space" of bimorphisms, a set of curves on a torus $C_1 \times C_2$, satisfying certain conditions. This makes the resulting correspondences symmetric under interchange of C_1 and C_2.

Lastly, unlike the mentioned methods, we have a natural way of matching/tracking curves of arbitrary topology.

References

1. Bakircioglu, M. , Joshi, S. and Miller, M. Landmark matching on brain surfaces via large deformation diffeomorphisms on the sphere, *Proc. of SPIE Med. Imag. Proc.*, no. 3661, ,pp.710-715, July 1999.
2. Bookstein, F. L. Morphometric Tools for Landmark data, *Cambridge Univ.Press*, New York,1991
3. Camion, V. and Younes, L. Geodesic interpolating splines, In *Proc. of EMM-CVPR01*, vol 2134 of Lecture Notes in Comp.Sc., pp.513527. Springer, 2001.
4. Chen, Y. ,Thiruvenkadam, S. , Gopinath, K. , Briggs, R. FMR Image registration Using the Mumford-Shah Functional and Shape Information, *World Multiconference on Systems, Cybernetics and Informatics*, pp.580-583, 2002.
5. Clarysee, P. and Friboulet, D. Tracking geometrical descriptors on 3-D deformable surfaces:Application to the left-ventricular surface of the heart,*IEEE Trans. Med.Imag.*, vol. 16, pp. 392-404, Aug. 1997.
6. Daatzikos, C. and Bryan, N. R. 1996. Using a deformable surface model to obtain a shape representation of the cortex,*IEEETrans.Med.Imag.*vol.15, pp.785-795.
7. Dupuis, P. , Grenander, U. and Miller, M. 1998. A variational formulation of a problem in image matching, *Quarterly of Applied Mathematics*,no. 56, pp. 587-600.
8. Glaunes, J. , Trouve, A. and Joshi, S. 2004. Diffeomorphic matching of distributions:A new approach for unlabelled point-sets and sub-manifolds matching, *Proc.IEEE CVPR 2004*, vol. 2, pp.712-718.
9. Glaunes, J. , Vaillant, M. and Osher, S. 2004. Landmark Matching via Large Deformation Diffeomorphisms on the Sphere, *J.Math.Img and Vision*, vol. 20, pp.179-200.
10. Guo, H. , Rangarajan, A. and Twining, J. C. 2004. Non-rigid registration of shapes via diffeomorphic point matching, *IEEE International Symposium on Biomedical Imaging: Macro to Nano*, pp.924 - 927.

11. Gwydir, S. H. and Buettner, M. H. 1994. Non-Rigid motion analysis and feature labelling of the growth cone, *Proc.IEEE workshop Biomed. Img. Anal.*, pp. 80 - 87.
12. Huang, X. , Paragios, N. and Younes, L. 2003. Registration of Structures in Arbitrary Dimension: Implicit Representations, Mutual Information & Free Form Deformations, *Technical Report DCS-TR-520*, Division of Computer and Information Sciences, Rutgers University, New Brunswick, April. 2003.
13. Joshi, S. C. , Miller, I. M. and Miller, I. M. 2000. Landmark matching via large deformation diffeomorphisms, *IEEE Trans. on Img.Proc.*, vol. 9, no.8.,pp.1357-1370.
14. Kambhamettu, C. and Goldgof, B. D. 2003. 3D Nonrigid Motion Analysis under Small Deformations, *Image and Vision Computing J*, vol. 21, no. 3, pp. 229-245.
15. Laskov, P. and Kambhamettu, C. 2003. Curvature-Based Algorithms for Nonrigid Motion and Correspondence Estimation, *IEEE Trans. PAMI*, vol. 25, no.10, pp.1349-1354.
16. Loncaric, S. 1998. A survey of shape analysis techniques, *Pattern Recognit.*, vol.31, no.8, pp.983-1001.
17. Maintz, J. and Viergever, M. 1998. A survey for Medical Image Registration.*Medical Image Analysis*, vol. 2, pp. 1-36.
18. Marsl, S. and Twining, C.J. 2004. Constructing diffeomorphic representations for the groupwise analysis of nonrigid registrations of medical images, *IEEE Trans. on Med.Imag.*, vol 23 ,no. 8 , pp.1006 - 1020
19. Matheny, A. and Goldgof, B. D. 1995. Use of three and four dimensional surface harmonics for rigid and non-rigid shape recovery and representation,*IEEE. Trans. PAMI*, vol.17, pp.967-981.
20. Meier, D. and Fisher, E. 2002. Parameter Space Warping: Shape-Based Correspondence Between Morphologically Different Objects *IEEE.Trans. on Med.Imag*, vol. 21, no. 1, pp. 31-47.
21. Memoli, F., Sapiro, G. and Osher, S.J., D. 2004. Solving Variational problems and Partial differential equations mapping into general target manifolds, *J.Comp.Phys*, 195, pp.263-292.
22. Osher, J.S. and Fedkiw, R. 2003. Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag, New York.
23. Paragios, N., Rousson, M. and Ramesh, V. 2003. Non-rigid registration using distance functions. *Computer Vision and Understanding*, vol.89, pp. 142-165.
24. Rousson, M. and Paragios, N. 2002. Shape priors for level set representations, *Proceedings of Computer Vision-ECCV2002*, pp. 7892.
25. Tagare, H.D., O'Shea, D. and Dunn, M.S. 2002. Non-Rigid Shape Comparison of Plane curves in Images, *J.Math.Img and Vision.*, vol. 16, pp. 57-68.
26. Thompson, P. and Toga, A. 1996. Surface-Based technique for warping three-dimensional images of the brain,*IEEE Trans.Med.Imag*, vol.15, pp.402-417.
27. Veltkamp, R. and Hagedoorn, M. 1999. State-of-the-art in Shape Matching, *Tech. rep. UU-CS-1999-27,Utrecht University.*

Incorporating Rigid Structures in Non-rigid Registration Using Triangular B-Splines

Kexiang Wang, Ying He, and Hong Qin

Department of Computer Science,
State University of New York at Stony Brook,
Stony Brook, NY 11790-4400, USA
{kwang, yhe, qin}@cs.sunysb.edu

Abstract. For non-rigid registration, the objects in medical images are usually treated as a single deformable body with homogeneous stiffness distribution. However, this assumption is invalid for certain parts of the human body, where bony structures move rigidly, while the others may deform. In this paper, we introduce a novel registration technique that models local rigidity of pre-identified rigid structures as well as global non-rigidity in the transformation field using triangular B-splines. In contrast to the conventional registration method based on tensor-product B-splines, our approach recovers local rigid transformation with fewer degrees of freedom (DOFs), and accurately simulates sharp features (C^0 continuity) along the interface between deformable regions and rigid structures, because of the unique advantages offered by triangular B-splines, such as flexible triangular domain, local control and space-varying smoothness modeling. The accurate matching of the source image with the target one is accomplished through the use of a variational framework, in which a composite energy, measuring the image dissimilarity and enforcing local rigidity and global smoothness, is minimized subject to pre-defined point-based constraints. The algorithm is tested on both synthetic and real 2D images for its applicability. The experimental results show that, by accurately modeling sharp features using triangular B-splines, the deformable regions in the vicinity of rigid structures are less constrained by the global smoothness regularization and therefore contribute extra flexibility to the optimization process. Consequently, the registration quality is improved considerably.

1 Introduction

For the last decade, image registration has become an important technique for various computer vision and medical applications, fusing the information from images acquired either at different times or on multiple modalities. A number of reviews have been documented in [1][2][3]. The earliest attempts made by [4][5] typically restrict the deformation between the corresponding images to be rigid and consider global geometric differences only. Later, non-rigid registration was introduced in [6][7] to additionally cope with local differences, resulting from different anatomy, intraoperative deformation, or distortion induced during imaging process. It is often assumed by the non-rigid registration that the objects in the matching images behave as if they were a single elastic body, *i.e.*, the stiffness is constant everywhere. However, this is rarely the case

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 235–246, 2005.

when the imaged anatomy contains both rigid and soft structures. A practical clinical example was described in [8] where the shape of the brain changed after subdural electrodes were implanted in a surgical procedure. Neither a single rigid body motion nor a nonlinear model with invariant smoothness can accurately represent the transformation between preoperative and postoperative scans since the electrodes translate and rotate only, while the others deform nonlinearly. Therefore, more appropriate methods are required to combine the modeling of both rigidity and non-rigidity in the recovered transformation. Especially, the C^0 continuity on the borders of rigid structures needs to be simulated correctly for precise registration.

In principle, we could build patient-specific physical model to predict the interaction between rigid structures and soft tissues. However, it is impractical to achieve solution with desired accuracy due to high computational cost and insufficient details on mass, elasticity, and other mechanical properties. The efforts made so far were either based on interpolatory spline scheme or through a variational framework. Little *et al.*[9] incorporated independent rigid objects in a modified thin-plate spline (TPS) based nonrigid registration. Anisotropic landmarks were introduced by Rohr *et al.*[10] to TPS to enforce local rigidity constraints. Duay *et al.*[11] simulated the rigid motions by adaptively adjusting TPS radial basis functions according to local stiffness. Tanner *et al.*[12] represented the deformation using B-splines and locally couple control points in order to model local rigidities. Most recently, Loeckx *et al.*[13] introduce a penalty term to keep voxel-based rigidities in their variational framework by enforcing the orthogonality of Jacobian matrix. Nevertheless, none of the above approaches, except that in [9], can precisely describe C^0 continuity in the displacement field. In spite of the attempt made in [14], it's not straightforward for thin-plate splines to be incorporated with variational framework, which is quite a powerful tool for intensity-based image registration. On the other hand, tensor-product B-splines has been widely used for optimization-based registration approaches [7][15][16]. Although it is possible for tensor-product B-splines to describe sharp features when the corresponding knots collapse, such features can not lie in arbitrary direction due to the regular domain of B-splines.

In this paper, we propose a novel non-rigid registration algorithm in which the recovered deformation field is represented by triangular B-splines. We first build the domain triangulation and adjust corresponding knots to the boundaries of pre-segmented rigid structures. As a result, the C^0 continuity is guaranteed at the desired places in the displacement field. The landmarks, selected at the vicinities of rigid objects, are brought into correspondence between source and target images as point-based constraints. The optimal transformation is then estimated by minimizing a composite energy function, which measures image discrepancy, deformation distortion, and desired local rigidities. Empowered by the numerous advantages of triangular B-splines, such as flexible domain, local control, space-varying smoothness modeling, etc., our registration approach makes the following contributions: The local linear motion in the global non-rigid transformation, caused by rigid structures, can be accurately recovered using relatively fewer degrees of freedom (DOFs), as long as the feature lines are properly aligned in the domain triangulation. With C^0 continuity modeled at the interface between rigid and non-rigid objects, the deformable region nearby can move more freely and tend to improve the registration quality considerably.

2 Theory and Construction of Triangular B-Splines

Triangular B-splines, introduced by Dahmen, Micchelli and Seidel[17], have numerous desirable properties for geometric design, such as their automatic smoothness, the ability to define a surface over arbitrary triangulation, and modeling sharp features between any desired adjacent primary knots [18]. Pfeifle and Seidel[19] presented an efficient algorithm to evaluate quadratic triangular B-splines and Franssen *et al.*[20] extended it to triangular B-splines of arbitrary degree.

The construction of triangular B-spline is as follows: let points $\mathbf{t}_i \in \mathbb{R}^2$, $i \in \mathbb{N}$, be given and define a triangulation

$$T = \{\Delta(I) = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] : I = (i_0, i_1, i_2) \in \mathcal{I} \subset \mathbb{N}^2\}$$

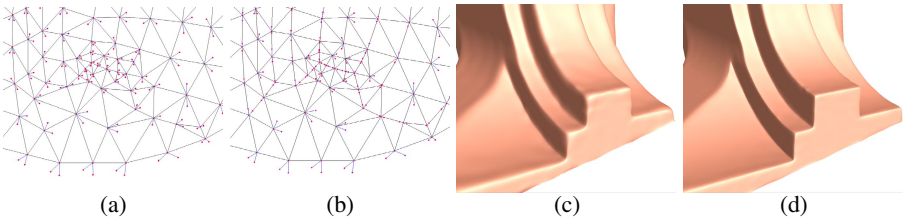
of a bounded region $D \subseteq \mathbb{R}^2$. Next, with every vertex $\mathbf{t}_i$ of T we associate a cloud of knots $\mathbf{t}_{i,0}, \ldots, \mathbf{t}_{i,n}$ such that $\mathbf{t}_{i,0} = \mathbf{t}_i$. The knots $\mathbf{t}_{i,0}|i \in \mathbb{N}$ are called primary knots, and $\mathbf{t}_{i,j}|i \in \mathbb{N}, j \geq 1$ are called sub-knots. For every triangle $I = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] \in T$,

1. all the triangles $[\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}]$ with $\beta = (\beta_0, \beta_1, \beta_2)$ and $|\beta| = \sum_{i=0}^{2} \beta_i \leq n$ are non-degenerate.
2. the set
$$interior(\cap_{|\beta| \leq n} X_\beta^I) \neq \emptyset, X_\beta^I = [\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}]. \quad (1)$$
3. If I has a boundary edge, say, $(\mathbf{t}_{i_0}, \mathbf{t}_{i_1})$, then the entire area $[\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,n}, \mathbf{t}_{i_1,0}, \ldots, \mathbf{t}_{i_1,n})$ must lie outside of the domain.

Then the triangular B-spline basis function N_β^I, $|\beta| = n$, is defined by means of simplex splines $M(\mathbf{u}|V_\beta^I)$ (for details about simplex splines, please refer to [21]) as

$$N(\mathbf{u}|V_\beta^I) = |d_\beta^I| M(\mathbf{u}|V_\beta^I),$$

where $V_\beta^I = \{\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,\beta_0}, \ldots, \mathbf{t}_{i_2,0}, \ldots, \mathbf{t}_{i_2,\beta_2}\}$ and



(a) (b) (c) (d)

Fig. 1. Modeling sharp features using triangular B-spline with degenerate knots. (a) The domain triangulation and regular knot configurations (no three knots in a domain triangle are collinear). (b) Place the sub-knots along the user-specified edges of domain triangulation. (c) A cubic spline surface reconstructed using the knot configurations in (a). The spline is C^2 continuous everywhere. (d) A cubic spline surface reconstructed using the knot configurations in (b). The spline is C^2 continuous on smooth regions and C^0 on sharp features. appropriately, we can model.

$$d_\beta^I = det(X_\beta^I) = det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{t}_{i_0,\beta_0} & \mathbf{t}_{i_1,\beta_1} & \mathbf{t}_{i_2,\beta_2} \end{pmatrix}.$$

Assuming (1), these B-spline basis functions can be shown to be all non-negative and to form a partition of unity. Then, the triangular B-spline is defined as

$$\mathbf{F}(\mathbf{u}) = \sum_{I \in \mathcal{I}} \sum_{|\beta|=n} \mathbf{c}_{I,\beta} N(\mathbf{u}|V_\beta^I), \tag{2}$$

where $\mathbf{c}_{I,\beta}$ is the control point. This spline is globally C^{n-1} continuous if all the sets X_β^I, $|\beta| \leq n$ are affinely independent.

One favorable advantage given by triangular B-splines is that by adjusting sub-knots to the feature lines explicitly, we can model local sharp features (*i.e.*, C^0 continuity) in the approximated space, while keeping the C^{n-1} smoothness over the other regions. This feature is demonstrated in a surface reconstruction example shown in Fig.1.

3 Method

Given source image I_s, and target image, I_t, defined on the domain $\Omega \subset \mathbb{R}^2$, the problem of registration is to find an optimal geometrical transformation $\mathbf{T} : \Omega \rightarrow \mathbb{R}^2$ such that the pixels in both images are matched properly.

3.1 Transformation Model

To reduce the global geometric differences between I_s and I_t, an initial alignment is achieved using conventional rigid registration algorithm. This obtained transformation will be used as the initial estimation for the following registration.

The concept of free-form deformation (FFD) is to deform an object by manipulation underlying control points. In our work, the FFD is decomposed as an identity transformation plus a displacement field, which is represented by triangular B-splines as:

$$\mathbf{T}(\mathbf{x}) = \mathbf{x} + \mathbf{u} = \mathbf{x} + \sum_{i=1..m} \phi_i B_i(\mathbf{x}), \tag{3}$$

where ϕ_i is the control point and B_i is the associated basis function.

Unlike tensor-product B-spline based FFD[15], whose domain is a rectangular lattice, our triangular B-spline based FFD has its domain built upon a tessellation of either triangles for 2D or tetrahedra for 3D. It is not difficult to triangulate the reference image domain Ω using established techniques. In order to model the sharp features (see Fig.3(c)) at the boundaries of pre-identified rigid bodies, we have to keep them in the triangulated tessellation. Such constraints can be satisfied using the triangulation algorithm proposed by Shewchuk[22]. According to the definition of triangular B-splines, the free-form deformation field has C^{n-1} continuity everywhere if there is no degeneracy for any triple of knots in the same triangle. However, we purposely collapse adjacent sub-knots to pre-identified feature lines in order to model desired C^0 continuity.

Due to the flexibility of the domain triangulation, it is also possible for users to overlay the registration domain exactly upon the region of interest (ROI), rather than covering the entire reference image. Thus, the computational effort will be saved considerably, especially when the ROI can be successfully extracted.

3.2 Point-Based Constraints

Point-based constraints are incorporated in our framework for better registration. The points on the boundary contours of rigid structure with high curvature are good candidates for landmarks (see Fig.3(d)). Assuming strict rigidity of bony structures, only two pairs of landmarks are required to fully recover local linear transformation (*i.e.*,translation and rotation), if there is no rotoinversion. In practice, we often introduce more constraints to ensure the stability of the registration.

Let $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ be the set of landmarks chosen on the reference domain (I_t in our implementation). Their correspondences in I_s are $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ such that:

$$\mathbf{T}(\mathbf{p}_j; \Phi) = \mathbf{q}_j \quad \text{for} \quad j = 1 \ldots n, \tag{4}$$

where Φ denotes the set of the control points of triangular B-splines. The above equations are treated as hard constraints and have to be strictly satisfied in the following optimization process. In most cases, the linear system of (4) is under-determined. But it is possible to become over-constrained when excessive landmarks are selected on a single spline patch. Two approaches can be used to solve such problem. One is to subdivide the triangular mesh, where there are overly-condensed landmarks, and re-initialize the domain triangulation. The drawback of it is that the problem dimension is increased accordingly. The other approach aims to find a compromised solution for (4), which will be discussed later.

3.3 Cost Function

In this paper, we register I_s to I_t using a variational approach, in which a metric measuring image similarity and constraints of global smoothness and local rigidity are combined into an overall cost function E_{total} that is defined as:

$$E_{total} = \alpha E_I + \beta E_R + \gamma E_S, \tag{5}$$

where α, β, and γ control the relative influence among three energy terms. In (5), E_I is the driving force behind the registration process and aims to maximize the image similarity, whereas E_R is a constraint term to ensure local rigidity and E_S tries to regularize the transformation as smooth as possible.

A number of approaches have been proposed in literature to calculate either similarity or dissimilarity between images. Mutual information[23][24] and correlation ratio[25] are the methods to measure image similarities, while the sum-of-squared-difference (SSD) measures the dissimilarities. In our current work, we simply use SSD metric to test the feasibility of our registration algorithm. The differences between I_s and I_t, represented by E_I, is evaluated by:

$$E_I = \frac{1}{2} \int_{\Omega} \left\| I_s(\mathbf{T}(\mathbf{x}; \Phi)) - I_t(\mathbf{x}) \right\|^2 d\mathbf{x}. \tag{6}$$

In the theory of continuum physics, the non-rigid transformation is often measured by *Green-St. Venant* strain tensor $\mathbf{E}$. Then a necessary and sufficient condition to obtain a local rigid transformation can be re-formulated as $\mathbf{E} = \nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u} = 0$, which enforces the strain tensor $\mathbf{E}$ to be zero over rigid structures. This is identical to that proposed in [13], where the Jacobian matrices are considered instead. We enforce the rigidity constraint by defining a penalty term as the integral of the *Frobenius* norm of $\mathbf{E}$. Since different structures in the image exhibit different deformation properties, and do not need to deform similarly, we introduce a characteristic function $w(\mathbf{x})$ to separate the rigid objects from deformable regions. The value of $w(\mathbf{x})$ is 1 on rigid structures and 0 elsewhere. The penalty term for local rigidity is given by:

$$E_R = \frac{1}{2}\int_{\Omega} w(\mathbf{x}) \left\|\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u}\right\|_F^2 d\mathbf{x}, \tag{7}$$

where $\|\cdot\|_F$ denotes the *Frobenius* norm.

A regularization term E_S, measuring the bending energy of a thin plate metal subject to external forces[7][15], is also incorporated to discourage improbable or impossible transformations. It depends on the 2nd derivatives of the deformation and is written as:

$$E_S = \frac{1}{2}\int_{\Omega}(1 - w(\mathbf{x})) \left[\left\|\frac{\partial^2\mathbf{u}}{\partial x^2}\right\|^2 + \left\|\frac{\partial^2\mathbf{u}}{\partial y^2}\right\|^2 + \left\|\frac{\partial^2\mathbf{u}}{\partial z^2}\right\|^2 + 2\left(\left\|\frac{\partial^2\mathbf{u}}{\partial x\partial y}\right\|^2 + \left\|\frac{\partial^2\mathbf{u}}{\partial y\partial z}\right\|^2 + \left\|\frac{\partial^2\mathbf{u}}{\partial x\partial z}\right\|^2\right)\right] d\mathbf{x}, \tag{8}$$

where the function $w(\mathbf{x})$ makes the regularization term valid only over non-rigid regions.

3.4 Optimization

The optimization problem is stated to find an ideal $\mathbf{\Phi}$ such that the overall energy (5) is minimized with the constraints in (4) satisfied. There are various algorithms available to accomplish such constrained nonlinear programming task. In particular, we convert the constrained optimization problem to a unconstrained one, rather than applying established methods directly. Then a simplex line search approach described in [26] is performed to update the parameters iteratively along the steepest descent of gradient until the cost function can not be decreased any further.

Putting (3) and (4) together, we discretize the point-based constraints and write them in a matrix format:

$$\mathbf{P} + \mathbf{C}\mathbf{\Phi} = \mathbf{Q}, \tag{9}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are the vectors collecting the landmark positions in I_t and I_s respectively, the vector $\mathbf{\Phi}$ consists of the control points of triangular B-splines, and the triangular B-spline basis functions constitute the matrix $\mathbf{C}$ which is extremely sparse and rank-deficient.

By solving the original optimization problem in the Null-Space of $\mathbf{C}$, we can successfully remove the point-based constraints. Then, the new parameter vector $\mathbf{\Psi}$ in Null-Space is related to the old one $\mathbf{\Phi}$ by the equation:

$$\mathbf{\Phi} = \mathbf{N}\mathbf{\Psi} + \mathbf{\Phi}_0, \tag{10}$$

in which $\mathbf{CN} = 0$ and $\mathbf{C}\mathbf{\Phi_0} = \mathbf{Q} - \mathbf{P}$. We use Gaussian-Jordan-Elimination-like approach proposed in [27] to construct $\mathbf{N}$, and solve for $\mathbf{\Phi_0}$ by either singular value decomposition (SVD) or QR decomposition, both of which are computationally viable here, since most columns in $\mathbf{C}$ are zero.

Instead of estimating the gradient of E_{total} using finite-difference approximation, we analytically calculate the derivative with respect to $\mathbf{\Psi}$ and obtain:

$$\frac{\partial E_{total}}{\partial \mathbf{\Psi}} = \mathbf{N^T}\left(\alpha\frac{\partial E_I}{\partial \mathbf{\Phi}} + \beta\frac{\partial E_R}{\partial \mathbf{\Phi}} + \gamma\frac{\partial E_S}{\partial \mathbf{\Phi}}\right), \tag{11}$$

where

$$\frac{E_I}{\partial \phi_i} = \int_\Omega \left(I_s(\mathbf{T}(\mathbf{x})) - I_t(\mathbf{x})\right) \nabla I_s \mid_{\mathbf{T(x)}} B_i(\mathbf{x}) d\mathbf{x}.$$

Let ϕ_{ij} denotes the $(3i + j)$-th component of $\mathbf{\Phi}$ and $B_{i,j}$ be the derivative of the basis function in j direction, where $j = 1, 2, 3$ for x, y and z coordinates, respectively. The derivative of the local rigidity penalty term is:

$$\frac{E_R}{\partial \phi_{ij}} = \int_\Omega w(\mathbf{x}) \sum_{s,t=1,2,3} \mathbf{M}_{st} \frac{\partial \mathbf{M}_{st}}{\partial \phi_{ij}} d\mathbf{x}$$

$$\mathbf{M}_{st} = \sum_{\substack{k=1\ldots m \\ r=1,2,3}} \phi_{kt} B_{k,s} + \phi_{ks} B_{k,t} + \phi_{ks}\phi_{kt} B_{k,r}^2$$

$$\frac{\partial \mathbf{M}_{st}}{\partial \phi_{ij}} = \sum_{r=1,2,3} \delta_{jt}\left(B_{i,s} + \phi_{is} B_{i,r}^2\right) + \delta_{js}\left(B_{i,t} + \phi_{it} B_{i,r}^2\right)$$

in which δ_{ij} is *Dirac* function which equals to 1 if and only if $i = j$. Likewise, the derivative of the regularization term is given by:

$$\frac{E_S}{\partial \phi_{ij}} = \int_\Omega (1 - w(\mathbf{x})) \left(\sum_{s,k=1,2,3} \phi_{kj} B_{k,s} B_{i,s} + \sum_{s,t,k=1,2,3} \phi_{k,j} B_{k,st} B_{i,st} \right) d\mathbf{x},$$

where $B_{i,st}$ stands for the second derivative of basis functions. For details on efficient evaluation of triangular B-spline basis functions and their derivatives, please refer to [19][20][28].

Note that the integration operation in (5) is performed only on the pixels of ROI. Therefore, we could significantly speedup the registration procedure if all the basis functions and their derivatives over the interested region are pre-computed.

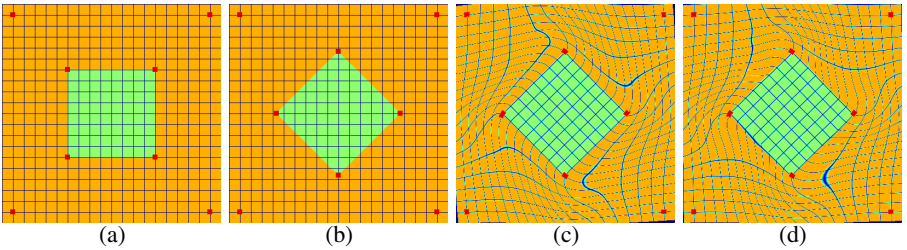


Fig. 2. The first experiment: (a) Source image. (b) Target image. (c) Registration result when sharp features are NOT modeled. The minimized energy terms are $E_S = 97.6$, $E_R = 39.9$. (d) Registration result when sharp features are modeled, the minimized energy terms are $E_S = 71.8$, $E_R = 30.8$.

4 Experimental Results

In order to evaluate the feasibility and applicability of the proposed algorithm, we test it on both synthetic and real data. Cubic triangular B-splines are chosen in the experiments to compare with the frequently used cubic tensor-product B-splines.

The first example demonstrated in Fig.2 doesn't consider matching image intensities(*i.e.*, $\alpha = 0$ in (5)), but tries to align corresponding points instead. A green square is included in the source image to represent a rigid object, and its counterpart is included in the target image with a rotation of $45°$. 8 pairs of landmarks are selected at the corners of both the image and the rigid square, and applied as the point-based constraints in the registration. The target image is chosen as the reference domain, which is triangulated into 32 patches, and the cubic triangular B-splines built on it have 361 control points. After applying our algorithm without and with sharp feature modeling respectively, the achieved registration results are plotted in Fig.2(c) and Fig.2(d). It is noticeable that the background and the square are more smoothly connected in Fig.2(c) than in Fig.2(d), because they are treated as a single elastic object in the former one, but considered as separate parts in the latter one. It is more physically appropriate to model C^0 continuity between the background and the square, when we simulate the interaction between them. Therefore, the method with sharp feature modeling can achieve better registration result (the minimized energy terms are $E_S = 71.8$, $E_R = 30.8$) than the other one ($E_S = 97.6$, $E_R = 39.9$), when the same parameter setting ($\beta = \gamma = 1$) is used.

For the second example, both images (see Fig.3(a)(e)) include three geometric objects to represent rigid structures, whose positions are quite different in the source and the target images. The reference domain (shown in Fig.3(b)) has 130 triangles and the triangular B-splines thus have 631 control points. 13 pairs of landmarks are picked up to ensure correct alignment between rigid structures (see Fig.3(f)). The registration result and the recovered deformation field are shown in Fig.3(g) and Fig.3(h). An alternative approach using tensor-product B-splines is also applied for the comparison purpose. Its domain is defined on a 25×25 to match the number of triangular B-spline control points. The comparison between the results from both approaches (shown in Fig.3(c) and Fig.3(g)) indicates that the tensor-product based method fails to align the images at a desired resolution, when there exist large deformations near rigid structures. In sharp

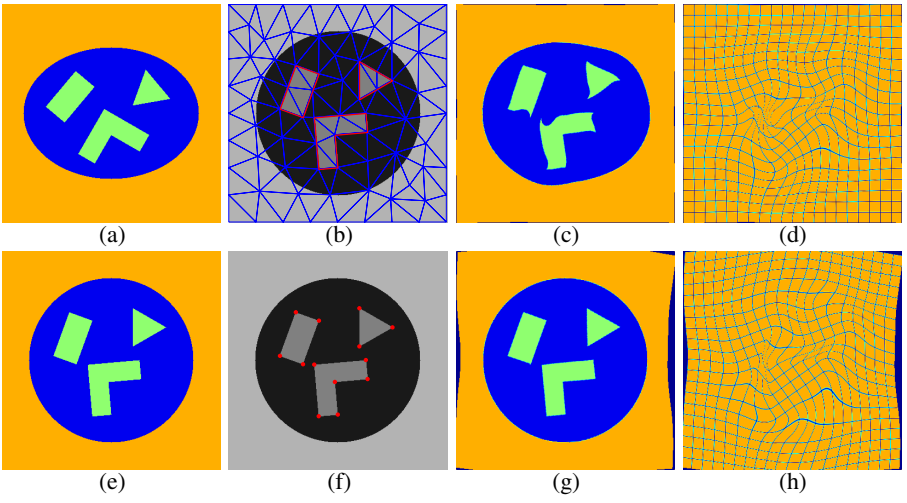


Fig. 3. The second experiment: (a) Source image. (e) Target image. (b) The domain triangulation with feature lines highlighted in red. (f) 13 landmarks are highlighted in red. (c) Registration result obtained from tensor-product B-spline based method. (g) Registration result obtained from triangular B-spline based method. (d) Deformation recovered using tensor-product B-spline based method. (h) Deformation recovered using triangular B-spline based method.

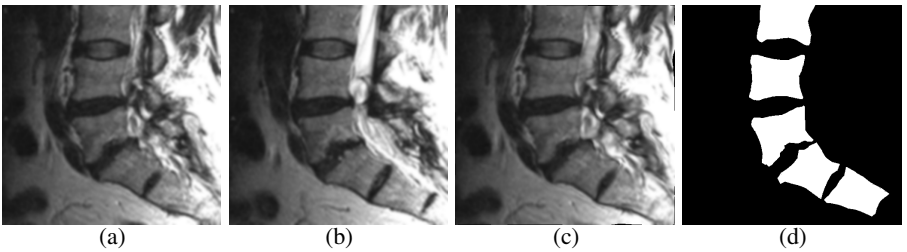
contrast, triangular B-spline is built on a flexible domain, so that its modeling power can be ideally concentrated on the interested region for better registration. Furthermore, its power of modeling sharp features helps to improve the registration quality far more better.

Two MRI images of human spines (see Fig.4(a)(b)) are used in the third experiment. The spinal bones are first segmented from the target image, then the characteristic function w is set accordingly to decide where the rigidity constraints should be applied. The source image is registered to the target image as shown in Fig.4, in which all of the rigid structures are successfully matched.

Our algorithm is implemented using MS VC++, and all experiments are conducted on a platform with 2.8GHz Pentium IV CPU and 1G RAM. Both synthetic images have the size of 400×400, and the size of the MRI images used for the third experiment is 512×512. The running time for the three experiments are about 1 minute, 6 minutes, and 12 minutes respectively.

5 Discussion and Conclusion

This paper presents a nonrigid registration technique in which the transformation between corresponding images are represented by triangular B-splines. By preserving feature lines in the domain triangulation and adjusting knots accordingly, the proposed method successfully recovers local rigid motions and accurately simulates C^0 continuities at desired regions, using relatively fewer degrees of freedom and lower degree



(a) (b) (c) (d)

Fig. 4. The third experiment: (a) Source image. (b) Target image. (c) Registration result. (d) Rigid structures segmented from the target image.

polynomials. The actual registration is done through the use of a variational framework, in which a constrained optimization problem is solved to reduce the differences between images and enforce both local rigidity and global smoothness at the same time. The method has been tested on both synthetic examples and real data for its efficacy.

Although tensor-product B-spline based approaches are still dominating in the field of non-rigid registration, their applicability is somehow limited due to the structure of their regular domain. On the contrary, our registration method can correctly delineate the boundaries of rigid bodies in its domain triangulation at a much coarser level, and thus model the local rigid motions more accurately. Furthermore, with the degenerate knots on the boundaries of rigid structures, C^0 continuity is automatically guaranteed in the described displacement field, and can be naturally coupled with the optimization process. This advantage enables us to precisely simulate the behavior of rigid objects inside elastic tissues. From the registration point of view, the deformable regions around the rigid structures may become less constrained by the regularization term and contribute extra flexibility to the minimization of the cost function (5). As a result, the registration quality can be considerably improved. An alternative way to model C^0 continuities could be to separate rigid and non-rigid regions into different domain pieces. However, extra efforts must be spent to keep the overall transformation consistent across different pieces in a different hierarchy, and in general, the variational approaches over irregular domains in a hierarchical fashion have not been fully explored. In this paper, only rigid structures with simple geometric shapes are considered in our experiments for the feasibility test. To accommodate more complicated structures, we can subdivide the domain mesh adaptively along their boundaries until the desired accuracy is achieved. The landmarks applied in our registration are interactively selected by users based on their knowledge and subjectivity. Naturally, the registration result is affected by the quality of landmark selection.

There are a few possible extensions to our current work. Although only the registration of two dimensional images is considered in this paper, it is much more natural to extend it to the area of volumetric data registration, and trivariant tetrahedral B-splines shall be exploited. Alternative metrics measuring image similarities, such as mutual information and normalized correlation, can be incorporated into our variational framework to deal with multi-modality registration. One limitation of our current approach is the necessity for image segmentation and landmark selection prior to our registration procedure. It would be ideal to have an automated method to select landmarks, seg-

ment images into different pieces, and accurately match corresponding rigid structures in order to streamline the task of medical image processing.

Acknowledgements

This research was partially supported by the NSF grant ACI-0328930, the ITR grant IIS-0326388, and the Alfred P. Sloan Fellowship.

References

1. Brown, L.G.: A survey of image registration techniques. ACM Comput. Surv. **24** (1992) 325–376
2. Maintz, J., Viergever, M.: A survey of medical image registration. Medical Image Analysis **2** (1998) 1–36
3. Zitová, B., Flusser, J.: Image registration methods: a survey. Image Vision Comput. **21** (2003) 977–1000
4. Peters, T., Davey, B., Munger, P., Comeau, R., Evans, A., Olivier, A.: Three-dimensional multimodal image-guidance for neurosurgery. IEEE Transactions on medical imaging **15** (1996) 121–128
5. Potamianos, P., Davies, B., Hibberd, R.D.: Intraoperative registration for percutaneous surgery. In: First International Symposium on Medical Robotics and Computer Asisted Surgery. Volume 1. (1995) 98–105
6. Cachier, P., Mangin, J.F., Pennec, X., Rivière, D., Papadopoulos-Orfanos, D., Régis, J., Ayache, N.: Multisubject non-rigid registration of brain mri using intensity and geometric features. In: MICCAI. (2001) 734–742
7. Rueckert, D., Sonoda, L., Hayes, C., Hill, D.L., Leach, M.O., Hawkes, D.J.: Non-rigid registration using free-form deformations: Application to breast mr images. IEEE Trans. Med. Imaging **18** (1999) 712–721
8. Edwards, P.J., Hill, D.L.G., Little, J.A., Sahni, V.A.: Medical image registration incorporating deformations. In: BMVC. (1995) 691–699
9. Little, J.A., Hill, D.L.G., Hawkes, D.J.: Deformations incorporating rigid structures. Computer Vision And Image Understanding **66** (1997) 223–232
10. Rohr, K., Fornefett, M., Stiehl, H.S.: Spline-based elastic image registration: Integration of landmark errors and orientation attributes. Computer Vision And Image Understanding **90** (2003) 153–168
11. Duay, V., D'Haese, P.F., Li, R., Dawant, B.M.: Non-rigid registration algorithm with spatially varying stiffness properties. In: ISBI. (2004) 408–411
12. Tanner, C., Schnabel, J.A., Chung, D., Clarkson, M.J., Rueckert, D., Hill, D.L.G., Hawkes, D.J.: Volume and shape preservation of enhancing lesions when applying non-rigid registration to a time series of contrast enhancing mr breast images. In: MICCAI. (2000) 327–337
13. Loeckx, D., Maes, F., Vandermeulen, D., Suetens, P.: Nonrigid image registration using free-form deformations with a local rigidity constraint. In: MICCAI (1). (2004) 639–646
14. Meyer, C.R., Boes, J.L., Kim, B., Bland, P.H., Zasadny, K.R., Kison, P.V., Koral, K., Frey, K.A., Wahl, R.L.: Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin-plate spline warped geometric deformations. Med. Image Anal. **1** (1996/7) 195–206
15. Rohlfing, T., Jr., C.R.M., Bluemke, D.A., Jacobs, M.A.: An alternating-constraints algorithm for volume-preserving non-rigid registration of contrast-enhanced mr breast images. In: WBIR. (2003) 291–300

16. Vemuri, B.C., Huang, S., Sahni, S., Leonard, C.M., Mohr, C., Gilmore, R., Fitzsimmons, J.: An efficient motion estimator with application to medical image registration. Medical Image Analysis **2** (1998) 79–98
17. Dahmen, W., Micchelli, C.A., Seidel, H.P.: Blossoming begets b-spline built better by b-patches. Mathematics of Computation **59** (1992) 97–115
18. Fong, P., Seidel, H.P.: An implementation of triangular b-spline surfaces over arbitrary triangulations. Computer Aided Geometric Design **10** (1993) 267–275
19. Pfeifle, R., Seidel, H.P.: Faster evaluation of qudartic bivariate dms spline surfaces. In: Proceedings of Graphics Interface'94. (1994) 182–189
20. Franssen, M., Veltkamp, R.C., Wesselink, W.: Efficient evaluation of triangular b-spline surfaces. Computer Aided Geometric Design **17** (2000) 863–877
21. Micchelli, C.A.: On a numerically efficient method for computing multivariate b-splines. Multivariate approximation theory (**51**) 211–248
22. Shewchuk, J.R.: Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures. Med. Phys. **24** (1997) 25–35
23. Collignon, A.: Multi-modality medical image registration by maximization of mutual information. PhD thesis, Catholic University of Leuven, Leuven, Belgium (1998)
24. Viola, P.A.: Alignment by maximization of mutual information. PhD thesis, Massachusetts Institute of Technology, Boston, MA, USA (1995)
25. Roche, A., Malandain, G., Pennec, X., Ayache, N.: The correlation ratio as a new similarity measure for multimodal image registration. In: MICCAI. (1998) 1115–1124
26. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press (1992)
27. Gill, P., Murray, W., Wright, M.: Practical Optimization. Academic Press (1981)
28. He, Y., Qin, H.: Surface reconstruction with triangular b-splines. In: GMP. (2004) 279–290

Geodesic Image Interpolation: Parameterizing and Interpolating Spatiotemporal Images

Brian B. Avants, C.L. Epstein, and J.C. Gee

Depts. of Radiology and Mathematics,
University of Pennsylvania,
Philadelphia PA 19101

Abstract. We develop a practical, symmetric, data-driven formulation, geodesic image interpolation (GII), for interpolating images with respect to geometric and photometric variables. GII captures, in implementation, the desirable properties of symmetry that comes from the theory of diffeomorphisms and Grenander's computational anatomy (CA). Geodesic diffeomorphisms are a desirable transformation model as they provide a symmetric deforming path connecting images or a series of images. Once estimated, this geodesic may be used to (re)parameterize and interpolate image sets in approximation of continuous, deforming dynamic processes. One may then closely recover the original continuous signal from a few samples. The method, based on our work in symmetric diffeomorphic image registration, generalizes the concept of point set reparameterization to the case where point sets are replaced by image sets. This problem differs from point reparameterization in that a variational image correspondence problem must be solved before resampling. Our image reparameterization method is applied to solve similar problems to point reparameterization: dense interpolation, matching and simulation of dynamic processes are illustrated.

1 Introduction

Deformation-based interpolation between a pair of images is a common requirement in computer vision. Shape-based or deformable interpolation methods have been used for image morphing and animation [1,2], interpolating motion sequences [3,4], reconstructing medical image volumes [5] and in texture mapping [6]. Shape-based/deformable data-driven interpolation methods have historically been limited by small deformation assumptions about the correspondence [7,8,5,9] and, relatedly, asymmetry (which is not present in linear/cubic/sinc interpolation). Some methods require prior segmentations [8,5]. The small deformation approximation has limited application in real world problems where large deformations are often present. The sparse landmark assumption, as used in Adobe Photoshop©, is also limited in that it requires manual intervention, is asymmetric and uses only a few correspondence points. Asymmetry originates, in all these approaches, in the image registration or transformation models underlying the correspondence computation.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 247–258, 2005.

The need for image registration is the main factor that distinguishes shape or registration-based interpolation from intensity-only interpolation. A common problem with image registration (IR) algorithms, in general, is asymmetry. Registering image I to J does not produce the same correspondences as registering J to I. This is particularly important in medicine, where results of clinical studies influencing patient treatment [10,11,12] should not depend upon arbitrary decisions made in the image processing chain, nor should an algorithm need to be tuned due to its unintuitive directionality. The goal of geodesic image interpolation is to resolve these complications in a fundamental way.

Our symmetric shape and appearance interpolation algorithm is illustrated with an example in figure 1. It will satisfy the principles:

1. Continuity – the algorithm's output should vary continuously with its inputs (also a condition of well-posedness), here, a set of images.
2. Anonymity – the output should not vary with permutations of the inputs, i.e., if the order of the images changes.
3. Unanimity – if all the inputs are identical, the output is the input.

One can easily see that linear intensity only interpolation satisfies these conditions. Eckmann and Shmuel gave this set of axioms, originally, in the context of social choice [13,14], a problem domain in economics. However, these principles apply naturally to image registration and interpolation, reflecting the assumption that a single "true" solution to the problem exists and that it should be invertible. Consider an algorithm, $\mathcal{A}$, that gives an interpolated image, K, "half-way between" images I and J, such that, $K = \mathcal{A}(I, J, 0.5)$. Logically, the algorithm should output K if one calls $\mathcal{A}(J, I, 0.5)$. Unfortunately, for a variety of reasons (including discretization error, lack of correspondence invertibility and the inherent ill-posedness of the problem), this is not the case in existing approaches.

The main advancement required, here, is a symmetric deformable image registration method that gives dense correspondence between I and J in space and time. Medical imaging interest in symmetric deformable image registration techniques was renewed largely by Christensen's inverse consistent image registration method (ICIR) [15], but has been a subject of image registration research since, at least, Thirion discussed the subject in [16]. However, neither algorithm uses large deformation formulations or computes dense, invertible maps in space and time, as we need here.

Large deformation geodesics were introduced into computer vision [17,18] for this purpose. Current developments in large deformation computational anatomy by Miller, Trouve and Younes extended the methods to include photometric variation [19] and to use the Euler-Lagrange equations of CA [20]. However, these methods are only symmetric with respect to their data and transformation terms *in theory* and their implementation requires parallel computation [21].

Our research differs, in particular, from Miller, et al's, in that we are focused not just on geodesic endpoints but also on reconstructing a *pair of maps* and inverses at any point along the geodesic. This enables us to map both image I and image J symmetrically to the same position along the geodesic. It also allows us to focus on "blending" images in shape and appearance and on the interpolation

problem, to our knowledge not studied by Miller, et al. This approach guarantees symmetry in a fast optimization algorithm, even when similarity terms such as mutual information are used [22]. Regardless of the input data or similarity terms, our algorithm will satisfy desirable continuity, anonymity and unanimity conditions as well as the theoretical symmetry conditions implied by geodesics. This is achieved by simple manipulation of the optimization formulation and algorithm.

Image registration, in the usual context, is interested in computing a map, $\phi(\mathbf{x}) = \mathbf{z}$, from image I to image J such that points in I correspond to points of J. That is, $I(\mathbf{x}) = J(\mathbf{z}) = I(\phi_1(\mathbf{x}))$ where, for simplicity, we take the equal sign literally. For interpolation, we must introduce a time parameter to the map ϕ_1. We refer to coordinates $\mathbf{x}$ in the time zero I domain, $\mathbf{z}$ in the time zero J domain and $\mathbf{y}$ in a common coordinate system that moves along the curve connecting $I(\mathbf{x})$ and $J(\mathbf{z})$. For deformation-based image interpolation we need to be able to define I and J at any $\mathbf{y}$ along the geodesic connecting the two images. This means we must have time parameterized maps that give $\phi_1(\mathbf{x}, t) = \mathbf{y} = \phi_2(\mathbf{z}, 1-t)$, allowing us to deform I and J such that,

$$\text{for any } \; t \; \in \; [0,1] \; I(\phi_1(\mathbf{x}, t)) = J(\phi_2(\mathbf{z}, 1-t)). \tag{1}$$

This gives a dense map in both space and time and is illustrated in figure 2. This paper will develop a method for symmetrically computing such maps.

These maps enable our interpolation method, geodesic image interpolation (**GII**), to operate at the interface of photometric and geometric variation. One can consider the method as a sliding metric between I and J that combines shape and appearance components of each. The method is fundamentally symmetric, will accomodate large deformations and also allows the flexibility of user-defined landmarks and statistical image similarity for the correspondence computation. Below, we follow background material on diffeomorphisms with a detailed development of the GII methods.

2 Mathematical Background on Diffeomorphisms

We now discuss some basic facts from the mathematics of diffeomorphisms, the mathematical underpinnings of GII. This section is derived from Arnol'd [23] and Marsden and Ratiu [24]. In this section, we will refer to ϕ as a general diffeomorphism.

We assume that the velocity fields, maps and images below are smooth, that is, infinitely differentiable or, at least, sufficiently differentiable. Velocity field smoothness guarantees the integrability necessary for generating diffeomorphisms. Typically, a linear operator, L, induces sufficient smoothness on the velocity field. See Dupuis [18] for a discussion in the context of image registration.

A diffeomorphism is a smooth one-to-one and onto map with a smooth inverse. Families of diffeomorphisms can be generated by integrating time-dependent velocity (vector) fields through an ordinary differential equation [23],

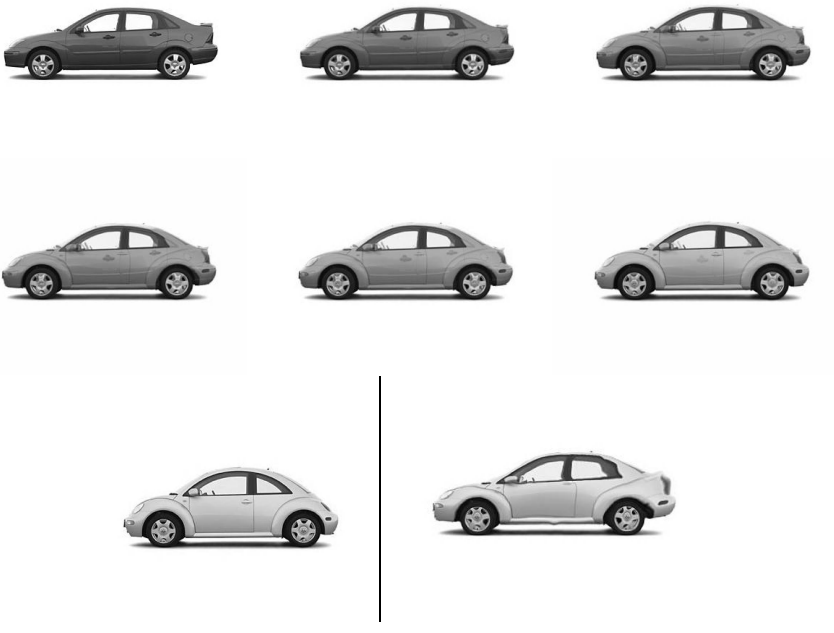


Fig. 1. Our automated shape and intensity interpolation method is used to morph between two automobile images, progressing left to right and top to bottom. Note that the transformation, ϕ_1, used for the morphing also details the shape difference and allows texture mapping. The beetle texture is mapped to the ford at bottom right. This mapping takes approximately two minutes with a mutual information similarity metric on an unoptimized Apple G4 processor.

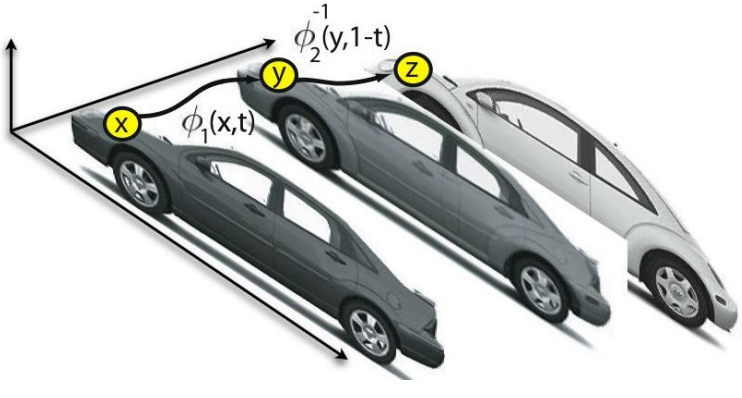


Fig. 2. An illustration of the geodesic path as it is used in interpolation. The curve shows the progress of one point as it proceeds along the interpolation path. The maps that parameterize the points are also given. The depth axis indicates increasing proximity to the shape and appearance of the beetle.

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \boldsymbol{V}(\mathbf{x}, t). \tag{2}$$

Note that the full derivative $d\phi$ reduces to the partial derivative when solving the *o.d.e.* in the Lagrangian frame. The $\boldsymbol{V}$ is a *material velocity* in the Lagrangian frame. The *spatial velocity* is computed at the tangent space to the diffeomorphism at time, t, such that $\boldsymbol{v}(\mathbf{y}, t) = \boldsymbol{V}(\mathbf{x}, t)$ with the map between them given by $\phi(\mathbf{x}, t) = \mathbf{y}$. Therefore, $\boldsymbol{V}(\mathbf{x}, t) = \boldsymbol{v}(\phi(\mathbf{x}, t), t)$. The velocities tell us how particles are moving through space: $\boldsymbol{V}$ assigns motion to a specific material point while $\boldsymbol{v}$ defines motion in a fixed coordinate system. The inverse map is $\phi^{-1}(\mathbf{y}, t) = \mathbf{x}$. We will drop explicit dependence on t or $\mathbf{x}$ where the meaning is clear. The group of deformable diffeomorphisms commonly used in image registration maps domain Ω to itself and is the identity at the boundary, $\partial\Omega$, such that $\phi(\partial\Omega) = \partial\Omega$. This boundary constraint assumes that rigid motion has been factored out. It also guarantees the transformation is everywhere one-to-one and onto and restricts the solution space to the diffeomorphic subgroup, $Diff_0$. [1]

The collection of diffeomorphisms forms the group, *Diff*. This is a Frechet Lie group [25] when *Diff* is C^∞. The length of a diffeomorphic path is similar to the length of a curve, $l(\phi) = \int_0^1 \|\boldsymbol{v}(\phi(t))\| dt$, where the infinitesimal increment in distance is given by the tangent to the diffeomorphism. A *geodesic* between ψ_1 and ψ_2, two elements of *Diff*, is defined by taking the infimum over all such paths [20],

$$D(\phi(0), \phi(1)) = \inf_{\phi} \int_0^1 \|L\boldsymbol{v}(\phi)\| dt, \tag{3}$$
$$\phi(0) = \psi_1 \text{ and } \phi(1) = \psi_2.$$

Taking the infimum guarantees that we have a geodesic between the elements in *Diff*. The length of the geodesic gives a metric distance measuring the amount of deformation caused by the diffeomorphism and will be fundamental to our symmetric formulation. The shape-based distance between images is given by setting $I(\phi(0)) = I$ and $I(\phi(1)) = J$. This distance does not depend on the origin of its measurement (it has right invariance) and is the basis for GII. Note the shorthand $\phi(1)I = I(\phi(\mathbf{x}, 1))$ where we use the notation convenient for the context.

3 Intrinsically Symmetric Image Registration

The diffeomorphism group, *Diff*, has several benefits for image processing. Foremost are that diffeomorphisms are composable and may be used to optimize variational energies, including landmark and intensity similarity measures, while maintaining domain topology. For example, our previous work [26] developed the Lagrangian Push Forward (LPF) method for diffeomorphic image and landmark matching.

[1] If a transformation is only a homeomorphism, one will lose, at least, the uniqueness properties given by initial conditions that are special to diffeomorphisms [23].

Without landmarks, LPF finds diffeomorphism, ϕ_1, connecting two images, I and J, by minimizing the global variational energy,

$$\inf_{\phi_1} \int_0^1 \|\boldsymbol{v}\|_L^2 dt + \int_\Omega \omega |I(\phi_1(\mathbf{x})) - J(\mathbf{z})|^2 d\Omega. \tag{4}$$

The first term is the deformation cost where $\|\boldsymbol{v}\|_L = \|L\boldsymbol{v}\|$ and L is a regularizer chosen from continuum mechanics, for example, $\nabla^2 + \mathbf{Id}$. The second term in the equation is a data term, Π, measuring the image difference with weight ω. The version of the similarity given here assumes I and J are identical up to a change of coordinates. Rarely used in morphing, the data term allows the image mapping to be fully automated or minimally guided by a user. Our work comparing chimpanzee and human cortices used both image similarity and expert landmarks to constrain the solution.

LPF generates both ϕ_1 and ϕ_1^{-1}, however it does not guarantee symmetric results numerically or when matches are inexact. That is, the geodesic path computed from I to J may not be the same when it is estimated from J to I. Nor does LPF directly generate ϕ_2, the map over all time defining motion of material in the J coordinate space. Although we use the novel ability of LPF to generate ϕ_1, ϕ_1^{-1}, we must reformulate equation 4 and extend LPF to give dense, symmetric space-time matching in real-world problems.

3.1 Symmetric Correspondence Formulation

A basic fact of diffeomorphisms allows them to be decomposed into parts [23]. We can thus write ϕ_1 defined in the domain cross parameterization space, $\Omega \times [0, 1]$, as

$$\phi_1(\mathbf{x}, 1) = \phi_2^{-1}(\phi_1(\mathbf{x}, s), 1 - s). \tag{5}$$

We also have $\phi_2(\mathbf{z}, 1) = \phi_1^{-1}(\phi_2(\mathbf{z}, 1 - s), s)$ and, at endpoints, $\phi_2(\mathbf{z}, 1) = \phi_1^{-1}(\mathbf{z}, 1)$ and $\phi_1(\mathbf{x}, 1) = \phi_2^{-1}(\mathbf{x}, 1)$. The map, ϕ_2, defines particle motion from the J coordinate system. Note that $\phi_1(\mathbf{x}, 1) = \phi_2(\mathbf{z}, 0) = \mathbf{z}$, the identity in the ϕ_2 space. For image registration, this also gives $I(\mathbf{x}) = J(\phi_2(\mathbf{z}, 1))$ with $\phi_1(\mathbf{x}, t) = \phi_2(\mathbf{z}, s)$ for all $t = 1 - s$.

These transformation components are important because it is more efficient and numerically symmetric to store/compute the ϕ_i and their inverses in $[0, 0.5]$, than throughout $[0, 1]$. This is because ϕ_1 and ϕ_2, defined in these intervals, allow one to reconstruct any point (or inverse) from ϕ_1 or ϕ_2 in $[0, 1]$. We will see how this is done in the following sections.

First, we give a variational energy that explicitly divides the image registration diffeomorphisms into two halves such that I and J contribute equally to the path and deformation is divided between them. This prior knowledge can be captured by including the constraint $D(\mathbf{Id}, \phi_1(\mathbf{x}, 0.5)) = D(\mathbf{Id}, \phi_2(\mathbf{z}, 0.5))$ directly in the optimization algorithm. The result is a method that finds correspondences with equal consideration of both images. Note that below we will derive the equations assuming Π as before, for simplicity. However, in actuality,

we have a variety of statistical image similarity measures (robust intensity difference, cross-correlation, mutual information) at our disposal, as in [22,27], or employ user landmarks as in [26].

A geodesic path ϕ_1 (equivalently ϕ_2) has the following global properties which should be reflected in our estimation method,

1. The geodesic minimizes the energy,

$$\inf_{\phi_1,\phi_2} \{ \int_0^1 \|\boldsymbol{v}\|_L^2 dt + \int_\Omega \omega(|\phi_1(1)I - J|^2 + |\phi_2(1)J - I|^2) \ d\Omega \}, \quad \text{where } \phi_1, \phi_2 \in \mathit{Diff}_0 \text{ relate as above.} \tag{6}$$

2. Its length is given by, for each i, $l(\phi_i) = \int_0^1 \|\boldsymbol{v}(\phi_i)\| dt$.

We now build the Intrinsically Symmetric variational energy, finding ϕ_1 and ϕ_2 by simultaneously optimizing the image registration problem both *forward* and *backward* in time. The motivation for solving this problem instead of (6) is to, first, induce symmetry and, second, it is more efficient to "meet" at the mean because we find two paths of length 0.5 rather than two of length 1.0. Another consideration is that we would like to use features from both images I and J to equally guide the correspondence. The same technique is used in dual source shortest path algorithms [28].

Define the image registration optimization time, $t \in [0,1]$ where t indexes both ϕ_1 and ϕ_2, though in opposite directions. The similarity seeks ϕ_1 such that $\phi_1(\mathbf{x},1)I = J$. [2] Recall the basic definition of diffeomorphisms allows us to write any geodesic through composing two parts. Then,

$$\begin{aligned} \phi_1(\mathbf{x},1)I &= J, \\ \phi_2^{-1}(\phi_1(\mathbf{x},s),1-s)I &= J, \\ \phi_2(\phi_2^{-1}(\phi_1(\mathbf{x},s),1-s),1-s)I &= \phi_2(\mathbf{z},1-s)J, \\ \phi_1(\mathbf{x},s)I &= \phi_2(\mathbf{z},1-s)J, \end{aligned} \tag{7}$$

gives the similarity term, $|\phi_1 I - \phi_2 J|^2$. The forward and backward energy is then, using t as a parameter and solving to time $t = 0.5$,

$$E_{sym}(I,J) = \inf_{\phi_1} \inf_{\phi_2} \int_{t=0}^{0.5} \omega\{ \|\boldsymbol{v}_1\|_L^2 + \|\boldsymbol{v}_2\|_L^2 \} dt + \int_\Omega |I(\phi_1(0.5)) - J(\phi_2(0.5))|^2 d\Omega. \tag{8}$$

Subject to:

$$\begin{aligned} &\boldsymbol{v}_1(0.5) = \boldsymbol{v}_2(0.5), \|\boldsymbol{v}_1(t)\|_L^2 = \|\boldsymbol{v}_2(t)\|_L^2 \\ &\text{with each } \phi_i \in \mathit{Diff}_0 \text{ the solution of:} \\ &d\phi_i/dt = \boldsymbol{v}_i(\phi_i(t)) \text{ with } \phi_i(0) = \mathbf{Id}. \end{aligned} \tag{9}$$

[2] We can interpret the equal sign as meaning equivalent according to our similarity criterion.

Minimization with respect to ϕ_1 and ϕ_2, upholding the arc length constraint, provides the *intrinsically symmetric* image registration (**ISIR**) solution and also solves a 2-mean problem. Landmarks may also be included, as in the LPF method, by dividing the similarity term, as done with the image match terms above. We assume that if either $\boldsymbol{v}_1 = \mathbf{0}$ or $\boldsymbol{v}_2 = \mathbf{0}$, then both are $\mathbf{0}$.

The two external constraints on the problem are critical. Recall that there is only a single velocity field (defining the geodesic's tangent) at any single time along the full geodesic. The first constraint, $\boldsymbol{v}_1(0.5) = \boldsymbol{v}_2(0.5)$, explicitly guarantees that this is true at the midpoint where the velocity fields merge in time. The second is the *a priori* geodesic averaging constraint, $D(\phi_1) = D(\phi_2)$, where D is the distance of equation 3. It is enforced, in practice, by Lagrangian multipliers. The geodesic mean constraint also enforces that the images I and J contribute equally to the result. The image configurations $I(\phi_1(0.5))$ and $J(\phi_2(0.5))$ are in "average" position. The total intrinsically symmetric transformation from I to J is $\phi_1(\mathbf{x}, 1) = \phi_2^{-1}(\phi_1(\mathbf{x}, 0.5), 0.5)$ (equation 5) and from J to I, $\phi_2(\mathbf{z}, 1) = \phi_1^{-1}(\phi_2(\mathbf{z}, 0.5), 0.5)$. This is distinct from inverse consistent image registration [29] in which a variational term is used to estimate consistency and where no mean is computed.

4 Geodesic Image Interpolation

We gave, in the previous section, an algorithm with symmetry properties one finds in linear averaging. It main outputs, ϕ_1 and ϕ_2 are defined in $[0, 0.5]$. For interpolation purposes, we need an *efficient and numerically symmetric* method to allow ϕ_1 and ϕ_2 to be defined anywhere in $[0, 1]$, the time/arc-length parameterization of ϕ_1 and ϕ_2. This dense definition in time will permit our geodesic image interpolation where the path between pixels, i and j, is given by a geodesic curve defined by the ϕ_i.

Consider the outputs ϕ_1 and ϕ_2. We compute the shape and intensity interpolation using the following properties of the methods:

1. Recall that the pixel i from $I(\mathbf{x})$ is in correspondence with pixel j at $J(\mathbf{z})$ through $\phi_2(\mathbf{z}, 1-s)$ and $\phi_1(\mathbf{x}, s)$. We then have any pixel pair on the geodesic by the definition of the ϕ_i in $[0, 0.5]$,

$$\phi_2(\mathbf{z}, 1 - s) = \phi_1(\phi_2(\mathbf{z}, 1), s), \tag{10}$$

and

$$\phi_1(\mathbf{x}, s) = \phi_2(\phi_1(\mathbf{x}, 1), 1 - s). \tag{11}$$

   These equations allow ϕ_1 in $[0, 1]$ to be found from ϕ_2 in $[0, 0.5]$ and ϕ_1 defined at time 1; likewise for ϕ_2. We choose, in practice, to compute ϕ_2 from $\phi_1, \phi_2(1)$ when $s < 0.5$ and ϕ_1 from $\phi_2, \phi_1(1)$ when $s > 0.5$. We resolve the ambiguity at $s = 0.5$ by an arbitrary but symmetric rule.
2. Consider two intensity values, i from I and j from J, each taken from their respective images warped to position $\phi_2(\mathbf{z}, 1 - s) = \phi_1(\mathbf{x}, s) = \mathbf{y}$. Linear

combination of the intensities may be parameterized by s such that geodesically interpolated intensity, k, is given as $k(s) = (1-s)i + sj$. If $s = 1$, then $k = j$.

3. Given statements 1) and 2) above, the values of k are well-defined in time interval $\tau = [0, 1]$ as ϕ_1, ϕ_1 and ϕ_2 (along with inverses) are also well-defined in τ.

This method uses the symmetric image correspondences gained by ISIR to perform geodesic image interpolation. The approach has the same properties of traditional linear interpolation, given a geodesic, but allows for the fact that interpolation sometimes requires a non-identity correspondence between pixels.

5 Applications and Results

Existing methods for registration-based interpolation [9] are asymmetric and use small deformation assumptions. Our method resolves this issue while allowing the interpolated image to be parameterized by arc length. The need for registration-based interpolation is two-fold. First, geodesic diffeomorphism-based methods permit one to interpolate slices in sparsely sampled volumetric data with a rigorous mathematical metric. Linear, intensity only interpolation leads to shape discontinuities and non-anatomical artifacts. This is illustrated in figures 3 and 4. Second, one often needs dense, continuous estimates to properly analyze discretely sampled dynamic processes, as when studying inter-patient lung dynamics [30].

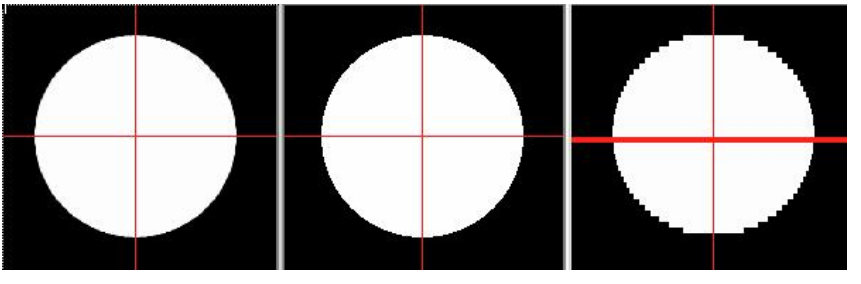


Fig. 3. The image at left is a section from an isotropic volumetric image of a sphere. The image was subsampled in the z-dimension from 200 to 50 slices. GII was used to reconstruct the 200 slice image and the result is shown at center. The reconstruction with linear interpolation is at right. Shape discontinuities caused by linear interpolation are not present in GII which almost exactly reconstructs the original image.

We now evaluate the ability of GII to interpolate missing data from real datasets. To achieve this, we simulate missing data by removing every other slice from 11 volumetric brain images. This procedure gives both realistic data input for the algorithm and a ground truth against which to compare the reconstruction. It also illustrates typical situations in which sparse sampling in one

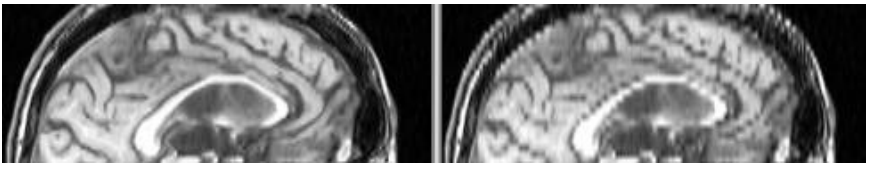


Fig. 4. The MRI image at left is reconstructed using GII while the image at right is reconstructed with linear interpolation. While gains in reconstruction intensity difference error are numerically modest (mean: 21.7% improvement, sdev: 15.1%), the visual difference is strikingly clear. This indicates that perhaps another measure, based on shape continuity, would be more indicative of performance gains with respect to perceptual value.

Table 1. Comparison of error between GII and linear interpolation. Larger values are better. The reported value is $100(1 - \text{GII Error/Lin Error})$ where Error is the intensity difference of the known slice and the interpolated slice.

Case:	1	2	3	4	5	6	7	8	9	10	11	12
% Geo Improvement:	12.4	12.7	18.4	17.8	13.8	18.3	19.4	18.6	20.0	15.6	34.0	65.8

image dimension or of a dynamic sequence causes anatomical discontinuity in the images. Then, for each pair of neighboring slice images, I and J, the intervening images may be interpolated from the geodesic connecting I and J or via traditional linear interpolation. Representative results of this study are shown in figure 4 and in table 1. The 12th case is the sphere example, where the sphere has a radius of 90 voxels in a $200x200x200$ voxel image.

We also choose a more taxing situation, where we operate on an 11 image sub-domain for each of two examples. This allows us to leave out 9 slices from each image. We can interpolate these 9 slices with GII and then evaluate the quality of the match between the interpolated and original data. This directly tests our ability to recover the "missing" slices from the image and corresponds to a sparse out-of-plane image acquisition protocol. These protocols are typically used in order to gain high in-plane resolution at the expense of volumetric sampling. Geodesic interpolation compares very favorably with linear interpolation (similar results were seen with sinc interpolation). In particular, obvious shape discontinuities are greatly reduced and anatomical boundaries are more distinct and continuous. *Numerically, we have found that GII recovers missing data, in this more texing experimental design, with 25-30% better intensity accuracy than linear interpolation.* Obvious improvements in visual clarity are also gained.

6 Conclusion

The GII method satisfies the axioms of continuity, symmetry and unanimity as does the linear interpolation method. The geodesic measure of distance is lever-

aged as a well-defined, deformation-based parameterization of deforming images, allowing one to "place" intermediate images at specified positions between a pair of "control" images. The algorithm can be used, in the pair-wise case, to compute spatio-temporal correspondences, which can be applied to reparameterizing dynamic sequences, given a pair of temporal end-points or to geodesic interpolation. We also illustrated its use, briefly, for texture mapping, recovering unknown intermediate shapes and missing data. Furthermore, ISIR generates a symmetric measure of the metric distance between images – this measure provides the basis for developing GII but can also be applied to defining a shape similarity measure for computer vision applications.

A valuable improvement to the interpolation method will be a full space-time optimizing solution. This will improve both the correspondence estimate and temporal smoothness at the cost of increased computation time and memory. We will also add higher order continuity models for image interpolation. We are particularly interested in large-scale application of these methods to developing a dynamic model of pulmonary anatomy. This will enable us to build a spatiotemporal model of the normal breathing cycle in order to improve our understanding of pathological conditions and normal variation in lung dynamics. We would also like to apply the methodology more extensively to texture mapping and motion interpolation problems. Finally, as the methods currently take on the order of seconds to minutes in $2D$ and minutes to hours in $3D$, improving coding strategies for these techniques is critical.

References

1. Thaddeus Beier and Shawn Neely, "Feature-based image metamorphosis," in *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1992, pp. 35–42, ACM Press.
2. "Special issue on morphing.," *The Visual Computer*, vol. 14, no. 8/9, pp. 359, 1998.
3. Andrew Witkin and Zoran Popović, "Motion warping," *Computer Graphics*, vol. 29, no. Annual Conference Series, pp. 105–108, 1995.
4. L. Tanco and A. Hilton, "Realistic synthesis of novel human movements from a database of motion capture examples," 2000.
5. G. T. Herman, J. Zheng, and C. A. Bucholtz, "Shape-based interpolation," *IEEE Comput. Graph. Appl.*, vol. 12, pp. 69–79, May 1992.
6. Huong Quynh Dinh, Anthony Yezzi, and Greg Turk, "Texture transfer during shape transformation," *ACM Trans. Graph.*, vol. 24, no. 2, pp. 289–310, 2005.
7. T.-Y. Lee and C.-H. Lin, "Feature-guided shape-based image interpolation," *IEEE Trans. Med. Imag.*, vol. 21, pp. 1479–1489, 2002.
8. S. P. Raya and J. K. Udupa, "Shape-based interpolation of multidimensional objects," *IEEE Trans. Med. Imag.*, vol. 9, pp. 34–42, 1990.
9. G.P. Penney, J.A. Schnabel, D. Rueckert, M.A. Viergever, and W.J. Niessen, "Registration-based interpolation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 7, pp. 922–926, 2004.
10. B. Dawant, R. Li, E. Cetinkaya, C. Kao, , J. Fitzpatrick, and P. Konrad, "Computerized atlas-guided positioning of deep brain simulators: A feasibility study," in *Workshop on Biomedical Image Registration*, J. Gee and J. B. Maintz, Eds., Philadelphia, July, 2003, pp. 142–150.

11. M. Ballmaier, J. T. O'Brien, E. J. Burton, P. M. Thompson, D. E. Rex, K. L. Narr, I. G. McKeith, H. DeLuca, and A. W. Toga, "Comparing gray matter loss profiles between dementia with lewy bodies and alzheimer's disease using cortical pattern matching: diagnosis and gender effects," *NeuroImage*, vol. 23, pp. 325–335, 2004.
12. M. Grossman, J. Farmer, S. Leight, M. Work, J. C. Gee, and et al., "Cerebrospinal fluid profile distinguishes frontotemporal dementia from Alzheimer's disease," *Brain*, 2005, in press.
13. B. Eckmann, "Social choice and topology," lecture notes, 2003.
14. S. Weinberger, "On the topological social choice model," *J. Econom. Theory*, vol. 115, no. 2, pp. 377–384, 2004.
15. G. Christensen and H. Johnson, "Consistent image registration," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 568–582, 2001.
16. J. Thirion, "Image matching as a diffusion process: an analogy with maxwell's demons," *Medical Image Analysis*, vol. 2, no. 3, pp. 243–260, 1998.
17. A. Trouve, "Diffeomorphism groups and pattern matching in image analysis," *Intl. J. Comp. Vis.*, vol. 28, no. 3, pp. 213–221, 1998.
18. P. Dupuis, U. Grenander, and M. I. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Quarterly of Applied Mathematics*, vol. 56, no. 3, pp. 587–600, 1998.
19. M. I. Miller and L. Younes, "Group actions, homeomorphisms and matching: a general framework," *Int. J. Computer Vision*, vol. 41, pp. 61–84, 2001.
20. M. Miller, A. Trouve, and L. Younes, "On the metrics and Euler-Lagrange equations of computational anatomy," *Annu. Rev. Biomed. Eng.*, vol. 4, pp. 375–405, 2002.
21. F. Beg, M. Miller, A. Trouve, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *Int. J. Comp. Vision*, vol. 61, pp. 139–157, 2005.
22. G. Hermosillo, C. Chefd'Hotel, and O. Faugeras, "A variational approach to multi-modal image matching," *Intl. J. Comp. Vis.*, vol. 50, no. 3, pp. 329–343, 2002.
23. V. I. Arnold, *Ordinary Differential Equations*, Springer-Verlag: Berlin, 1991.
24. J. Marsden and T. Ratiu, *Introduction to Mechanics and Symmetry*, Springer-Verlag: New York., 1999.
25. R. Schmid, "Infinite dimensional Lie groups with applications in mathematical physics," *Journal of Geometry and Symmetry in Physics*, vol. 1, pp. 1–67, 2004.
26. B. Avants, P. T. Schoenemann, and J. C. Gee, "Landmark and intensity-driven lagrangian frame diffeomorphic image registration: Application to structurally and functionally based inter-species comparison," in press, *Medical Image Analysis* special issue on **WBIR 2003**, 2005.
27. T. Yoo, Ed., *Insight Into Images: Theory for Segmentation, Registration and Image Analysis*, chapter Avants et. al., Non-rigid registration, A. K. Peters Ltd., Natick, MA, 2004.
28. R. Rivest, T. Cormen, and C. Leiserson, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1992.
29. H. J. Johnson and G. E. Christensen, "Consistent landmark and intensity-based image registration," *IEEE Trans. Med. Imaging*, vol. 21, no. 5, pp. 450–461, 2002.
30. T. Sundaram, B. Avants, and J. C. Gee, "A dynamic model of average lung deformation using area-based reparameterization and shape averaging of lung MR images," in *Proceedings of Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2004, pp. II:1000–1007.

A Variational Approach for Object Contour Tracking

Nicolas Papadakis, Etienne Mémin, and Frédéric Cao

IRISA/INRIA,
Campus de Beaulieu,
35 042 Rennes Cedex, France
{npapadak, memin, fcao}@irisa.fr

Abstract. In this paper we describe a new framework for the tracking of closed curves described through implicit surface modeling. The approach proposed here enables a continuous tracking along an image sequence of deformable object contours. Such an approach is formalized through the minimization of a global spatio-temporal continuous cost functional stemming from a Bayesian Maximum a posteriori estimation of a Gaussian probability distribution. The resulting minimization sequence consists in a forward integration of an evolution law followed by a backward integration of an adjoint evolution model. This latter pde include also a term related to the discrepancy between the curve evolution law and a noisy observation of the curve. The efficiency of the approach is demonstrated on image sequences showing deformable objects of different natures.

1 Introduction

Tracking curves and contours is an important and difficult problem in computer vision. As a matter of fact, the shapes of deformable or rigid objects may vary a lot along an image sequence. These changes are due to perspective effects, self occlusions or to complex deformations of the object itself. The intrinsic continuous nature of these features and their high dimensionality makes difficult the conception of efficient non linear Bayesian filters as sampling in large scale dimension is usually completely inefficient. Besides, the use of lower dimensional features such as explicit parametric curves is limited to the visual tracking of objects with well defined shapes and that do not exhibit any change of topology [2,15]. This kind of representation is for instance very difficult to settle when focusing on the tracking of temperature level curves in satellite atmospheric images, or simply when the aim is to track an unknown deformable object with no predefined shape.

In such a context, approaches based on level set representation have been proposed [5,6,8,11,12,14,16]. Nevertheless, apart from [11], all these solutions aim more at estimating successive instantaneous segmentation maps than at really tracking objects. Indeed, in a formal point of view, they cannot be really considered as tracking approaches for several reasons. First of all, these methods are very sensitive to noise [9]. Unless the introduction of some statistical knowledges on the shape [3,7,13] these approaches do not allow to handle partial occlusions of the target. Since these methods do not include any temporal evolution law on the tracked object shape, they are not able to cope with severe failures of the imaging sensor (for instance a complete loss of image

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 259–270, 2005.

data, a severe motion blur or high saturation caused by over exposure). And finally, no error assessment on the estimation is in addition possible. For all these reasons, only approaches introducing basically a competition between a dynamical evolution model and a measurement process of the target of interest enable to handle a robust tracking in natural way. On the same basis, we propose here a variational method allowing to combine these two ingredients for the tracking of non parametric curves.

Unlike the technique proposed in [11] which explicitly also introduces a dynamic law in the curve evolution, our work includes in the same spirit as a Kalman smoothing a temporal smoothing along the whole image sequence. In the same way as a Kalman smoother the technique we propose here allows to estimate the conditional expectation of the state process given all the available measurements extracted in the whole image sequence. Nevertheless, unlike stochastic techniques our approach allows to handle features of very high dimension.

The variational tracking technique we introduce relies on data assimilation concepts used for instance in meteorology [1,4,17]. As will be demonstrated in the experimental section, such a technique enables to handle naturally partial occlusions and a complete loss of image data on long time period without resorting to complex mechanisms.

2 A System for Contour Tracking

As we wish to focus in this work on the tracking of non parametric closed curves that may exhibit topology changes during the time of the analyzed image sequence, we will rely on an implicit level set representation of the curve of interest $\Gamma(t)$ at time $t \in [t_0, \tau]$ of the image sequence [12,16]. Within that framework, the curve $\Gamma(t)$ enclosing the target to track is implicitly described as the zero level set of a function $\phi(\boldsymbol{x}, t) : \Omega \times \mathbb{R}+ \rightarrow \mathbb{R}$:

$$\Gamma(t) = \{\boldsymbol{x} \in \Omega \mid \phi(\boldsymbol{x}, t) = 0\},$$

where Ω stands for the image spatial domain. This representation enables an Eulerian representation of the evolving contours. As such, it allows to avoid the inescapable regriding ad-hoc processes of the different control points associated to any explicit spline based Lagrangian description of the evolving curve. The problem we want to face thus consists in estimating for a whole time range the state of an unknown curve, and hence of its associated implicit surface ϕ. To that end, we first define an *a priori* evolution law of the unknown surface. We will assume that the curve is transported at each frame instant by a velocity fields ,$\boldsymbol{w}(t)$, and diffuses according to a mean curvature motion. In term of the implicit surface this evolution model reads:

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + \boldsymbol{\nabla}\phi(\boldsymbol{x}, t)^T \boldsymbol{w}(t) = \epsilon\kappa \|\boldsymbol{\nabla\phi}\|, \tag{1}$$

where κ denotes the curve curvature. Introducing the surface normal, equation (1) can be written as:

$$\frac{\partial \phi}{\partial t} = -\left(\boldsymbol{w} \cdot \boldsymbol{n} - \epsilon\kappa\right) \|\boldsymbol{\nabla\phi}\|, \tag{2}$$

where the normal and the curvature are given directly in term of surface gradient,

$$\text{with} \quad \kappa = div\left(\frac{\boldsymbol{\nabla}\phi}{\|\boldsymbol{\nabla}\phi\|}\right) \quad \text{and } \boldsymbol{n} = \frac{\boldsymbol{\nabla}\phi}{\|\boldsymbol{\nabla}\phi\|}.$$

At the initial time, the implicit function is assigned to a signed distance function up to a white noise. More precisely, the value of $\phi(\boldsymbol{x}, t_0)$ is set to the distance $g(\boldsymbol{x}, \Gamma(t_0))$ of the closest point of a given initial curve $\Gamma(t_0)$, with the convention that $g(\boldsymbol{x}, t_0)$ is negative inside the contour, and positive outside. An additive white noise process is added in order to model the uncertainty we have on the initial curve. Associated to this evolution model and to the initialization process we previously described, we will assume that an observation function $Y(t)$ which constitutes a noisy measurement of the target is available. This function will be assumed to be related to the unknown state function ϕ through a differential operator $\mathbb{H}$:

$$Y(\boldsymbol{x}, t) = \mathbb{H}(\phi(\boldsymbol{x}, t)) + \varepsilon(\boldsymbol{x}, t), \tag{3}$$

and a white Gaussian noise $\varepsilon(\boldsymbol{x}, t)$. Let us note that in our case, the continuous observation function, $Y(t)$, is obtained from discrete image frames, I_n, through multiplication by a family of localization functions. These functions can be defined from delta functions at the observed time and location, or from more advanced spatio-temporal averaging functions. Gathering all the elements we have described so far, we get the following system for our tracking problem:

$$\begin{cases} \frac{\partial \phi}{\partial t} + \mathbb{M}(\phi) = \eta(\boldsymbol{x}, t) \\ \phi(\boldsymbol{x}, t_0) = g(\boldsymbol{x}, \Gamma(t_0)) + \nu(\boldsymbol{x}, t) \\ Y(\boldsymbol{x}, t) = \mathbb{H}(\phi) + \varepsilon(\boldsymbol{x}, t) \end{cases} \tag{4}$$

In this system, $\mathbb{M}$, denotes the differential operator involved in equation (2) and η, ν and ε are time varying zero mean Gaussian noise functions defined on the whole image plane, with covariance functions $Q(\boldsymbol{x}, t, \boldsymbol{x}', t')$, $B(\boldsymbol{x}, \boldsymbol{x}')$, $R(\boldsymbol{x}, t, \boldsymbol{x}' t')$ respectively. The noise functions represent the different errors involved in the different components of our system.

3 Variational Tracking Formulation

3.1 Penalty Function

Considering a system such as the one we previously settled comes to fix the conditional probability $p(\phi(t)|g)$, $p(g|\phi(t_0))$ and $p(Y(t)|\phi(t))$. From these pdf's, one get the *a posteriori* density function up to a normalization constant. As all the error distributions involved here are Gaussian, the *a posteriori* pdf is also Gaussian. The maximization of this distribution is thus equivalent to the minimization of the following quadratic penalty function:

$$\begin{aligned} J(\phi) = & \frac{1}{2} \int_{\Omega,t} \int_{\Omega,t} \left(\frac{\partial \phi}{\partial t} + \mathbb{M}(\phi) \right)^T (\boldsymbol{x}, t) Q^{-1}(\boldsymbol{x}, t, \boldsymbol{x}', t') \left(\frac{\partial \phi}{\partial t} + \mathbb{M}(\phi) \right) (\boldsymbol{x}', t') dt' d\boldsymbol{x}' dt d\boldsymbol{x} \\ & + \frac{1}{2} \int_{\Omega} \int_{\Omega} \left(\phi(\boldsymbol{x}, t_0) - g(\boldsymbol{x}, \Gamma(t_0)) \right)^T B^{-1}(\boldsymbol{x}, \boldsymbol{x}') \left(\phi(\boldsymbol{x}', t_0) - g(\boldsymbol{x}', \Gamma(t_0)) \right) d\boldsymbol{x}' d\boldsymbol{x} \\ & + \frac{1}{2} \int_{\Omega,t} \int_{\Omega,t} \left(Y - \mathbb{H}(\phi) \right)^T (\boldsymbol{x}, t) R^{-1}(\boldsymbol{x}, t, \boldsymbol{x}', t') \left(Y - \mathbb{H}(\phi) \right) (\boldsymbol{x}', t') dt' d\boldsymbol{x}' dt d\boldsymbol{x}. \end{aligned} \tag{5}$$

In this functional, $\boldsymbol{x}$, denotes the spatial domain coordinates defined on the image domain Ω and t is a continuous time index lying within the image sequence time interval $[t_0 = 0; \tau]$. The minimizer of this expression minimizes a sum of norms expressing all the possible correlations between errors at arbitrary two points of the image sequence. The double space and time integrations are here due to the fact that the covariance functions are first assumed to be non null for any two points $(\boldsymbol{x}, t)$ and $(\boldsymbol{x}', t')$ (correlated case) in order to stick to the most general case before considering a simplified uncorrelated case corresponding to diagonal correlation matrices in a discrete setting.

In order to devise a minimizing sequence let us now derive the associated Euler-Lagrange equations.

3.2 Euler-Lagrange Equations

Function ϕ is a minimum of functional J, if it is also a minimum of a cost function $J(\phi + \beta\theta(\boldsymbol{x}, t))$, where $\theta(\boldsymbol{x}, t))$ belongs to a space of admissible functions and β is a positive parameter. In other words, ϕ must cancel the directional derivative :

$$\delta J_\phi(\theta) = \lim_{\beta \to 0} \frac{dJ(\phi + \beta\theta(\boldsymbol{x}, t))}{d\beta} = 0.$$

The cost function $J(\phi + \beta\theta(\boldsymbol{x}, t))$ reads

$$\begin{aligned} J = &\frac{1}{2}\int_\Omega \int_0^\tau \left[\left(\frac{\partial \phi}{\partial t} + \beta\frac{\partial \theta}{\partial t} + \mathbb{M}(\phi + \beta\theta)\right)^T \int_\Omega \int_0^\tau Q^{-1}\left(\frac{\partial \phi}{\partial t} + \beta\frac{\partial \theta}{\partial t} + \mathbb{M}(\phi + \beta\theta)\right) dt' d\boldsymbol{x}'\right] dt d\boldsymbol{x} \\ &+ \frac{1}{2}\int_\Omega \int_\Omega (\phi + \beta\theta - g)^T B^{-1}(\phi + \beta\theta - g) d\boldsymbol{x}' d\boldsymbol{x} \\ &+ \frac{1}{2}\int_\Omega \int_0^\tau \int_\Omega \int_0^\tau (Y - \mathbb{H}(\phi + \beta\theta))^T R^{-1}(Y - \mathbb{H}(\phi + \beta\theta)) dt' d\boldsymbol{x}' dt d\boldsymbol{x}. \end{aligned} \tag{6}$$

Adjoint Variable. In order to perform an integration by part – to factorize this expression by θ– we introduce an "adjoint variable" λ defined by:

$$\lambda(\boldsymbol{x}, t) = \int_\Omega \int_0^\tau Q^{-1}\left(\frac{\partial \phi}{\partial t} + \mathbb{M}(\phi)\right) dt' d\boldsymbol{x}', \tag{7}$$

as well as *linear tangent operators* $(\frac{\partial \mathbb{M}}{\partial \phi})$ and $(\frac{\partial \mathbb{H}}{\partial \phi})$ defined by

$$\lim_{\beta \to 0} \frac{d\mathbb{M}(\phi + \beta\theta)}{d\beta} = \frac{\partial \mathbb{M}}{\partial \phi}(\theta). \tag{8}$$

By taking the limit $\beta \to 0$, the derivative of expression (6) then reads

$$\begin{aligned} \lim_{\beta \to 0} \frac{dJ}{d\beta} = &\int_\Omega \int_0^\tau \left(\frac{\partial \theta}{\partial t} + \frac{\partial \mathbb{M}}{\partial \phi}\theta\right)^T \lambda(\boldsymbol{x}, t) dt d\boldsymbol{x} \\ &+ \int_\Omega \int_\Omega \theta^T(\boldsymbol{x}, 0) B^{-1}(\phi(\boldsymbol{x}', 0) - g(\boldsymbol{x}', 0)) d\boldsymbol{x}' d\boldsymbol{x} \\ &- \int_\Omega \int_0^\tau \int_\Omega \int_0^\tau \left(\frac{\partial \mathbb{H}}{\partial \phi}\theta\right)^T R^{-1}(Y - \mathbb{H}(\phi)) dt' d\boldsymbol{x}' dt d\boldsymbol{x} \\ &= 0. \end{aligned} \tag{9}$$

Considering the three following integrations by parts, we can get rid of the partial derivatives of the admissible function θ in expression (9), i.e.

$$\begin{aligned}\int_\Omega \int_0^\tau \frac{\partial \theta}{\partial t}\lambda(\boldsymbol{x},t)dtd\boldsymbol{x} = \int_\Omega \theta^T(\boldsymbol{x},\tau)\lambda(\boldsymbol{x},\tau)d\boldsymbol{x} - \int_\Omega \theta^T(\boldsymbol{x},0)\lambda(\boldsymbol{x},0)d\boldsymbol{x} \\ - \int_\Omega \int_0^\tau \theta^T(\boldsymbol{x},t)\frac{\partial \lambda}{\partial t}dtd\boldsymbol{x},\end{aligned} \tag{10}$$

$$\int_\Omega \int_0^\tau \left(\frac{\partial \mathbb{M}}{\partial \phi}\theta\right)^T \lambda(\boldsymbol{x},t)dtd\boldsymbol{x} = \int_\Omega \int_0^\tau \theta^T (\frac{\partial \mathbb{M}}{\partial \phi})^* \lambda(\boldsymbol{x},t)dtd\boldsymbol{x}, \tag{11}$$

$$\begin{aligned}\int_\Omega \int_0^\tau \int_\Omega \int_0^\tau \left(\frac{\partial \mathbb{H}}{\partial \phi}\theta\right)^T R^{-1}(Y-\mathbb{H}(\phi))dt'd\boldsymbol{x}'dtd\boldsymbol{x} = \\ \int_\Omega \int_0^\tau \int_\Omega \int_0^\tau \theta^T (\frac{\partial \mathbb{H}}{\partial \phi})^* R^{-1}\left(Y-\mathbb{H}(\phi)\right)dt'd\boldsymbol{x}'dtd\boldsymbol{x}.\end{aligned} \tag{12}$$

In the two last equations, we have introduced adjoint operators $(\frac{\partial \mathbb{M}}{\partial \phi})^*$ and $(\frac{\partial \mathbb{H}}{\partial \phi})^*$ as compact notations of the integration by parts of the associated linear tangent operators. Gathering all these elements, equation (9) can be rewritten as

$$\begin{aligned}&\lim_{\beta\to 0}\frac{dJ}{d\beta} = \\ &\int_\Omega \theta^T(\boldsymbol{x},\tau)\lambda(\boldsymbol{x},\tau)d\boldsymbol{x} + \int_\Omega \theta^T(\boldsymbol{x},0)\left[\int_\Omega \left(B^{-1}(\phi(\boldsymbol{x}',0)-g(\boldsymbol{x}',0))-\lambda(\boldsymbol{x},0)\right)d\boldsymbol{x}'\right]d\boldsymbol{x} \\ &+ \int_\Omega \int_0^\tau \theta^T\left[\left(-\frac{\partial \lambda}{\partial t}+(\frac{\partial \mathbb{M}}{\partial \phi})^*\lambda\right) - \int_\Omega \int_0^\tau (\frac{\partial \mathbb{H}}{\partial \phi})^* R^{-1}(Y-\mathbb{H}(\phi))dt'd\boldsymbol{x}'\right]dtd\boldsymbol{x} = 0.\end{aligned} \tag{13}$$

Forward-Backward Equations. Since the functional derivative must be null for arbitrary independent admissible functions in the three integrals of expression (13), all the other members appearing in the three integral terms must be identically null. We finally obtain a coupled system of forward and backward PDE's with two initial and end conditions:

$$\lambda(\boldsymbol{x},\tau) = 0 \tag{14}$$

$$-\frac{\partial \lambda}{\partial t}+(\frac{\partial \mathbb{M}}{\partial \phi})^*\lambda = \int_\Omega \int_0^\tau (\frac{\partial \mathbb{H}}{\partial \phi})^* R^{-1}(Y-\mathbb{H}(\phi))dtd\boldsymbol{x} \tag{15}$$

$$\lambda(\boldsymbol{x},0) = \int_\Omega \left(B^{-1}(\phi(\boldsymbol{x}',0)-g(\boldsymbol{x}',0))\right)d\boldsymbol{x}' \tag{16}$$

$$\frac{\partial \phi(\boldsymbol{x},t)}{\partial t}+\mathbb{M}(\phi(\boldsymbol{x},t)) = \int_\Omega \int_0^\tau Q\lambda(\boldsymbol{x}',t')dt'd\boldsymbol{x}'. \tag{17}$$

The forward equation (17) corresponds to the definition of the adjoint variable (7) and has been obtained introducing Q, the pseudo-inverse of Q^{-1}, defined as [1]:

$$\int_\Omega \int_0^\tau Q(\boldsymbol{x},t,\boldsymbol{x}',t')Q^{-1}(\boldsymbol{x}',t',\boldsymbol{x}'',t'')dt'd\boldsymbol{x}' = \delta(\boldsymbol{x}-\boldsymbol{x}'')\delta(t-t'').$$

We will discuss the discretization of these equations in the next section. Before that, we can make several remarks. First of all, we can see that eq. (14) constitutes an explicit end condition for the adjoint evolution model eq.(15). This adjoint evolution model can be integrated backward from the end condition assuming the knowledge of an initial guess for ϕ to compute the discrepancy $Y - \mathbb{H}(\phi)$. To perform this integration, we also need to have an expression of the *adjoint evolution operator*. Let us recall, that this operator is defined from an integration by part of the *linear tangent operator* associated to the evolution law operator. The analytic expression of such an operator is obviously not accessible in general. Nevertheless, it can be noticed that a discrete expression of this operator can be easily obtained from the discretization of the linear tangent operator. As a matter of fact, the adjoint of the linear tangent operator discretized as a matrix consists simply of the transpose of that matrix. Knowing a first solution of the adjoint variable, an initial condition for the state variable can be obtained from (16) and a pseudo inverse expression of the covariance matrix B. From this initial condition, (17) can be finally integrated forward.

Incremental State Function. The previous system can be modified slightly to produce an adequate initial guess for the state function. Considering a function of state increments linking the state function and an initial condition function, $\delta\phi = \phi - \psi$, and linearizing the operator $\mathbb{M}$ around the initial condition function ψ:

$$\mathbb{M}(\phi) = \mathbb{M}(\psi) + \frac{\partial \mathbb{M}}{\partial \psi}(\delta\phi),$$

it is possible to split equation (17) into two pde's with an explicit initial condition:

$$\psi(\boldsymbol{x}, 0) = g(\boldsymbol{x}, \Gamma(t_0)) \tag{18}$$

$$\frac{\partial \psi}{\partial t} + \mathbb{M}(\psi) = 0 \tag{19}$$

$$\frac{\partial \delta\phi}{\partial t} + (\frac{\partial \mathbb{M}}{\partial \psi})\delta\phi = \int_\Omega \int_0^\tau Q(\boldsymbol{x}', t', \boldsymbol{x}, t)\lambda(\boldsymbol{x}, t) dt d\boldsymbol{x}. \tag{20}$$

The first equation initializes function ψ as a signed distance function corresponding to the initial contours. Integrating forward equation (19) provides an initial guess of the state function (assuming the increment is initially null). This initial guess can then be used for the backward integration of the adjoint variable (15). The increment state function is updated by a forward integration of equation (20). These two last integrations successively iterated until convergence constitute the overall process.

4 Curve Tracking Implementation

In this section, we describe further the implementation of the method we propose for object contour tracking. We present the discretization scheme we used and give the analytic expression of the tangent linear operator associated to our evolution model.

4.1 Tangent Linear Operator

Considering a (nonlinear) operator $\mathbb{G}$ mapping one element of an initial functional space to another functional space, the tangent linear operator to $\mathbb{G}$ at point m is a linear operator defined by the limit:

$$\lim_{\gamma \to 0} \frac{\mathbb{G}(m + \gamma h) - \mathbb{G}(m)}{\gamma} = \frac{\partial \mathbb{G}}{\partial m}(h) \tag{21}$$

where γh is a small perturbation in the initial space. The tangent linear operator $\frac{\partial \mathbb{G}}{\partial m}$ is also known as the Gâteaux derivative of $\mathbb{G}$ at point m. Let us note that the Gâteaux derivative of a linear operator is the operator itself.

In our case, the evolution operator reads:

$$\mathbb{M}(\phi) = \boldsymbol{\nabla}\phi^{T}\boldsymbol{w} - \epsilon||\boldsymbol{\nabla}\phi||div\left(\frac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||}\right).$$

This operator can be turned into a more tractable expression:

$$\mathbb{M}(\phi) = \boldsymbol{\nabla}\phi \cdot \boldsymbol{w} - \epsilon\left(\Delta\phi - \frac{\boldsymbol{\nabla}^{T}\phi\nabla^{2}\phi\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||^{2}}\right).$$

After some calculations, the tangent linear operator to $\mathbb{M}$ at point ψ finally reads:

$$(\frac{\partial \mathbb{M}}{\partial \psi})\delta\phi = \boldsymbol{\nabla}\delta\phi \cdot \boldsymbol{w} - \epsilon\Big[\Delta\delta\phi - \frac{\boldsymbol{\nabla}\psi^{T}\nabla^{2}\delta\phi\boldsymbol{\nabla}\psi}{||\boldsymbol{\nabla}\psi||^{2}} + 2\frac{\boldsymbol{\nabla}\psi^{T}\nabla^{2}\psi}{||\boldsymbol{\nabla}\psi||^{2}}\left(\frac{\boldsymbol{\nabla}\psi\boldsymbol{\nabla}\psi^{T}}{||\boldsymbol{\nabla}\psi||^{2}} - Id\right)\boldsymbol{\nabla}\delta\phi\Big].$$

4.2 Algorithm Specification

Up to now, we did not specified yet the observation function Y associated to our tracking problem. In order to have the simplest interaction as possible, we defined it in the same space as ϕ, that is to say $\mathbb{H} = Id$. The observation of the evolving object contour is set to the signed distance map to an observed closed curve, $g(\boldsymbol{x}, \Gamma(t))$. Such curves are assumed to be generated by a basic threshold segmentation method or provided by some moving object detection method. These observations are generally of bad quality. As a matter of fact, in the first case, very noisy curves are observed whereas in the later case, when the object motion is too slow, there is no detection at all. As for the motion field $\boldsymbol{w}$, we used in this work an efficient and robust version of the Horn and Schunck optical-flow estimator [10].

Combining equations (14-15-16) and (18-19-20), we finally get the following iterative tracking system:

$$\psi^{k=0}(\boldsymbol{x}, t_0) = g(\boldsymbol{x}, \Gamma(t_0)) \tag{22}$$

$$\frac{\partial \psi^{0}}{\partial t} + \mathbb{M}(\psi^{0}) = 0 \tag{23}$$

$$\lambda^k(\tau) = 0 \tag{24}$$

$$-\frac{\partial \lambda^k}{\partial t} + (\frac{\partial \mathbb{M}}{\partial \psi^k})^* \lambda^k = \int_0^\tau R^{-1}\left(Y - \psi^k\right) \tag{25}$$

$$\lambda^k(t_0) = B^{-1}\delta\phi^k(t_0) \tag{26}$$

$$\frac{\partial \delta\phi^k}{\partial t} + (\frac{\partial \mathbb{M}}{\partial \psi^k})\delta\phi^k = \int_0^\tau Q\lambda^k(t). \tag{27}$$

A forward integration of the initial condition function (23) is done at the first iteration. Index k, represents the current iteration which consists of a backward integration of the adjoint function and a forward integration of the increment function (24 - 27). At the end of the iteration , ψ is updated according to the relation $\psi^{k+1} = \phi^k = \psi^k + \delta\phi^k$. We have chosen to represent the covariance matrix B as the diagonal matrix $B(\boldsymbol{x}, \boldsymbol{x}) = Id - e^{-|\psi(\boldsymbol{x},t_0)|}$. In a similar way, we define matrix R from the observations Y, as

$$R = R_{min} + (R_{max} - R_{min})(Id - e^{-|Y(\boldsymbol{x},t)|}).$$

This observation covariance matrix has therefore lower values in the vicinity of the observed curves and higher values faraway from them. When there is no observed curve, all the value of this covariance matrix are set to infinity. Otherwise, covariance matrix Q, has been fixed to a constant diagonal matrix.

4.3 Operator Discretization

We will denote by $\phi^t_{i,j}$ the value of ϕ at image grid point (i, j) at time $t \in [0; \tau]$. Using (23) and a semi-implicit discretization scheme, the following discrete evolution model is obtained:

$$\frac{\phi^{t+\Delta t}_{i,j} - \phi^t_{i,j}}{\Delta t} + \mathbb{M}_{\phi^t_{i,j}} \phi^{t+\Delta t}_{i,j} = 0.$$

Considering ϕ_x and ϕ_y, the horizontal and vertical gradient matrices of ϕ, the discrete operator $\mathbb{M}$ is obtained as :

$$\mathbb{M}_{\phi^t_{i,j}} \phi^{t+\Delta t}_{i,j} = \begin{pmatrix} (\phi_x^{t+\Delta t})_{i,j} \\ (\phi_y^{t+\Delta t})_{i,j} \end{pmatrix}^T \boldsymbol{w} - \frac{\epsilon}{||\boldsymbol{\nabla}\phi^t_{i,j}||^2} \begin{pmatrix} -(\phi_y^t)_{i,j} \\ (\phi_x^t)_{i,j} \end{pmatrix}^T \nabla^2 \phi^{t+\Delta t}_{i,j} \begin{pmatrix} -(\phi_y^t)_{i,j} \\ (\phi_x^t)_{i,j} \end{pmatrix},$$

where we used usual finite differences for the advection term $\boldsymbol{\nabla}\phi^T \boldsymbol{w}$ and the Hessian matrix $\nabla^2\phi$. The discrete linear tangent operator (27) is similarly defined as:

$$\frac{\partial \mathbb{M}}{\partial \phi^t_{i,j}} \delta\phi^{t+\Delta t}_{i,j} = \mathbb{M}_{\phi^t_{i,j}} \delta\phi^{t+\Delta t}_{i,j} - \frac{2\epsilon \begin{pmatrix} A & B \end{pmatrix}}{||\boldsymbol{\nabla}\phi||^4} \begin{pmatrix} (\delta\phi_x^{t+\Delta t})_{i,j} \\ (\delta\phi_y^{t+\Delta t})_{i,j} \end{pmatrix},$$

where A and B are defined as:

$$A = \phi^t_x\phi^t_y(\phi^t_{xy}\phi^t_x - \phi^t_{xx}\phi^t_y) + (\phi^t_y)^2(\phi^t_{yy}\phi^t_x - \phi^t_{xy}\phi^t_y),$$
$$B = \phi^t_x\phi^t_y(\phi^t_{xy}\phi^t_y - \phi^t_{yy}\phi^t_x) + (\phi^t_x)^2(\phi^t_{xx}\phi^t_y - \phi^t_{xy}\phi^t_x).$$

As previously indicated, the discretization of the adjoint evolution model is obtained as the transposed matrix corresponding to the discretization of the derivative of the evolution law operator. Otherwise, we used a conjugated gradient optimization for the iterative solver involved in the implicit discretization.

5 Numerical Results

In this section, we present results we obtained for three different kinds of image sequences. The first sequence is a 16 frames sequence presenting a moving skate fish on the sand (fig. 1). As this kind of fish possesses natural camouflage mechanisms, its luminance and texture is very similar to the surrounding sand. The contours of such an object are therefore really difficult to extract. For this sequence we used a simple segmentation algorithm based on selection of intensity level curves. To further demonstrate the robustness of our tracking approach, we only considered observations at every third frames (i.e for $k = 0, 3, 6, 9, 12, 15$). It can be noticed on the second row of figure 1 that the global shape and the successive locations of the skate have been well reconstructed at all time instants $\in [t_0, t_{16}]$. The noisy and instable observed contours have been smoothed in an appropriated way. For instance, it can be outlined that the tech-

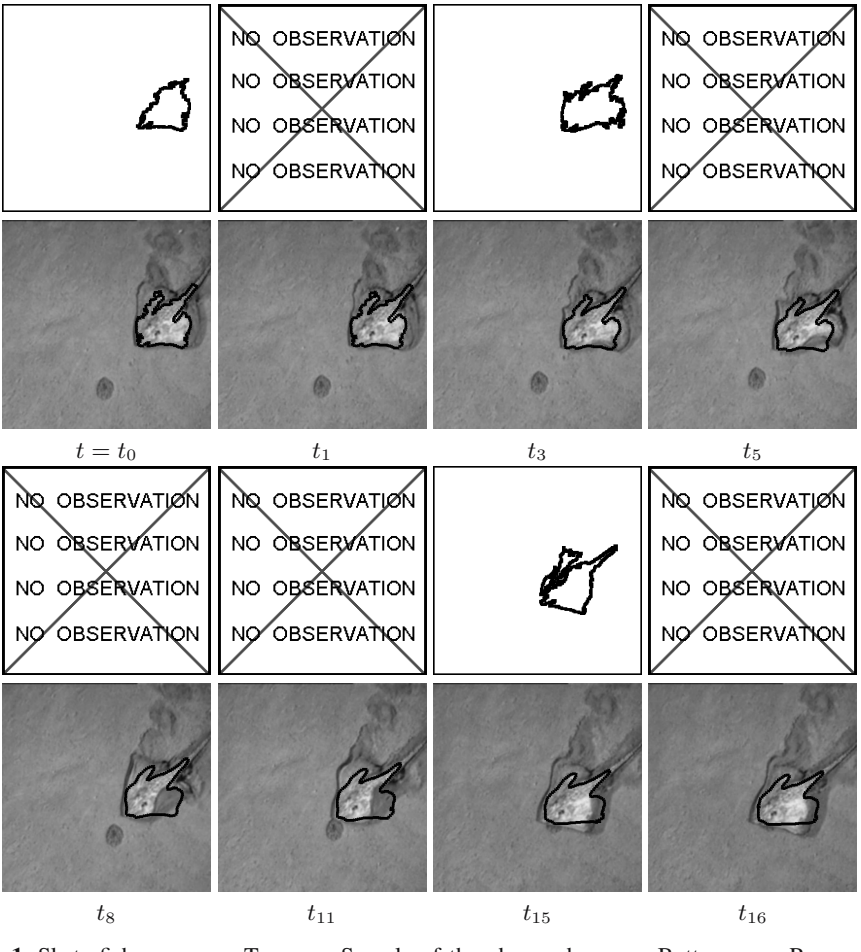


Fig. 1. Skate fish sequence. Top row: Sample of the observed curves. Bottom row: Recovered curve superimposed on the corresponding image.

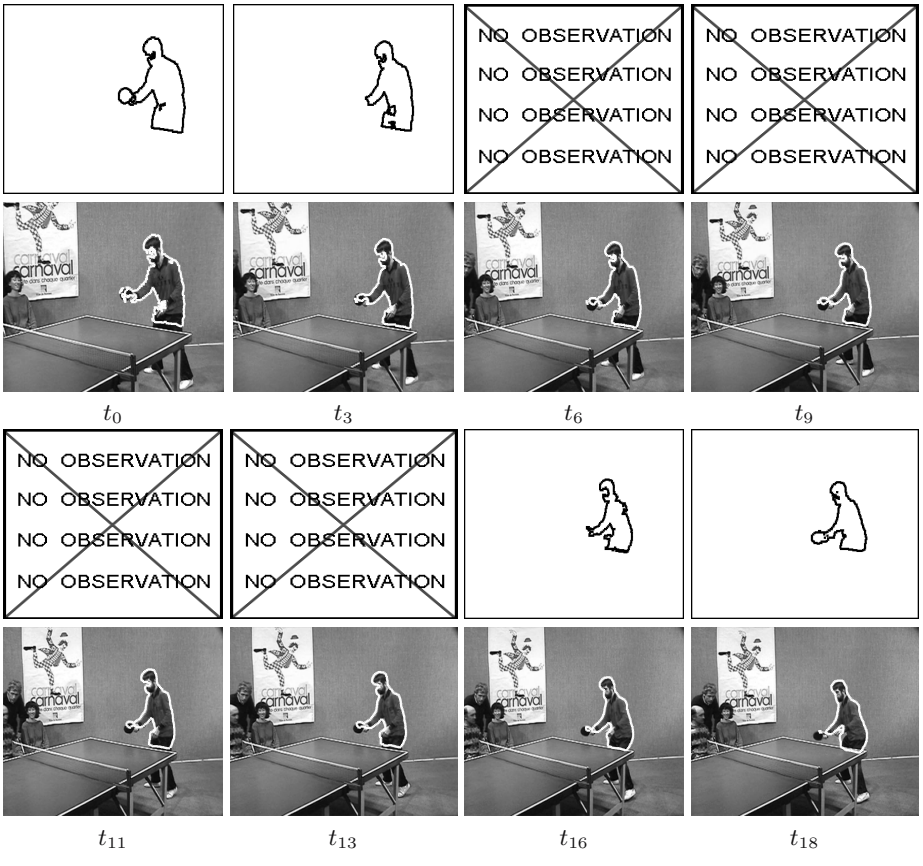


Fig. 2. Ping pong player sequence.Top row: Sample of the observed curves. Bottom row: Recovered curve superimposed on the corresponding image.

nique has been able to cope with the partial occlusion generated by the lifting of the skate ventral fin (see images going from #5 to #15 in fig. 1).

The second sequence shows a person playing ping-pong. This is a 20 frames sequence where the camera is slightly moving backward. The observed curves are here provided by a motion detection method. For this sequence, no mask were detected between frames #5 and #14. Mask contours were thus only available for frame #0 to #4, and for frames #15 to #19. It can be noticed in addition that the observed curves are locally varying a lot between two consecutive frames. For example the racket is not always recovered by the motion detection technique. We show in figure 2 a sample of the observed curves and the corresponding results. We can observe that the recovered curves follow quite well the shape of the player even in the time interval for which no observation was available.

As a last example, we show on Figure 3 results obtained on a meteorological image sequence of the Meteosat infra red channel. The observed curve is a level line at a given value within a region of interest. We aim here therefore at tracking an iso-

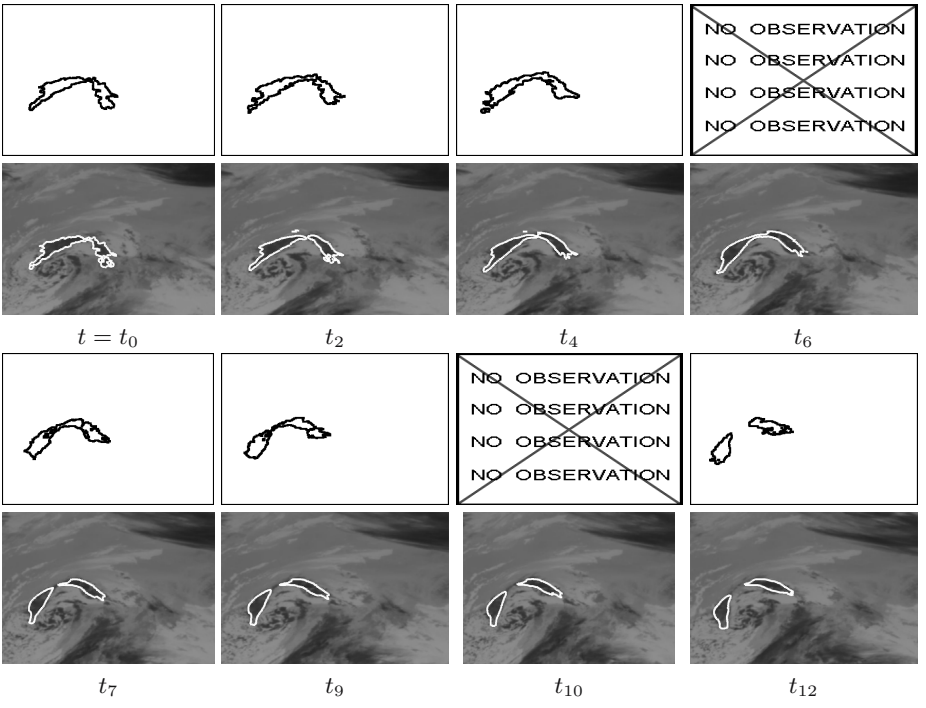


Fig. 3. Clouds sequence. Top row: Sample of observed curves. Bottom row: Recovered results superimposed on the corresponding image.

temperature curve. The results demonstrate that the technique we propose keeps the adaptive topology property of level set methods, and in the same time, incorporates a consistent temporal prior for the curve evolution.

As for the computation time of the method, our code takes less than 1 minute for a forward-backward integration of a 20 frame sequence. It has to be noticed that our code, written in C, has not been particularly optimized. For instance, the different integration considered have been performed on the whole image plane. A significant reduction of the computational load could be probably obtained considering a narrow band technique [16].

6 Conclusions

In this paper, we have presented a new technique for object contours tracking. The proposed technique allows to estimate the contours location for each time instant within the time interval of the analyzed sequence. In a similar way to a stochastic smoothing the estimation is led considering the whole set of the available measurements extracted from the image sequence. The technique is nevertheless totally different. It consists to integrate two coupled pde's representing the evolution of a state function and of an ad-

joint variable respectively. The method incorporates only few parameters. Similarly to Bayesian filtering techniques, these parameters mainly concern the definition of the different error models involved in the considered system. In our case, we have an additional parameter that weight the mean curvature motion appearing in our dynamic evolution model. The value of this parameter tunes the degree of smoothing of the curve (in our experiences it has been always fixed to the same value of 0.1).

Acknowledgments. This work was supported by the European Community through the IST FET Open FLUID project (`http://fluid.irisa.fr`).

References

1. A.F. Bennet. *Inverse Methods in Physical Oceanography*. Cambridge University Press, 1992.
2. A. Blake and M. Isard. *Active contours*. Springer-Verlag, London, England, 1998.
3. D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnoerr. Diffusion snakes: introducing statistical shape knowledge into the Mumford-Shah functionnal. *Int. J. Computer Vision*, 50(3):295–313, 2002.
4. F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus*, (38A):97–110, 1986.
5. R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Trans. on Image Processing*, 10(10):1467–1475, 2001.
6. R. Kimmel and A. M. Bruckstein. Tracking level sets by level sets: a method for solving the shape from shading problem. *Comput. Vis. Image Underst.*, 62(1):47–58, 1995.
7. M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. Conf. Comp. Vision Pattern Rec.*, 2000.
8. A.R. Mansouri. Region tracking via level set PDEs without motion computation. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(7):947–961, 2003.
9. P. Martin, P. Réfrégier, F. Goudail, and F. Guérault. Influence of the noise model on level set active contours segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(6):799–803, 2004.
10. E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. Image Processing*, 7(5):703–719, 1998.
11. M. Niethammer and A. Tannenbaum. Dynamic geodesic snakes for visual tracking. In *CVPR (1)*, pages 660–667, 2004.
12. S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
13. N. Paragios. A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE trans. on Med. Imaging*, 22(6), 2003.
14. N. Paragios and R. Deriche. Geodesic active regions: a new framework to deal with frame partition problems in computer vision. *J. of Visual Communication and Image Representation*, 13:249–268, 2002.
15. N. Peterfreund. Robust tracking of position and velocity with Kalman snakes. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(6):564–569, 1999.
16. J.A. Sethian. *Level set methods*. Cambridge University-Press, 1996.
17. O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Journ. of Roy. Meteo. soc.*, 113:1311–1328, 1987.

Implicit Free-Form-Deformations for Multi-frame Segmentation and Tracking

Konstantinos Karantzalos[1,2] and Nikos Paragios[1]

[1] ATLANTIS, CERTIS, Ecole Nationale des Ponts et Chaussees,
6-8 avenue Blaise Pascal, Cite Descartes, 77455 - Marne-La-Vallee, France
{karank, nikos.paragios}@cermics.enpc.fr
http://atlantis.enpc.fr/
[2] School of Rural and Survey Engineering, National Technical University of Athens,
Heroon Polytechniou 9, Zographou, 15780, Greece
karank@central.ntua.gr

Abstract. In this paper, we propose a novel technique to address motion estimation and tracking. Such technique represents the motion field using a regular grid of thin-plate splines, and the moving objects using an implicit function on the image plane that is a cubic interpolation of a "level set function" defined on this grid. Optical flow is determined through the deformation of the grid and consequently of the underlying image structures towards satisfying the constant brightness constraint. Tracking is performed in similar fashion through the consistent recovery in the temporal domain of the zero iso-surfaces of a level set that is the projection of the Free Form Deformation (FFD) implicit function according to the cubic spline formulation. Such an approach is a compromise between dense motion estimation and parametric motion models, introduces smoothness in an implicit fashion, is intrinsic, and can cope with important object deformations. Promising results demonstrate the potentials of our approach.

1 Introduction

Motion perception is a fundamental task of biological vision with motion estimation and tracking being the most popular and well-addressed applications. To this end, given a sequence of images, one would like to recover the 2D temporal displacement (optical flow) and the position of objects of particular interest. These applications often serve as input to high-level vision tasks, like 3D reconstruction, etc.

Dense optical flow estimation is an ill-posed problem. The problem itself is rather ill-posed since [1] the number of unknowns to be recovered is greater to the number of constraints. Such constraints are determined through the linearization of the visual or intensity preservation constraint [2]. Smoothness constraints [3] are often considered to overcome the ill-poseness of the estimation process and often lead to satisfactory results. A step further refers to the use of parametric motion estimation [4] where the motion in the entire image plane or some

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 271–282, 2005.

portions of it is represented with a linear function of the pixel coordinates. To this end, robust statistical methods [5] were considered to account for outliers in the estimation process leading to promising results [6,7] when the assumption on the motion form induced by the model is respected from the data. One can claim that parametric motion models are efficient representations of optical flow, a good compromise between low complexity and reasonable flow estimates that suffer at the object boundaries. Moreover neither the case of non-planar or objects undergoing non-rigid deformations can be addressed through such a formulation.

Tracking non-rigid objects is a task that has gained particular attention in computational vision. Starting from the pioneering formulation of the snake model [8] several attempts to address tracking through the deformation of contours can be found in the literature either model-free [9] or model-based [10]. Level set methods [11] is an established technique [12] to track moving interfaces through model-free [13] or model-based [14] methods with the advantage of being implicit, intrinsic and parameter-free. However they suffer from computational expensive processing [15] while one should preserve the form of the implicit functions through frequent re-initialization steps. Such a limitation was addressed in [16] where a finite element approach was considered to implement a level set flow.

In this paper, we introduce a higher-order polynomial approach to address dense optical flow estimation and tracking within the level set approach. To this end, we represent motion using a free form deformation of a super-imposed regular connected grid, an excellent alternative to dense motion estimation as well as to parametric motion models. Tracking is addressed through the modification of a "level set" function on the FFD space such that its projection on the image space captures the object boundaries. Visual preservation, consistence in the object appearance and smoothness constraints are used to determine the deformation of the implicit grid towards simultaneous motion estimation and tracking of objects in successive frames.

Prior art in joint optical flow estimation and tracking has mostly addressed the case of parametric (mostly affine) motion within the standard level set formulation [17,18,14,19,20]. The reminder of this paper is organized according to the following fashion; In the next section, we briefly introduce the level set method and the free form deformation model. Our variational model to recover optical flow estimations and perform tracking is described in section 3. The optimization process is presented in section 4, while experimental results and discussion are part of section 5.

2 Free Form Deformations and Implicit Level Sets

Let us consider an image:

$$\mathcal{I}(x, y) = \{(x, y) | 1 \leq x \leq X, 1 \leq y \leq Y\}$$

and a regular lattice of control points superimposed to this image:

$$\mathbf{P}_{m,n} = (\mathbf{P}^x_{m,n}, \mathbf{P}^y_{m,n});\ m = 1, ..., M, \quad n = 1, ..., N$$

One can introduce a third dimension on this grid, a discrete function $\mathbf{\Phi}(;)$, such that input image is approximated through a tensor product of Cubic B-spline:

$$\mathcal{I}(x,y) \approx \sum_{k=0}^{3}\sum_{l=0}^{3} B_k(u)B_l(v)\mathbf{\Phi}_{i+k,j+l}$$

with $i = \lfloor \frac{x}{X} \cdot M \rfloor - 1,\ j = \lfloor \frac{y}{Y} \cdot N \rfloor - 1$ and $B_k(u)$ is the k^{th} basis function of a Cubic B-spline:

$$B_0(u) = (1-u)^3/6, \quad B_1(u) = (3u^3 - 6u^2 + 4)/6$$
$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6, \quad B_3(u) = u^3/6$$

with $u = \frac{x}{X} \cdot M - \lfloor \frac{x}{X} \cdot M \rfloor$ ($B_l(v)$ is defined in a similar fashion with $v = \frac{y}{Y} \cdot N - \lfloor \frac{y}{Y} \cdot N \rfloor$). We assume that (sixteen) adjacent control points are needed to produce the observed value at any given pixel of the image. The parameters of this new representation consist of the position of the grid points and the value embedded function at these points $\Theta = (\mathbf{P}^x_{m,n}, \mathbf{P}^y_{m,n}, \mathbf{\Phi}_{m,n})$.

Furthermore one can consider a deformation of this grid (deformation of the image) starting from an initial configuration $\mathbf{P}$, and the deforming control lattice as

$$\mathbf{P}' = \mathbf{P} + \mathbf{\Delta P}$$

that can be considered as an incremental free form deformation with the deformations of the control points in both directions according to:

$$\mathbf{\Delta P} = \{(\delta\mathbf{P}^x_{m,n}, \delta\mathbf{P}^y_{m,n})\};\ (m,n) \in [1,M] \times [1,N]$$

The essence of FFD is to deform an object by manipulating a regular control lattice $\mathbf{P}$ overlaid on its volumetric embedding space. Once a deformation has been applied, the displacement of a pixel (x,y) given the deformation of the control lattice from $\mathbf{P}$ according to $\mathbf{\Delta P}$, is defined in terms of a tensor product of Cubic B-spline:

$$\begin{aligned}\mathcal{T}(\mathbf{\Delta P};(x,y)) &= ((x,y)) + \delta T(\mathbf{\Delta P};(x,y)) \\ &= \sum_{k=0}^{3}\sum_{l=0}^{3} B_k(u)B_l(v)(\mathbf{P}_{i+k,j+l} + \delta\mathbf{P}_{i+k,j+l})\end{aligned}$$

Such deformation field $\mathcal{T}(\mathbf{\Delta P};x,y)$ [21] is a popular approach in graphics, animation and rendering [22]. Opposite to optical flow techniques, FFD techniques support smoothness constraints, exhibit robustness to noise and are suitable for modelling large and small non-rigid deformations. Furthermore, under certain conditions, it can support a dense registration paradigm that is continuous and guarantees a one-to-one mapping.

The level set method [11] consists of representing and evolving an evolving interface $\partial\mathcal{R}(p)$ with the zero-level set of an embedding surface Φ. Such representation can lead to a natural handling of changing the topology of $\partial\mathcal{R}(p)$. Numerical simulations on Φ may be developed trivially and intrinsic geometric

properties of the evolving interface can be estimated directly from the level set function.

Let $\phi : \Omega \rightarrow \mathcal{R}^+$ be a Lipschitz function that refers to a level set representation:

$$\phi(p;t) = \begin{cases} 0 & , p \in \partial\mathcal{R}(t) \\ +\mathcal{D}((p), \partial\mathcal{R}(t)) > 0 & , p \in \mathcal{R}(t) \\ -\mathcal{D}((p), \partial\mathcal{R}(t)) < 0 & , p \in [\Omega - \mathcal{R}(t)] \end{cases} \quad (1)$$

where Ω is the image domain (bounded) and $\mathcal{D}(p, \partial\mathcal{R}(t))$ is the minimum Euclidean distance between the pixel p and the interface $\partial\mathcal{R}(t)$. Then, the level set formulation can be considered as an optimization framework. To this end, one can define the approximations of DIRAC and HEAVISIDE distributions [23,24]:

$$\delta_a(\phi) = \begin{cases} 0, & |\phi| > \alpha \\ \frac{1}{2\alpha}\left(1 + \cos\left(\frac{\pi\phi}{a}\right)\right), & |\phi| < \alpha \end{cases}$$

$$H_\alpha(\phi) = \begin{cases} 1, & \phi > \alpha \\ 0, & \phi < -\alpha \\ \frac{1}{2}\left(1 + \frac{\phi}{\alpha} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{a}\right)\right), & |\phi| < \alpha \end{cases} \quad (2)$$

These functions can be used to define contour-based as well as region-based energetic modules for the evolving interface in the level set space [23]:

$$(i) \underbrace{\iint_\Omega H_\alpha(\phi(p)) r_1(I(p)) dxdy}_{regional\ module}, \quad (ii) \underbrace{\iint_\Omega \delta_\alpha(\phi(p)) b(I(p)) |\nabla\phi(p)| dxdy}_{boundary\ module}$$

where r and b are *region* and *boundary* positive monotonically decreasing data-driven functions. The first term [i] is a grouping component that accounts for some regional properties (modulo the definition of r) of the area defined by the evolving interface. The second term [ii] is a combination of a boundary attraction term (modulo the definition of b) and a smoothness component [25,26].

Within the selected representation, one can consider a function $\mathbf{\Phi}$ defined at the lattice $\mathbf{P}$ to be a level set function, if

$$\phi(x, y) = \sum_{k=0}^{3}\sum_{l=0}^{3} B_k(u) B_l(v) \mathbf{\Phi}_{i+k, j+l}$$

and

$$\phi(x, y) = \begin{cases} 0 & , p \in \partial\mathcal{R}(t) \\ +\mathcal{D}((p), \partial\mathcal{R}(t)) > 0 & , p \in \mathcal{R}(t) \\ -\mathcal{D}((p), \partial\mathcal{R}(t)) < 0 & , p \in [\Omega - \mathcal{R}(t)] \end{cases}$$

One now can use such a formulation to encode motion estimation and tracking. Motion is represented with the deformation of the original lattice $\mathbf{P}$ while tracking will be addressed through the evolution of a "level set function" $\mathbf{\Phi}$ defined on the same lattice.

3 Optical Flow Estimation

Optical flow estimation is equivalent with recovering a pixel-wise deformation field $\mathcal{T}(\mathbf{\Delta P}; x, y)$ that creates visual correspondences between two consecutive images f and g. Optical flow estimation within FFD is now equivalent with finding the best lattice $\mathbf{P}$ configuration such that the overlaid structures (images) coincide. One can consider the Sum of Squared Differences (SSD) as the data-driven term to recover the deformation field $\mathcal{T}(\mathbf{\Theta}; \mathbf{x})$;

$$E_{data}(\mathbf{\Theta}) = \iint_{\Omega} \big(f(\mathbf{x}) - g(\mathcal{T}(\mathbf{\Delta P}; x, y))\big)^2 dxdy$$

Such an error norm is very sensitive to occlusions as well as to outliers and therefore it can be replaced with a robust estimator, or like an an M-estimator. Such a method assigns weights to the constraints at the pixel level that are disproportional to their residual error therefore rejecting the motion outliers. to this end, one should define the influence function, $\psi(x)$ like for example the Tukey's estimator:

$$\rho(x) = \begin{cases} x(K_\sigma - x) \text{ if } |x| < K_\sigma \\ 0 \qquad\quad \text{otherwise} \end{cases}$$

where K_σ characterizes the shape of the robust function and is updated at each iteration leading to the following cost function:

$$E_{data}(\mathbf{\Delta P}) = \iint_{\Omega} \rho(r)\ dxdy = \iint_{\Omega} \rho(f(\mathbf{x}) - g(\mathcal{T}(\mathbf{\Delta P}; x, y)))\ dxdy$$

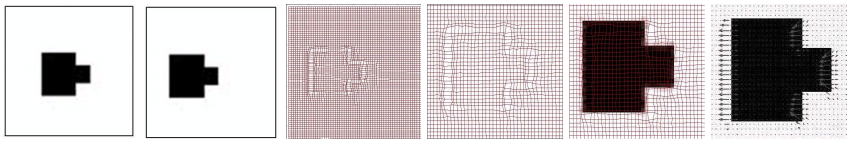
While such a model can be quite efficient it still suffers from the aperture problem. One can consider additional constraints to the constant brightness assumption like the gradient preservation assumption, recently introduced in [27] leading to the following cost function;

$$\begin{aligned} E_{data}(\mathbf{\Delta P}) = \alpha \iint_{\Omega} \rho\big(f(\mathbf{x}) - g(\mathcal{T}(\mathbf{\Delta P}; x, y))\big) dxdy \\ +\beta \iint_{\Omega} \rho\big(\big|\big|\nabla f - \nabla g(\mathcal{T}(\mathbf{\Delta P}; x, y))\big|\big|\big) dxdy \end{aligned}$$

a constraint that improves the estimation of the optical flow on the object boundaries where the visual constancy assumption is often violated.

The use of thin plate splines to represent motion introduces in an implicit form some smoothness constraint that can deal with a limited level of deformation. In order to account for outliers and noise, one can replace the error-two norm with more appropriate robust metrics [5]. In order to further preserve the regularity of the recovered motion flow, one can consider an additional smoothness term on the deformation field $\delta\mathbf{P}$. We consider a computationally efficient smoothness term:

$$E_{smooth}(\mathbf{\Delta P}) = \iint \Big(|T_x(\mathbf{\Delta P}; x, y)|^2 + |T_y(\mathbf{\Delta P}; x, y)|^2\Big)\ dxdy$$



(a) (b) (c) (d) (e) (f)

Fig. 1. Binary case with a global one pixel movement to the left : (a) first frame f, (b) second frame g, (c) deformed FFD grid, (d) Zoom on deformed grid, (e) Zoom on deformed grid overlaid to first image (h)Zoom on grid's flow overlaid to second image

Such smoothness term is based on a classic error norm that has certain known limitations. Within the proposed framework, an implicit smoothness constraint is also imposed by the Spline FFD. Therefore there is not need for introducing complex and computationally expensive regularization components.

Then the global deviations from the data-driven term and the smoothness constraints term can now be integrated to define an objective function that upon optimization will provide a smooth motion field that establishes correspondences between the two images:

$$\begin{aligned} E_{flow}(\mathbf{\Delta P}) = \alpha \iint_{\Omega} \rho\big(f(\mathbf{x}) - g(\mathcal{T}(\mathbf{\Delta P}; x, y))\big) dxdy \\ +\beta \iint_{\Omega} \rho\big(\big|\big|\nabla f - \nabla g(\mathcal{T}(\mathbf{\Delta P}; x, y))\big|\big|\big) dxdy \\ +\gamma \iint \left(|T_x(\mathbf{\Delta P}; x, y)|^2 + |T_y(\mathbf{\Delta P}; x, y)|^2\right) \; dxdy \end{aligned}$$

Multilevel Incremental Free-Form Deformation (MIFFD): A straightforward application of the FFD manipulation cannot always guarantee the successful motion estimation between the two images. One reason for this is that we limit the maximum displacement of a control point to approximately a half of the spacing between control points in order to make the deformation function one-to-one. The correspondences that each time can be caught are according to what level (how coarse or fine) of the FFD's grid has been chosen. Here, we present the MIFFD technique that overcomes the drawbacks of the straightforward method, since it can handle both large and small non-rigid deformations. Multiresolution control lattices are used according to a coarse-to-fine strategy. From a coarser level of the control lattice that can deal better with large displacements we proceed continuously to a finer level. At each level, we can solve for the incremental deformation of the control lattice using the scheme presented in the previous section. In the end, the overall dense deformation field for motion estimation is defined by these incremental deformations from all levels.

Let $\mathbf{P}^1, ..., \mathbf{P}^K$ denote a hierarchy of control point meshes at different resolutions. Each control mesh $\mathbf{P}^k$ and the associated spline-based FFD defines a transformation $\mathcal{T}^k(\mathbf{\Delta P}; x, y)$ at each level of resolution and the total deformation $\delta\mathcal{T}(x, y)$ for a pixel (x, y) in a hierarchy of K levels is:

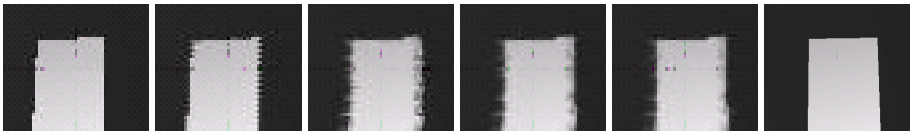


Fig. 2. Multilevel Optical Flow estimation MIFFD (4 levels). From the second image g (top - left), the reconstructed images from the estimated flow are shown, until the first image f (last one) is approximated. Final Energy 7% of initial Energy.

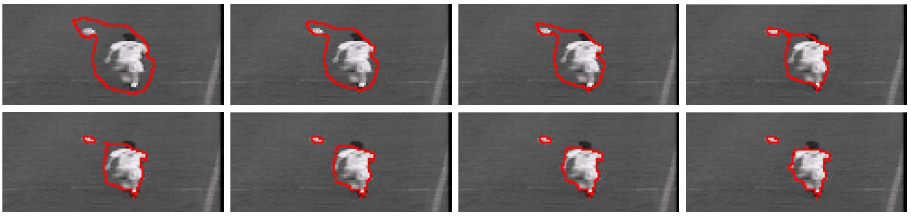


Fig. 3. Curve Propagation on 006 frame of player sequence

$$\delta\mathcal{T}(x,y) = \sum_{k=0}^{K} \delta\mathcal{T}^k(\mathbf{\Delta P}^{\mathrm{k}}; x, y)$$

The hierarchy of control lattices can have arbitrary number of levels, but typically 3-4 levels are sufficient to handle both large and small deformations. Such an optimization will lead to successful estimation of the motion field but does not address tracking.

Let us consider without loss of generality that an object is present in the scene. The task of tracking consists of recovering the successive positions of a planar curve $\gamma(;)$ such that the object is properly delineated in time. In order to address this demand we consider a level set curve to represent objects.

4 Object Tracking

Tracking is performed through the consistent recovery in the temporal domain of the zero iso-surfaces of a level set $\gamma(\mathbf{\Delta P})$ that is the projection of the FFD implicit function according to the cubic spline formulation.

Based on region-driven model free image segmentation techniques, objects boundaries are approached through a curve propagation technique (Figure 4). The essence of this approach is to optimize the position and the geometric form of the curve by measuring information along that curve, and within the regions that compose the image partition.

To this end, one can assume without loss of generality that objects are uniform that is also the case for the background. In that case, given an initial position of the curve, one can determine global region-driven $r_{obj}(f)$ and $r_{bg}(f)$ functions provide a statistical description of the inside and outside object area:

(a) (b) (c) (d) (e) (f)

Fig. 4. Tracking result on Test Sequences: (a) Initial contour on first image f of Figure 1 sequence, (b) Curve propagation result on same image, (c) Object boundaries after applying the transformation of MIFFD flow, (d) Initial contour on first image f of Figure 2 sequence, (e) Curve propagation result on same image, (h)Object boundaries after applying the transformation of MIFFD flow.

$$r_{obj}(f(x,y)) = \frac{(\mu_{obj} - f(x,y))^2}{\sigma^2_{obj}}, \quad r_{bg}(f(x,y)) = \frac{(\mu_{bg} - f(x,y))^2}{\sigma^2_{bg}}$$

where μ_{obj} is the mean and σ_{obj} the covariance matrix of the object appearance (similar definition for the background). In cases where the assumption of Gaussian densities seems unrealistic one can consider a more flexible parametric density function - gaussian mixture - to describe the visual properties of the object and the background.

In the case of static images, one can perform object extraction through the separation of image pixels according to their match with the expected appearance properties of the object and the background. Such an optimization can be considered on the lattice space, that is

$$E_{object}(\mathbf{\Phi}) = \alpha \iint \delta \left(\Phi \left(\sum_{k=0}^{3} \sum_{l=0}^{3} B_k(u) B_l(v) \mathbf{\Phi}_{i+k,j+l} \right) \right) \left| \nabla \sum_{k=0}^{3} \sum_{l=0}^{3} B_k(u) B_l(v) \mathbf{\Phi}_{i+k,j+l} \right| d\Omega$$
$$+\beta \iint_{\Omega} H \left(\sum_{k=0}^{3} \sum_{l=0}^{3} B_k(u) B_l(v) \mathbf{\Phi}_{i+k,j+l} \right) r_{obj}(f(x,y)) d\Omega$$
$$+\beta \iint_{\Omega} \left[1 - H \left(\sum_{k=0}^{3} \sum_{l=0}^{3} B_k(u) B_l(v) \mathbf{\Phi}_{i+k,j+l} \right) \right] r_{bg}(f(x,y)) d\Omega$$

where the first term imposes smoothness constraints while the second address a background/object separation according to the expected visual properties of the two class. One now can consider the separation of the object/background in both frames f and g given the deformation of the grid through the FFD one can address tracking through the minimization of

$$E_{tracking}(\mathbf{\Delta P}, \mathbf{\Phi}) = \alpha \iint \delta\left(\Phi\left(\sum_{k=0}^{3}\sum_{l=0}^{3} B_k B_l \mathbf{\Phi}_{i+k,j+l}\right)\right) \left|\nabla \sum_{k=0}^{3}\sum_{l=0}^{3} B_k B_l \mathbf{\Phi}_{i+k,j+l}\right| d\Omega$$
$$+\beta \iint_{\Omega} H\left(\sum_{k=0}^{3}\sum_{l=0}^{3} B_k B_l \mathbf{\Phi}_{i+k,j+l}\right) r_{obj}(f(x,y)) d\Omega$$
$$+\beta \iint_{\Omega} \left[1 - H\left(\sum_{k=0}^{3}\sum_{l=0}^{3} B_k B_l \mathbf{\Phi}_{i+k,j+l}\right)\right] r_{bg}(f(x,y)) d\Omega$$
$$+\beta \iint_{\Omega} H\left(\sum_{k=0}^{3}\sum_{l=0}^{3} B_k B_l \mathbf{\Phi}_{i+k,j+l}\right) r_{obj}(g(\mathcal{T}(\mathbf{\Delta P}; x, y))) d\Omega$$
$$+\beta \iint_{\Omega} \left[1 - H\left(\sum_{k=0}^{3}\sum_{l=0}^{3} B_k B_l \mathbf{\Phi}_{i+k,j+l}\right)\right] r_{bg}(g(\mathcal{T}(\mathbf{\Delta P}; x, y))) d\Omega$$

where α, β are constant coefficients and the assumption that the object/background properties do not change from one frame to the next. One can relax this constraint through the estimation of visual descriptors in both frames.

Such a tracking term can be integrated with the optical flow estimation term to simultaneously address dense optical flow estimation and object tracking.

$$E(\mathbf{\Delta P}, \mathbf{\Phi}) = E_{flow}(\mathbf{\Delta P}) + E_{tracking}(\mathbf{\Delta P}, \mathbf{\Phi})$$

The lowest potential of this cost function will provide visual correspondences between the two images, and recover optimal successive positions of objects in time [Figure 5 and 6].

5 Implementation

The calculus of variations and a gradient descent method can be used to optimize such an objective function. A minimizer must fulfill the Euler-Lagrange equation both in the deformation space [$\mathbf{\Delta P}$] as well as in the implicit space [$\mathbf{\Phi}$];

$$\frac{\partial}{\partial \mathbf{\Delta P}} E(\mathbf{\Delta P}, \mathbf{\Phi}) = 0, \quad \frac{\partial}{\partial \mathbf{\Phi}} E(\mathbf{\Delta P}, \mathbf{\Phi}) = 0$$

One can further develop these conditions using the chain rule;

$$\frac{\partial}{\partial \mathbf{\Delta P}} E(\mathbf{\Delta P}, \mathbf{\Phi}) = \frac{\partial E_{flow}(\mathbf{\Delta P})}{\partial \mathbf{\Delta P}} + \frac{\partial E_{tracking}(\mathbf{\Delta P}, \mathbf{\Phi})}{\partial \mathbf{\Delta P}}$$

while in the case of the implicit FFD level set the flow consists only one term;

$$\frac{\partial}{\partial \mathbf{\Phi}} E(\mathbf{\Delta P}, \mathbf{\Phi}) = \frac{\partial E_{tracking}(\mathbf{\Delta P}, \mathbf{\Phi})}{\partial \mathbf{\Phi}}$$

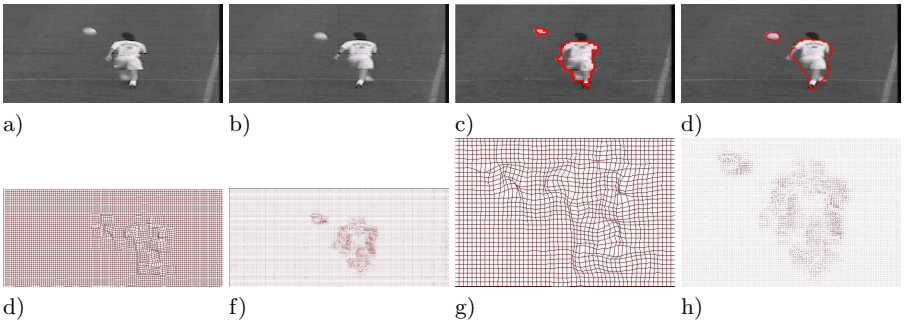


Fig. 5. Player Sequence Recovery (frames R006 -R007): (a) first image f (R006), (b) second image g (R007), (c) Object boundaries on f image from Level Set propagation, (d) Object boundaries on g image after applying the transformation of MIFFD flow, (e) Deformed Grid, (f) Deformed Grid's Flow, (g) Zoom on deformed grid, (h) Zoom on deformed grid's flow

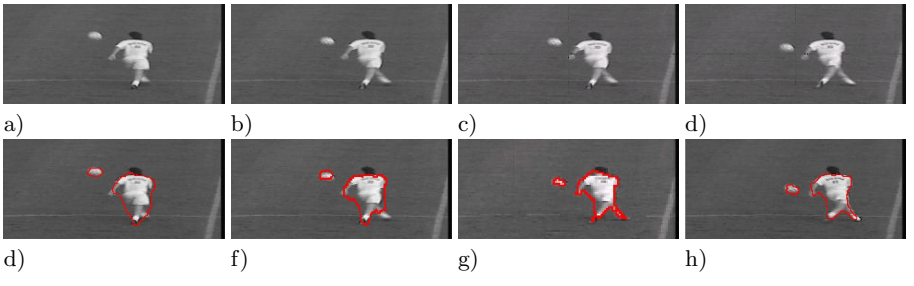


Fig. 6. Player Sequence: Recovered Frames R007-R008 and R009-R010: (a) frame R007, (b) frame R008, (c) Object boundaries from previous recovered frames (Figure 5), (d) Object boundaries after applying the transformation of MIFFD flow, (e) frame R009, (f) frame R010, (g) Object boundaries from Level Set propagation, (h) Object boundaries after applying the transformation of MIFFD flow

In practice, the proposed framework works in the following fashion. Given an initial contour, the implicit level function is estimated in the lattice space. Then, in parallel one updates the motion parameters of the process as well as deforming the contour. To this end, an adaptive estimation of the regional descriptors is considered as well as frequent re-initializations of the lattice implicit function. Upon a steady state solution, the lattice deformations as well as the object positions are recovered in successive frames. Such positions are used to initialize the process in the next couple of frames and the process is repeated until convergence.

6 Discussion

In this paper we have presented a novel algorithm to optical flow estimation and tracking. Our approach introduces the concept of joint motion estimation

and tracking in superimposed spaces of higher order polynomials like thin plane spline level set. The selected representation of motion guarantees one-to-one correspondences, smoothness on the deformation field and is of low complexity. Parallel to that we address tracking through the recovery of explicit correspondences between the object temporal positions in the level set space that is implicit, intrinsic and parameter free. Promising results, as shown in Figure 5 and 6, demonstrate the potentials of the proposed formulation that address in a simultaneous fashion dense optical flow estimation and non-rigid tracking. Classical optical flow test sequences like the Yosemite sequence can not be used for validation because there exist not an apparent object for tracking.

One can consider numerous extensions of the method. The use of FFD that also encode the structure of the image is a prominent one. The grid that was considered to represent motion has a fixed topology and the motion of each image pixel is reproduced using the same number of neighboring elements that are distributed according to the same topology. One can consider modifying the grid dependencies and connections according to the image structure. In terms of tracking, the case of multiple objects is to be addressed. Within the proposed framework one can consider the one-to-one constraint on the correspondences and preserve topology or relax such a constraint to address topological changes from one image to the next. Such a perspective is to be investigated. Last, but not least the use of a 3D deformation grid can be considered to account for motion decomposition in layers.

References

1. J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12:43–77, 1994.
2. B. Horn and B. Schunck. Determinating Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.
3. J. Weickert and C. Schn orr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14:245–255, 2001.
4. B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
5. P. Huber. *Robust Statistics*. John Wiley & Sons, 1981.
6. M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:973–986, 1996.
7. J-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6:348–365, 1995.
8. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. In *IEEE International Conference in Computer Vision*, pages 261–268, 1987.
9. M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *European Conference on Computer Vision*, volume I, pages 343–356, 1996.

10. T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.
11. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
12. S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003.
13. N. Paragios and R. Deriche. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:266–280, 2000.
14. D. Cremers. A Variational Framework for Image Segmentation Combining Motion Estimation and Shape Regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 53–58, 2003.
15. J. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
16. M. Weber, A. Blake, and R. Cipolla. Sparse Finite Elements for Geodesic Contours with Level-Sets. In *European Conference on Computer Vision*, pages 391–404, 2004.
17. N. Paragios and R. Deriche. Unifying Boundary and Region-based Information for Geodesic Active Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II:300–305, 1999.
18. A. Yezzi, L. Zollei, and T. Kapur. A Variational Framework for Joint Segmentation and Registration. In *IEEE Mathematical Methods in Biomedical Image Analysis*, pages 44–51, 2001.
19. I. Dydenko, D. Friboulet, and I. Magnin. A Variational Framework for Affine Registration and Segmentation with Shape Prior: Application in Echocardiographic Imaging. In *IEEE Workshop in Variational and Level Set Methods*, pages 209–217, 2003. Faugeras, O. and Paragios, N. (eds).
20. A. Yilmaz, X. Li, and B. Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 2004.
21. T. Sederberg and S. Parry. Free-Form Deformation of Solid Geometric Models. In *ACM SIGGRAPH*, volume 4, pages 151–160, 1986.
22. P. Faloutsos, M. van de Panne, and D. Terzopoulos. Dynamic Free-Form Deformations for Animation Synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 3:201–214, 1997.
23. H-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 127:179–195, 1996.
24. T. Chan and L. Vese. Active Contours without Edges. *IEEE Transactions on Image Processing*, 10:266–277, 2001.
25. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. In *IEEE International Conference in Computer Vision*, pages 694–699, 1995.
26. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *IEEE International Conference in Computer Vision*, pages 810–815, 1995.
27. T. Brox, A. Bruhn, and J. Weickert. High accuracy optical flow estiation based on a theory for warping. In *European Conference on Computer Vision*, pages 158–163, 2004.

A Surface Reconstruction Method for Highly Noisy Point Clouds*

DanFeng Lu[1], HongKai Zhao[2], Ming Jiang[1], ShuLin Zhou[1], and Tie Zhou[1]

[1] LMAM, School of Mathematical Sciences, Peking Univ.
[2] Department of Mathematics, University of California, Irvine
zhao@math.uci.edu

Abstract. In this paper we propose a surface reconstruction method for highly noisy and non-uniform data based on minimal surface model and tensor voting method. To deal with ill-posedness, noise and/or other uncertainties in the data we processes the raw data first using tensor voting before we do surface reconstruction. The tensor voting procedure allows more global and robust communications among the data to extract coherent geometric features and saliency independent of the surface reconstruction. These extracted information will be used to preprocess the data and to guide the final surface reconstruction. Numerically the level set method is used for surface reconstruction. Our method can handle complicated topology as well as highly noisy and/or non-uniform data set. Moreover, improvements of efficiency in implementing the tensor voting method are also proposed. We demonstrate the ability of our method using synthetic and real data.

1 Introduction

Surface reconstruction is to retrieve the original surface from the partial information of that surface. The partial information can include points, pieces of curves and surfaces. In our paper, we mainly consider reconstruction from unorganized point clouds. Surface reconstruction is an important task in many applications such as computer vision, computer graphics, medical imaging, computer aided design and scientific computing.

The main difficulties of surface reconstruction from point clouds include unknown connection or ordering information among the data points, unknown topology of the original surface, and noise and/or non-uniformity in the data. Based on different representations of reconstructed surfaces, most previous reconstruction approaches can be classified as parametric or non-parametric (implicit surfaces). One parametric approach is NURBS (Non-Uniform Rational

* H. Zhao is partially supported by ONR, DARPA and Sloan Fellowship. M. Jiang is partially supported by the National Basic Research Program of China under Grant 2003CB716101, National Science Foundation of China under Grants 60325101, 60272018 and 60372024, and Engineering Research Institute, Peking University. S. Zhou and T. Zhou are partially supported by the National Basic Research Program of China under Grant 2003CB716101, National Science Foundation of China under Grant 60372024, and Engineering Research Institute, Peking University.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 283–294, 2005.

B-Spline) [1] in which the reconstructed surface is smooth, and the data set can be non-uniform. However, this method require a nice parameterization of the surface and possible patching of different pieces for the reconstruction, which can be difficult for an arbitrary data set. Also, it is difficult to treat noisy data. Another popular computational geometry algorithm is based on Delaunay triangulations and Voronoi diagrams to construct triangulated surfaces [2, 3, 4, 5, 6]. For this kind of method, it is challenging to find the right connections among all data points in three and higher dimensions, especially for noisy and highly non-uniform data. Implicit surface methods try to find an implicit function such that a particular level set of this function fits the data best and is extracted as the reconstructed surface [7, 8, 9, 10, 11, 12, 13, 14, 15]. Implicit methods usually have topological flexibility, a simple data structure and depth/volumetric information. However it is a challenge to deal with open surfaces.

To deal with noisy data a variational formulation is usually used and is composed of both a fitting term for the data and a regularization term for the reconstructed surface. There are two issues for this approach: (1) all data points, even outliers, are treated equally and can affect the final reconstruction; (2) there is a lack of effective communications among all data points and the balance of the fitting term and the regularization term is usually local during the reconstruction/evolution which can cause the evolving surface trapped into local minimum easily. For highly noisy data, these approaches will likely to fail.

Tensor voting method, proposed by Medioni et al. [16], is a nice feature extraction algorithm. By designing an appropriate voting procedure among all data points a tensor field and an associated saliency field can be constructed. Coherent geometric information can be extracted from the tensor field and the saliency field. However, using tensor voting method alone is difficult to reconstruct a smooth and well-represented surface.

In this paper, we propose a surface reconstruction method combining the minimal surface model [17, 18] and the tensor voting method for highly noisy and/or non-uniform data. We use tensor voting method to preprocess the noisy data as well as to provide coherent information for the minimal surface model. We show that our model can handle significant noise in the data.

2 New Surface Reconstruction Model

2.1 Minimal Surface Model

In [13] the following weighted minimal surface model is proposed: let $\mathcal{S}$ denote the data set which can include points, pieces of curves and surfaces. Define

$$d(x) = \text{dist}(x, \mathcal{S}) \tag{1}$$

to be the distance function to $\mathcal{S}$. Then define the surface energy functional as:

$$E(\Gamma) = \left[\int_{\Gamma} d^p(x) ds \right]^{1/p} . \tag{2}$$

Here Γ is an arbitrary surface and ds is the surface area. Thinking of $d(x)$ as a potential function for $\mathcal{S}$, this energy is the L^p norm of potential on Γ. The purpose is to try to find a local minimizer of the energy functional that behaves like a minimal surface or an elastic membrane attached to the data set.

Level set method [19] is used to evolve an initial guess to the steady state. Define the corresponding level set function to be $\phi(x,t)$. The energy functional can be reformulated as:

$$E(\phi) = \left[\int d^p(x)\delta(\phi(x))|\nabla\phi(x)|dx\right]^{1/p} , \tag{3}$$

where the integration domain can be any open set (e.g., the computation domain) that contains the zero level set of ϕ.

The gradient flow for the level set function $\phi(x,t)$ ([13, 20]) is:

$$\frac{\partial\phi}{\partial t}=|\nabla\phi|\left[\int d^p(x)\delta(\phi)|\nabla\phi|dx\right]^{1/p-1}\times d^{p-1}(x)\left[\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|}+\frac{1}{p}d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right]. \tag{4}$$

By neglecting a scaling factor we can simplify (4) as:

$$\frac{\partial\phi}{\partial t} = |\nabla\phi|\left[\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|}+\frac{1}{p}d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right] . \tag{5}$$

The term $\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|}$ corresponds to the attraction by the distance field and the term $d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$ corresponds to a minimal surface regularization weighted by the distance function, where $\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$ is the mean curvature of the surface. The parameter $1/p$ balance the potential force and surface tension. Since the nonlinear regularization due to surface tension has a desirable scaling $d(x)$, the membrane is more flexible close to the data and is more rigid away from the data. Fast implementations were discussed in [14]. The minimal surface model can handle complicated topologies and construct a surface that is smoother than triangulated surface in three dimensions. It can deal with noisy and non-uniform data to some extent by balancing the attraction of the data (fitting) and the surface area regularization. However, the minimal surface model can not deal with highly noisy data because (1) The distance field is the distance to all data set. If there are many outliers, the evolution surface will be attracted by all these points and get stuck. (2) There is a lack of global communications or denoising for the noisy data set. The surface regularization (the curvature) term is very local and is only related to the evolution surface not to the data set.

2.2 Tensor Voting Method

Tensor voting method [16] allows more effective and robust communications among the data to extract coherent geometric features and saliency. A second order symmetric tensor is used to store geometric information, orientation information and saliency. The tensor can be visualized as an ellipse in 2D, and an

ellipsoid in 3D. The shape of the tensor defines the geometric information (point, curve, or surface element), and its size represents the saliency. In 3D, a surface is represented by a tensor in the shape of an elongated ellipsoid (stick tensor) with its major axis along the surface normal. A curve is represented by a tensor in the shape of a flat ellipsoid (plate tensor) that is perpendicular to the curve's tangent. An isolated point has no orientation preference and is represented by a tensor in the shape of a spherical ellipsoid (ball tensor). The tensor field is generated by a voting procedure.

We give a brief review of the basic idea behind tensor voting in 2D. Suppose there exists a smooth curve connecting the origin O and a point P and suppose that the normal to the curve at O is known. Then what is the most likely normal direction at P? Fig. 1 illustrates the situation. It can be argued [16] that the osculating circle connecting O and P is the most likely connection since it keeps the curvature constant along the hypothesized circular arc. So the most likely normal is given by the normal to the circular arc at P (thick black arrow in Fig. 1). This normal at P is oriented such that its inner product with the normal at O is nonnegative. The length of this normal, which represents the voting strength, is inversely proportional to the arc length s and curvature k. So the decay function of vote strength is defined as:

$$DF(s, \kappa, \sigma) = e^{-\frac{s^2 + c\kappa^2}{\sigma^2}} \quad , \tag{6}$$

where σ controls smoothness, which also determines the effective neighborhood size [21]. c is a constant which controls the decay with high curvature, and about its value we refer readers to [22]. We here set $c = 3.57$. If we vote for all locations of P, we get a 2D stick voting field from O.

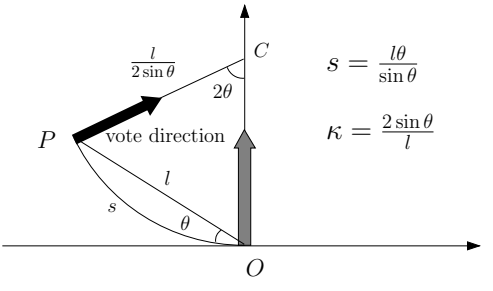


Fig. 1. Voting between two points

We denote the stick vote at P receiving from O as $[v_x \; v_y]^T$. Other points cast votes to P as well. So at P we can get a symmetric positive tensor by summing all votes received:

$$S = \begin{bmatrix} \sum v_x^2 & \sum v_x v_y \\ \sum v_y v_x & \sum v_y^2 \end{bmatrix} \quad . \tag{7}$$

Let the two eigenvalues of S be $\lambda_1 \geq \lambda_2 \geq 0$ and two corresponding eigenvectors be $\hat{e}_1$ and $\hat{e}_2$. S can be rewritten as

$$S = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^T + \lambda_2(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T) \ . \tag{8}$$

$\hat{e}_1\hat{e}_1^T$ is called a 2D stick tensor $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T$ is called a 2D ball tensor. The stick saliency field $\lambda_1 - \lambda_2$ indicates the saliency of curve. The larger the difference the more likely P is on a curve whose normal is $\hat{e}_1$. Here we give an example of stick saliency field in 2D for eight points from a circle in Fig. 2. The saliency field gives a good indication of the circle.

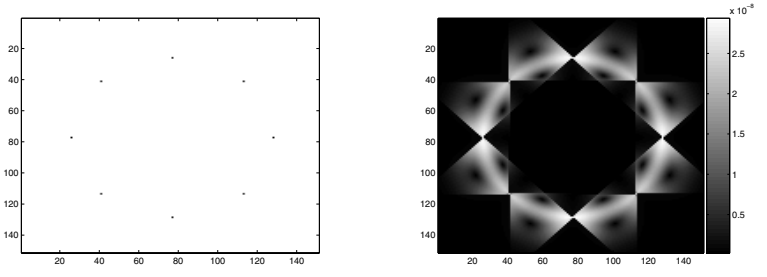


Fig. 2. The stick saliency field for eight points from a circle

If the input is discrete points, each point has a ball tensor initially. A tensor field is generated by a tensor voting procedure. For every pair of points, for example O voting on P in the previous example, since no direction information is available initially, a discrete set of uniformly distributed directions are used as possible normals at O and vote on P with the weight function in (6). By summing votes from all directions at all other points we can get a tensor at each data point and we can further generate a tensor field.

In 3D a tensor field can be decomposed as

$$S = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^T + (\lambda_2 - \lambda_3)(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T) + \lambda_3(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T) \ , \tag{9}$$

where $\hat{e}_1\hat{e}_1^T$ is a 3D stick tensor, $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T$ is a 3D plate tensor, $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T$ is a 3D ball tensor. The stick saliency field $\lambda_1 - \lambda_2$ represents the saliency of surface with normal $\hat{e}_1$, the field of $\lambda_2 - \lambda_3$ represents the saliency of curve with tangent direction orthogonal to both $\hat{e}_1$ and $\hat{e}_2$, and λ_3 represents the saliency of junction or isolated point. We denote the stick saliency field $\lambda_1 - \lambda_2$ as $s(x)$ which will play an important role in our new model.

In summary, we use the following tensor voting procedure for our data points:

1. Tensor calculus. Generate the initial stick tensor information at every original data point.
2. Voting process. Every data point propagates its tensor information to neighboring grid points and generate a tensor field in the computation domain.
3. Feature extraction. Extract the geometric features and saliency field at each grid point.

Although tensor voting method can reveal coherent geometric features and saliency for data points, to use the method directly for surface reconstruction has the following disadvantages: (1) The tensor information is not very accurate or sharp, especially in the case of complicated topology and/or geometry. Surface regularization after reconstruction is needed [23]. (2) To extract the surface from the saliency field directly the algorithm is quite complex, and need to tune parameters of threshold empirically [16, 23].

2.3 Our New Model

The above discussions show that the stick saliency field from tensor voting among all data points contains more global and robust information for our surface reconstruction. The strength of the saliency field gives a good likelihood indication of surface at each point. Therefore we incorporate the stick saliency field and combine it with the distance field for surface reconstruction. Let

$$k(x) = 1 - \frac{s(x)}{M}, \quad M = \max\{s(x)\} \tag{10}$$

be the normalized stick saliency field. Then we define our new evolution as:

$$\frac{\partial \phi}{\partial t} = \alpha \nabla d(x) \cdot \nabla \phi + \beta \nabla k(x) \cdot \nabla \phi + \gamma d(x) |\nabla \phi| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \ . \tag{11}$$

The term $\nabla d(x) \cdot \nabla \phi$ corresponds to the attraction of the data set through the distance field. The term $\nabla k(x) \cdot \nabla \phi$ corresponds to attraction of saliency field. These two terms advect the surface closer to the data set as well as to high saliency region. The term $d(x)|\nabla \phi| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$ corresponds to a weighted surface tension which regularizes the reconstructed surface. Tuning the parameters α, β can balance the effect of two fields, and the value of γ affect the smoothness of the reconstructed surface. For highly non-uniform data, the saliency field can provide more useful information than the distance field.

For highly noisy data sets, we first use tensor voting to remove outliers, i.e., those points that are not likely on the surface. After we get the normalized stick saliency field we remove those points whose saliency value is smaller than a threshold. This procedure can be repeated if necessary. This preprocessing step allows us to clean up the data substantially even for very noisy data, which will be shown later by examples. After this step we redo the tensor voting procedure for the remaining data set and generate a new tensor and saliency field. Then we use the above model for surface reconstruction.

We do not advocate of only using saliency field to propagate the surface. In some situations with simple topological structure and surface details, we can solely use saliency field, e.g., Fig. 5(c). However, the global tensor voting process usually results in a quite smeared saliency field, i.e., the gradient of the saliency field is not sharp, especially if the data set is sparse or has noise or complicated topological or geometric structures. Moreover, this makes the evolution slow too.

3 Numerical Implementation

3.1 The Level Set Method for Surface Evolution

Since we do not know a priori the topology of the final surface, we use level set method for surface evolution according to (11). This equation is of the same type of the minimal surface model used in [17]. The two convection terms $\nabla k(x) \cdot \nabla \phi(x)$ and $\nabla k(x) \cdot \nabla \phi(x)$ are treated in the same way. We refer readers to [17] for implementation details. Here is our implementation procedure:

1. If the original data is noisy, we first use tensor voting method to remove outliers in the data set.
2. Get distance function $d(x)$ and normalized saliency field $k(x)$.
3. Start with an initial guess of Γ and evolve it to steady state using (11).

The distance field is computed using the fast sweeping method which is of $O(N + M)$ complexity, where N is the number of grid point and M is the number of data point [13, 17]. Local level set method [24] is used to cut down the computation cost.

To further accelerate the computation, we can neglect the curvature term and just use the two convection terms initially to evolve the surface as suggested in [18]. This allows us to remove more strict CFL condition due to the curvature term. When the evolution is near steady state, we can put in the curvature term to make the final surface smoother.

3.2 Some Improvements in Implementing the Tensor Voting Method

Generating the Initial Stick Tensor. In Sect. 2.2, we give out the original method of generating the initial stick tensor. However using stick tensor to simulate ball tensor or plate tensor is time-consuming and is not very accurate.

We consider the communication of two data points $P,\ Q$ in Fig. 3. Without any prior information, the most likely relationship of these two points is that they are on the straight line connecting them. So they give each other a plate tensor. Let $\hat{e}_{PQ} = (t_x,\ t_y,\ t_z)$ be the unit vector pointing from P to Q. Taking into account the decay with the increasing distance, we define the plate tensor as:

$$e^{\frac{l^2}{\sigma^2}}(I - \hat{e}_{PQ}\hat{e}_{PQ}^T) = e^{\frac{l^2}{\sigma^2}} \cdot \begin{bmatrix} 1 - t_x^2 & -t_x t_y & -t_x t_z \\ -t_x t_y & 1 - t_y^2 & -t_y t_z \\ -t_x t_z & -t_y t_z & 1 - t_z^2 \end{bmatrix}, \tag{12}$$

where l is the distance between $P,\ Q$. When l is bigger than a threshold we can ignore their communication. For every data point, summing up the contribution from all neighbors gives it the initial tensor.

Suppose the number of data points is N, and every point has M neighbors. Then if we use k stick tensor to represent a ball tensor, the original method needs $O(kNM)$ operations and the value of k can be large to represent all directions well. While our new algorithm needs $O(NM)$ operations.

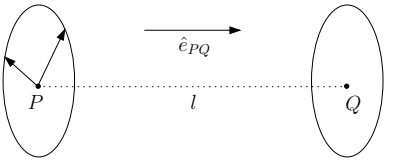


Fig. 3. Communication between two data points

Voting Process. In the second step of tensor voting, every data point propagates its tensor information to its neighboring grid points like in Fig. 1. But every time a data point votes on a grid point, we need to compute the weight $DF(s, \kappa, \sigma)$. It is a time-consuming process. In fact, to reduce the computation cost, we can first establish an index table of the weight function, then when we need to compute the contribution from a point to another point we can refer to the table which significantly speed up the computation. Moreover, the higher resolution the index table has, the better we approximate the true weight function. In our implementation, we use an index table whose resolution is three times of that of our computation grids. As showed in Fig. 4, suppose we know the coordinates of point P, Q, and know the unit stick vector $\hat{e}_P$ at P, we look up the index table to get the value of $DF(s, \kappa, \sigma)$. We demonstrate this in 2D. First, we can determine the unit stick vector $\hat{e}_Q$ at Q voted from P by the formula:

$$\hat{e}_Q = \hat{e}_P - 2\frac{\overrightarrow{QP}}{\| \overrightarrow{QP} \|}\left(\frac{\overrightarrow{QP} \cdot \hat{e}_P}{\| \overrightarrow{QP} \|}\right) \quad . \tag{13}$$

Then we find the corresponding weight from the index table by l_x and l_y. Times this weight with the stick saliency value at P gives the stick saliency value at Q voted from P. Here,

$$l_x = \| \overrightarrow{QP} \| \ \sqrt{1-c^2}, \quad l_y = \| \overrightarrow{QP} \| \ |c|, \quad c = \frac{\overrightarrow{QP} \cdot \hat{e}_P}{\| \overrightarrow{QP} \|} \quad . \tag{14}$$

The establishment of the index table is relatively simple, we only need to store the contributions from a unit stick tensor to its neighboring points.

Tune the Range of θ. In the original tensor voting method, a point vote on another point only if $\theta \leq \pi/4$ (see Fig. 1). However, for our surface reconstruction purpose it seems that two nearby points on a smooth surface (relative to the grid size) are not likely to form a large angle. In our experiments for real data, we get better results if we restrict θ in a smaller interval. In our experiments, we set $\theta \leq \pi/12$. In the following, we use $\theta = \psi$ to mean the range of θ is $[0, \ \psi]$.

4 Experimental Results

In this section we present experimental results for our method. All results are displayed using OpenDX.

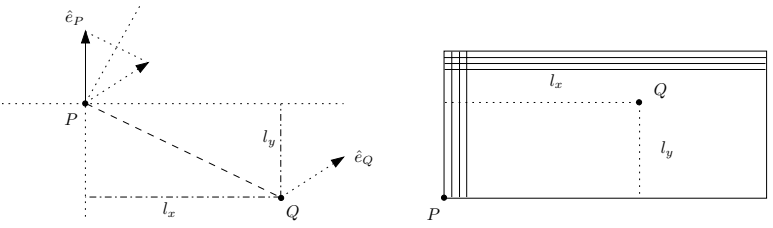
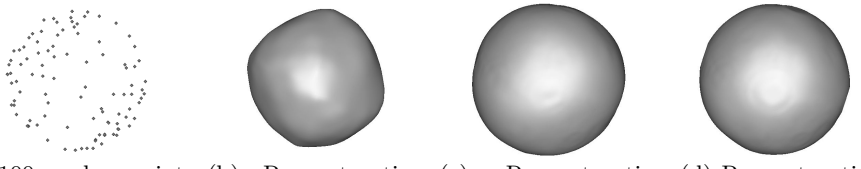


Fig. 4. Use the index table instead of computing $DF(s, \kappa, \sigma)$



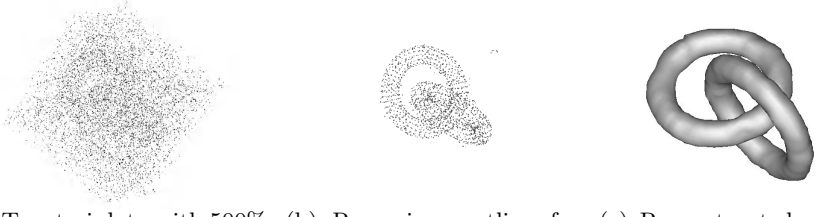
(a) 100 random points on a ball

(b) Reconstruction using minimal surface model. $\alpha = 1,\ \beta = 0,\ \gamma = 0.5$

(c) Reconstruction only using stick saliency field. $\alpha = 0,\ \beta = 10,\ \gamma = 0.5$

(d) Reconstruction using the new model. $\alpha = 1,\ \beta = 10,\ \gamma = 0.5$

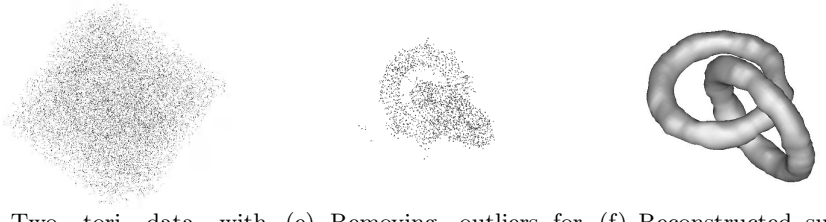
Fig. 5. Surface reconstruction from non-uniform data



(a) Two tori data with 500% noise

(b) Removing outliers for (a). The threshold is 0.5

(c) Reconstructed surface from (b). $\alpha = 1, \beta = 5, \gamma = 0.05$

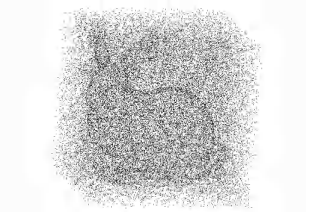


(d) Two tori data with 1000% noise

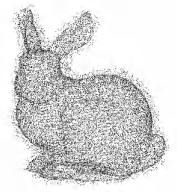
(e) Removing outliers for (d). The threshold is 0.5

(f) Reconstructed surface from (e). $\alpha = 1, \beta = 11, \gamma = 0.05$

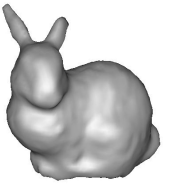
Fig. 6. Surface reconstruction from noisy two tori data. $\sigma = 3,\ \theta = \pi/18$.



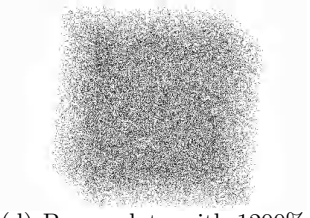
(a) Bunny data with 500% noise



(b) Removing outliers for (a). The threshold is 0.5.



(c) Reconstructed result from (b). $\alpha = 1$, $\beta = 2$, $\gamma = 0$.



(d) Bunny data with 1200% noise

(e) Removing outliers for (d). The threshold is 0.6.

(f) Reconstructed result from (e). $\alpha = 1$, $\beta = 2$, $\gamma = 0$.

Fig. 7. Surface reconstruction from noisy bunny data. $\sigma = 3$, $\theta = \pi/18$.

Table 1. Experimental data

	2 tori		bunny	
Reconstructed result	Fig. 6(c)	Fig. 6(f)	Fig. 7(c)	Fig. 7(f)
Noisy ratio	500%	1000%	500%	1200%
Total points	7200	13200	215682	467311
Points after removing noise	1601	2413	62807	119232
Time for tensor voting at twice	8.11s	14.313s	1606.015s	6514.141s
Evolution time	87.516s	119.5s	57.875s	132.875s

The computations are carried out on a CPU of AMD Athlon XP 3200+ and 1GB memory. All the reconstructions are on a $61 \times 61 \times 61$ grid. We use both synthetic and real data to show the ability of handling highly noisy and non-uniform data.

In Fig. 5 we show the case of non-uniform data. The data set is 100 random points on a sphere. We can see from the result that the reconstructed surface (d) using our new model is better than (b) reconstructed from minimal surface model or (c) which only uses stick saliency field for reconstruction.

In Fig. 6, Fig. 7, we demonstrate the ability of dealing with highly noisy data for the new model. We add points randomly in the box that contain the original data. The noise ratio is the ratio between the number of added points and the number of original data points. The original data sets for the two tori and the bunny have 1200 and 35947 points respectively. Some experimental data

are displayed in Table 1 from which we can see that when the number of data points is large, the most expensive step is tensor voting.

References

[1] D. F. Rogers, *An introduction to NURBS: with historical perspective*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[2] N. Amenta and M. Bern, "Surface reconstruction by voronoi filtering," *Discrete and Comput. Geometry*, vol. 22, pp. 481–504, 1999.

[3] H. Edelsbrunner, "Shape reconstruction with delaunay complex," in *LATIN '98: Proceedings of the Third Latin American Symposium on Theoretical Informatics*, (London, UK), pp. 119–132, Springer-Verlag, 1998.

[4] J. D. Boissonnat, "Geometric structures for three dimensional shape reconstruction," *ACM Trans. Graphics 3*, pp. 266–286, 1984.

[5] N. Amenta, M. Bern, and D. Eppstein, "The crust and the -skeleton: Combinational curve reconstruction," *14th ACM Symposium on Computational Geometry*, 1998.

[6] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 415–421, ACM Press, 1998.

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 71–78, ACM Press, 1992.

[8] C. L. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3d scans," in *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 109–118, ACM Press, 1995.

[9] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 303–312, ACM Press, 1996.

[10] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt, "Implicit surface-based geometric fusion," *Comput. Vis. Image Underst.*, vol. 69, no. 3, pp. 273–291, 1998.

[11] J. Bloomenthal and B. Wyvill, eds., *Introduction to Implicit Surfaces*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[12] R. Whitaker, "A level set approach to 3D reconstruction from range data," *International journal of Computer Vision*, 1997.

[13] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang, "Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method," *Computer Vision and Image Understanding*, vol. 80, pp. 295–319, 2000.

[14] H. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," in *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, (Washington, DC, USA), p. 194, IEEE Computer Society, 2001.

[15] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 67–76, ACM Press, 2001.

[16] G. Medioni, M.-S. Lee, and C.-K. Tang, *A computational framework for segmentation and grouping*. Elsevier, 2000.
[17] H. Zhao, "Fast sweeping method for Eikonal equations," *Mathematics of Computation*, 2004.
[18] H. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction and deformation using the level set method," *Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver*, July, 2001.
[19] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Springer, New York, 2002.
[20] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, pp. 179–195, 1996.
[21] J. Jia and C.-K. Tang, "Inference of segmented color and texture description by tensor voting," vol. 26, pp. 771–786, June 2004.
[22] W.-S. Tong, C.-K. Tang, P. Mordohai, and G. Medioni, "First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 594–611, 2004.
[23] C.-K. Tang and G. Medioni, "Inference of integrated surface, curve, and junction descriptions from sparse 3d data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1206–1223, 1998.
[24] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, "A PDE based fast local level set method," *J. Comput. Phys.*, vol. 155, pp. 410–438, 1999.

A C^1 Globally Interpolatory Spline of Arbitrary Topology

Ying He, Miao Jin, Xianfeng Gu, and Hong Qin

Center for Visual Computing (CVC) and Department of Computer Science,
Stony Brook University, Stony Brook, NY, 11794-4400, USA
{yhe, mjin, gu, qin}@cs.sunysb.edu

Abstract. Converting point samples and/or triangular meshes to a more compact spline representation for arbitrarily topology is both desirable and necessary for computer vision and computer graphics. This paper presents a C^1 manifold interpolatory spline that can exactly pass through all the vertices and interpolate their normals for data input of complicated topological type. Starting from the Powell-Sabin spline as a building block, we integrate the concepts of global parametrization, affine atlas, and splines defined over local, open domains to arrive at an elegant, easy-to-use spline solution for complicated datasets. The proposed global spline scheme enables the rapid surface reconstruction and facilitates the shape editing and analysis functionality.

1 Introduction

Constructing smooth interpolatory spline surfaces from any data input in 3D is frequently needed in visual computing. Given a scattered point cloud, $\{\mathbf{P}_i = (x_i, y_i, z_i)\}_{i=1}^m$, and associated normal vectors $\{\mathbf{n}_i = (nx_i, ny_i, nz_i)\}_{i=1}^m$, the goal of this paper is to find a smooth surface $\mathbf{F}$ that interpolates both the vertex positions and their normals simultaneously of complicated topological type.

Unlike most of the conventional methods which typically trim parametric spline surfaces defined over open planar domains, stitch them along their trimmed boundaries with care, and enforce the smoothness requirements of certain degree across their common boundaries, our spline scheme is global and interpolatory. It can faithfully reconstruct smooth shapes of any manifold from geometric input without resorting to any patching and/or trimming operations. The technical core of our new approach is the Powell-Sabin spline defined over any open, triangulated domain. The primary goal is the exact interpolation (for both vertices and their normals), therefore, the Powell-Sabin spline scheme is an ideal candidate for this requirement. Nonetheless, the technical challenge is how to generalize the Powell-Sabin spline defined over planar, triangulated domains to a global spline spanning over domain of complicated topology without any cutting and patching work. We accomplish this mission through the following steps: (1) The initial, raw data input is globally parameterized in order to map the 3D geometry onto a 2D domain; (2) For any 3D point, we are only interested in a certain localized 2D region in its vicinity; (3) We decompose the entire 3D geometry into a suite of overlapping regions and construct their corresponding affine atlases on 2D; (4) These affine

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 295–306, 2005.

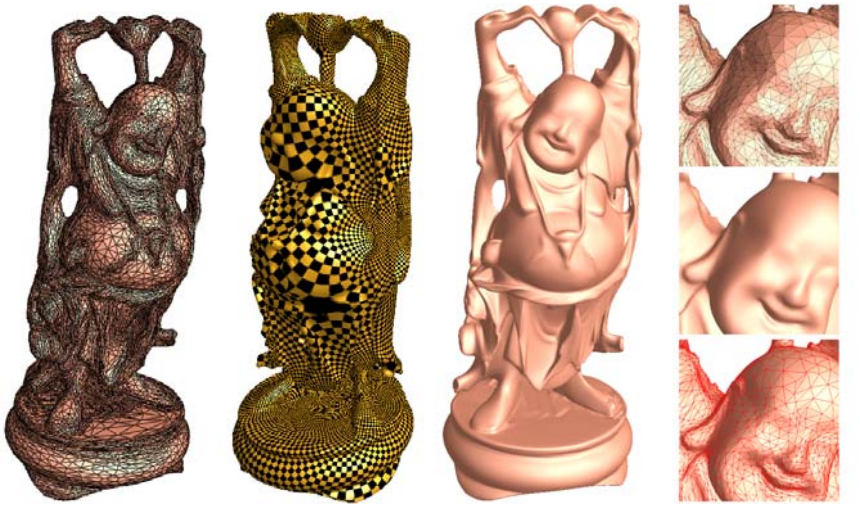


Fig. 1. Globally interpolatatory spline: (a) A genus-6 Buddha model with 25K vertices; (b) Global conformal parameterization; (c) A global C^1 spline surface which interpolates all the vertices and their normals of (a); (d) Close-up view: *top*, original mesh; *middle*, spline surface; *bottom*, spline surface with the red curves corresponding to the edges in the mesh

charts in 2D constitute all the local parametric domain for defining all the open Powell-Sabin spline surfaces that interpolate only a subset of data points; (5) These locally defined spline surfaces span across their neighbors and share some common regions; and finally (6) We build a globally interpolatory spline by collecting all the control points and using all the affine atlases as their global domain.

2 Previous Work

2.1 Planar Powell-Sabin Spline

Powell-Sabin splines are functions in the space $S_2^1(\Delta_{ps})$ of C^1 continuous piecewise quadratic functions on a Powell-Sabin refinement [1]. Such a refinement Δ_{ps} can be obtained from an arbitrary triangulation Δ by splitting each triangle into six subtriangles with a common interior point. In contrast to triangular Bézier splines, where imposing smoothness conditions between the patches requires a great number of nontrivial relations between the control points to be satisfied, the C^1 continuity of a Powell-Sabin spline is guaranteed for any choice of the control points.

The first B-spline representation of Powell-Sabin spline was derived by Shi et al. [2]. However, their construction approach had serious drawbacks from the numerical point of view. Dierckx [3] resolved the numerical problem by constructing a normalized *B*-spline basis for Powell-Sabin splines. This representation has a very nice geometric interpretation involving the tangent control triangles for manipulating the Powell-Sabin surfaces. Since then, the normalized Powell-Sabin spline has been receiving much attention in the computer aided geometric design community. Surface approximation and

interpolation using Powell-Sabin spline have been reported in [4,5,6]. Windmolders and Dierckx solved the subdivision problem for uniform Powell-Sabin splines, that is on triangulations with all equilateral triangles [7]. Recently, Vanraes et al. present the subdivision rule for general Powell-Sabin spline [8].

2.2 Interpolatory Spline

Interpolation is a very useful and intuitive feature in computer aided geometric design. Two different research directions have been pursued. One is based on the subdivision surfaces that recursively subdivide the control mesh, such as the butterfly scheme [9] or modified butterfly scheme [10]. The other direction consists of building a patch of smoothly joined parametric patches. This paper focuses on the spline based interpolation scheme. There exists a vast literature on interpolation by splines over triangulations (see the survey [11] and the references therein). In the interest of the space, we only cite few of them which are closely related to our work.

Hahmann and Bonneau [12] presented a piecewise quintic G^1 spline surface interpolating the vertices of a triangular surface mesh of arbitrary topological type. They further improved the method without imposing any constraint on the first derivatives and thus avoid any unwanted undulations when interpolating irregular triangulations [13]. Nürnberger and Zeilfelder presented [14] a local Lagrange interpolation scheme for C^1-splines of degree $q \geq 3$ on arbitrary triangulations. This interpolating spline yields optimal approximation order and can be computed with linear complexity.

2.3 Manifold Construction

There are some related work on defining functions over manifold. Grimm and Hugues [15] pioneered a generic method to extend B-splines to surfaces of arbitrary topology, based on the concept of overlapping charts. Cotrina et al. proposed a C^k construction on manifold [16,17]. Ying and Zorin [18] presented a manifold-based smooth surface construction method which has C^∞-continuous with explicit nonsingular parameterizations. Recently, Gu, He and Qin [19] developed a general theoretical framework of manifold splines in which spline surfaces defined over planar domains can be systematically generalized to any manifold domain of arbitrary topology (with or without boundaries). Manifold spline is completely different from the above methods in that: 1) The transition functions of manifold spline must be affine. Therefore, the requirements of manifold spline is much stronger. That is why topological obstruction plays an important role in the construction. 2) Manifold spline produces the polynomial or rational polynomials. On any chart, the basis functions are always polynomials or rational polynomials, and represented as B-splines or rational B-splines.

In [19], Gu et al. defined the manifold spline based on triangular B-spline [20]. This construction requires a complicated data fitting procedure when converting points to splines. Inspired by [19], we strive to devise a globally interpolatory splines that are founded upon the original work of [3]. Our method is different from the above methods in that: 1) All the existing developments of Powell-Sabin splines are defined on the planar domain; 2) The existing global interpolatory splines need patching and stitching work; 3) All the manifold constructions except the manifold splines do not produce

globally polynomials or rational polynomials. Our work generalizes the planar Powell-Sabin spline to arbitrary manifold without any patching and stitching work. Also, due to the nice properties of the normalized Powell-Sabin spline, our method can interpolate both positions and normals.

3 The Globally Interpolatory Spline

This section first reviews the normalized planar Powell-Sabin B-spline [3] and then presents all the necessary components for our global spline scheme.

3.1 Powell-Sabin Spline on the Planar Domain

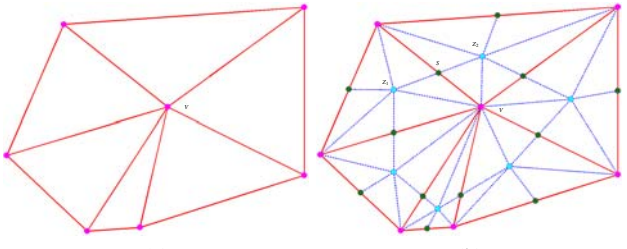
Let Ω be a polygonal domain in $\mathbb{R}^2$ and let Δ be a conforming triangulation of Ω, comprising triangles ρ_j, $j = 1,\ldots,N_t$, having vertices $V_i := (x_i, y_i), i = 1,\ldots,N_v$. A Powell-Sabin refinement, Δ_{ps} of Δ is the refined triangulation, obtained by subdividing each triangle of Δ into six sub-triangles as follows. Select an interior point Z_j in each triangle ρ_j and connect it with the three vertices of ρ_j and with the points Z_{j_1},Z_{j_2},Z_{j_3} where ρ_{j_1},ρ_{j_2},ρ_{j_3} are the triangles adjacent to ρ_j (See Figure 2). We denote by $S_2^1(\Delta_{ps})$ the space of piecewise C^1 continuous quadratic polynomials on Δ_{ps}. Powell and Sabin [1] proved that the dimension of the space $S_2^1(\Delta_{ps})$ equals to $3N_v$ and any element of $S_2^1(\Delta_{ps})$ is uniquely determined by its value and its gradient at the vertices of Δ, i.e., there exists a unique solution $s(x,y) \in S_2^1(\Delta_{ps})$ for the interpolation problem

$$s(V_i) = f_i,\ \frac{\partial}{\partial x}s(V_i) = f_{x,i},\ \frac{\partial}{\partial y}s(V_i) = f_{y,i},\ i = 1,\ldots,N_v. \tag{1}$$

So given the function and its derivative values at each vertex V_i, the Bézier ordinates on the domain sub-triangles are uniquely defined and the continuity conditions between sub-triangles are automatically enforced.

Dierckx [3] showed that each piecewise polynomial $s(x,y) \in S_2^1(\Delta_{ps})$ has a unique representation

$$s(x,y) = \sum_{i=1}^{N_v}\sum_{j=1}^{3} c_{ij}B_i^j(x,y),\ (x,y) \in \Omega \tag{2}$$



(a) (b)

Fig. 2. The Powell-Sabin refinement Δ^* (b) of a triangulation Δ (a)

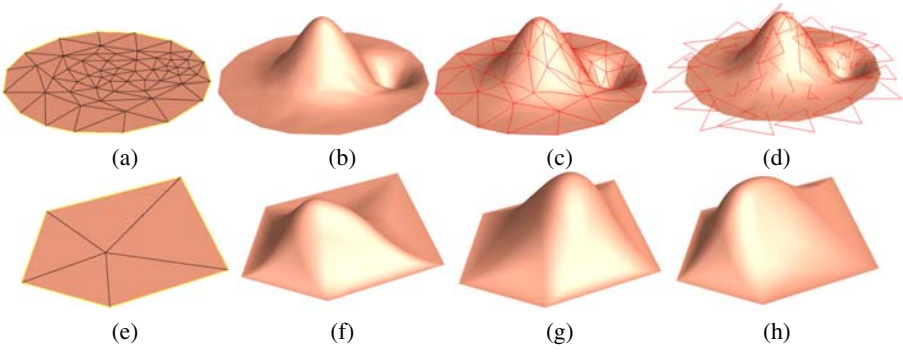


Fig. 3. Powell-Sabin spline over a planar domain: (a) Domain etriangulation; (b) Spline surface; (c) Spline surface, the red curves correspond to the edges in the domain triangulation; (d) Spline surface overlaid by the control triangles (shown in red) which are tangent to the surface; (e) The molecule of one vertex **v**; (f)-(h) Three basis functions associated with vertex **v**

where the basis functions form a partition of unity, i.e.,

$$B_i^j(x,y) \geq 0 \tag{3}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{3} B_i^j(x,y) = 1 \text{ for all } x,y \in \Omega \tag{4}$$

Furthermore, these basis functions have local support: $B_i^j(x,y)$ vanishes outside the so-called molecule M_i of vertex V_i, which is the union of all triangles $\mathcal{T}_k$ containing V_i.

The basis functions $B_i^j(x,y)$ can be obtained by finding three linearly independent triplets $(\alpha_{ij},\beta_{ij},\gamma_{ij})$, $j=1,2,3$ for each vertex V_i. $B_i^j(x,y)$ is the unique solution of the interpolation problem with $(f_k, f_{xk}, f_{yk}) = (\delta_{ki}\alpha_{ij}, \delta_{ki}\beta_{ij}, \delta_{ki}\gamma_{ij})$, where δ_{ki} is the Kronecker delta. The triplets $(\alpha_{ij},\beta_{ij},\gamma_{ij})$, $j=1,2,3$ are determined by the following Dierckx's algorithm [3,21]:

1. For each vertex v_i, find its Powell-Sabin triangle points, which are the immediately surrounding Bézier domain points of the vertex v_i and vertex v_i itself.
2. For each vertex v_i, find a triangle $t_i(Q_{i1},Q_{i2},Q_{i3})$ which contains all the Powell-Sabin triangle points of v_i from all the triangles in the molecule M_i. Denote $Q_{ij} = (X_{ij},Y_{ij})$ the position of vertex Q_{ij}.
3. Three linearly independent triplets of real numbers $\alpha_{ij},\beta_{ij},\gamma_{ij}$, $j=1,2,3$ can be derived from the Powell-Sabin triangle t_i of a vertex v_i as follows:
   $(\alpha_{i1},\alpha_{i2},\alpha_{i3}) =$ Barycentric coordinate of v_i with respect to t_i,
   $(\beta_{i1},\beta_{i2},\beta_{i3}) = ((Y_{i2}-Y_{i3})/h, (Y_{i3}-Y_{i1})/h, (Y_{i1}-Y_{i2})/h)$,
   $(\gamma_{i1},\gamma_{i2},\gamma_{i3}) = ((X_{i3}-X_{i2})/h, (X_{i1}-X_{i3})/h, (X_{i2}-X_{i1})/h)$,
   where $h = \det \begin{pmatrix} 1 & 1 & 1 \\ X_{i1} & X_{i2} & X_{i3} \\ Y_{i1} & Y_{i2} & Y_{i3} \end{pmatrix}$.

We then define the control triangles as $T_i(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \mathbf{C}_{i3})$. Dierckx proved that the normalized Powell-Sabin spline has a very nice geometric interpretation that the control triangle is tangent to the spline surface [3].

Figure 3 illustrates an example of Powell-Sabin spline surface over a planar triangulated domain. Note that, their basis functions $B_i^j(\mathbf{u})$ vanish outside the molecule M_i (see Figure 3(e-h)). Furthermore, the control points $(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \mathbf{C}_{i3})$ form a control triangle which is always tangent to the spline surface at $\mathbf{s}(\mathbf{v}_i)$ (see Figure 3(d)).

3.2 Generalizing Powell-Sabin Spline to Arbitrary Topology

In [19], Gu et al. addressed several key technical issues of manifold splines in which spline surfaces defined over planar domains can be systematically extended to manifold domains of arbitrary topology. In a nutshell, a manifold spline can be intuitively interpreted as a set of spline patches that are automatically glued in a coherent and consistent way without any gap, such that all the patches collectively cover the entire manifold. The surface evaluation can be easily conducted using the control points and corresponding basis functions of any overlapping patches, without leading to any inconsistency. The followings are the necessary theoretical results which enable our global spline scheme based on Powell-Sabin's approach.

Theorem 1. *The sufficient and necessary condition for a manifold M to admit manifold spline is that M must be an affine manifold.*

This theorem implies that the existence of manifold splines solely depends on the existence of affine atlas. If the domain manifold M is an affine manifold, we will be able to directly generalize the local spline patches to a global spline defined on M. Details about the affine manifold and affine atlas can be found in the Appendix.

Theorem 2. *The only closed surface admitting affine atlas is of genus one. All oriented open 2-manifolds admit an affine atlas.*

Theorem 2 points out that not all surfaces admit the affine atlas. The topological obstruction of a global affine atlas is the Euler class. In fact, by removing one point from the closed domain manifold, we can convert it to an affine manifold.

Theorem 3 (Affine atlas deduced from conformal structure). *Given a closed genus g surface M, and a holomorphic 1-form ω. Denote by $Z = \{\text{zeros of } \omega\}$ the zero points of ω. Then the size of Z is no more than $2g-2$, and there exists an affine atlas on M/Z deduced by ω.*

Essentially, Theorem 3 indicates that an affine atlas of a manifold M can be deduced from its conformal structure in a straightforward fashion.

3.3 Algorithmic Details

Given a triangle mesh M of arbitrary topological type, we want to find a manifold Powell-Sabin spline which interpolates the vertices of M and their normals. Our spline surface construction algorithm consists of two consecutive steps: (1) compute the global conformal parameterization; and (2) construct the global spline.

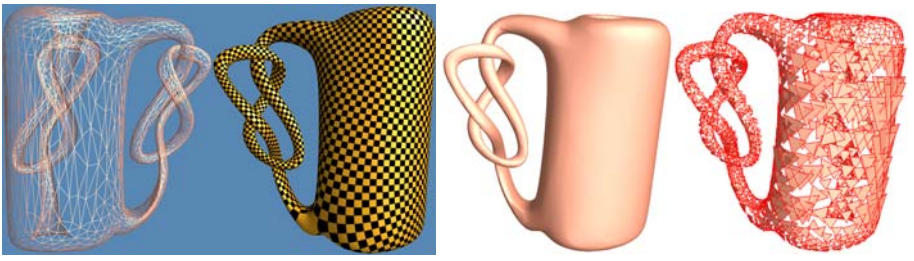


Fig. 4. Interpolation of a genus-2 model. (a) The two-hole bottle model with 2K vertices; (b) Global conformal parameterization; (c) Spline surface; (d) Control triangles.

Compute Global Conformal Parameterization and Affine Atlas. As mentioned above, the domain manifold M admits a manifold spline if and only if it has affine atlas, which can be deduced from the conformal structure of M directly. Thus, in order to construct our global spline, we shall first compute the conformal structures of the domain manifold M. A conformal atlas is an atlas such that all transition functions are analytic. Two conformal atlases are compatible if their union is still a conformal atlas. All compatible conformal atlases form conformal structure. It is known that all surfaces have conformal structure and are called Riemann surfaces. The algorithm to compute global conformal parameterization and affine atlas is as follows:

1. Compute the holomorphic 1-form ω of M using Gu-Yau's algorithm [22].
2. Remove the zero points Z of ω and the adjacent faces.
3. Construct an open covering for M/Z. For each vertex $\mathbf{V}_i$, take the union of all faces within its molecule as an open set, denoted by U_i.
4. Test if the union of any two U_i's is a topological disk by checking the Euler number. If not, subdivide U_i.

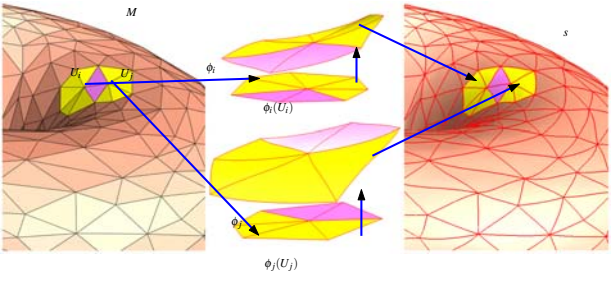


Fig. 5. Constructing local spline patches: The parametric domain M is a triangular mesh of arbitrary topology as shown on the left. The polynomial spline surface $\mathbf{s}$ is shown on the right. Two overlapping spline patches are magnified and highlighted in the middle. On each parameter chart (U_i, ϕ_i), (U_j, ϕ_j), the surface is a locally defined planar Powell-Sabin spline patch. For the overlapping part, its two planar domains differ only by an affine transformation ϕ_{ij}.

5. Pick one vertex $p_i \in U_i$, for any vertex $p \in U_i$, define $\phi_i(p) = \int_{p_i}^{p} \omega$.
6. Compute coordinate transition functions $\phi_{ij} = \int_{p_i}^{p_j} \omega$.

Global Spline Construction. Note that the evaluation of Powell-Sabin spline over any planar region relies on the computation of Barycentric coordinates of the parameter with respect to the domain triangles. If we change the parameter by an affine transformation, the evaluation is invariant and the final shape of the spline surface will not be changed. Figure 5 highlights the transition from local patches to the global spline. The algorithm to construct the global spline is as follows:

1. Prepare the underlying parameteric domain (For any vertex $\mathbf{V}_i \in M$, denote by (U_i, ϕ_i) its parametric chart which contains the molecule of $\mathbf{V}_i$).
2. Compute the three linear independent triplets, $(\alpha_{ij}, \beta_{ij}, \gamma_{ij})$, $j = 1, 2, 3$. Build the basis functions using the above Dierckx's algorithm.
3. Assign the control points $(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \mathbf{C}_{i3})$ which satisfy

$$\mathbf{V}_i = \sum_{j=1}^{3} \alpha_{ij} \mathbf{C}_{ij} \tag{5}$$

and

$$\frac{(\mathbf{C}_{i1} - \mathbf{C}_{i2}) \times (\mathbf{C}_{i1} - \mathbf{C}_{i3})}{\| (\mathbf{C}_{i1} - \mathbf{C}_{i2}) \times (\mathbf{C}_{i1} - \mathbf{C}_{i3}) \|} = \mathbf{n}_i = (nx_i, ny_i, nz_i)^T \tag{6}$$

One can prove that the control triangle $(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \mathbf{C}_{i3})$ is tangent to the spline surface $\mathbf{s}$ at $\mathbf{V}_i$, i.e.,

$$\mathbf{s}(\phi_i(\mathbf{V}_i)) = \mathbf{V}_i \tag{7}$$

$$\frac{\mathbf{s}_u(\phi_i(\mathbf{V}_i)) \times \mathbf{s}_v(\phi_i(\mathbf{V}_i))}{\| \mathbf{s}_u(\phi_i(\mathbf{V}_i)) \times \mathbf{s}_v(\phi_i(\mathbf{V}_i)) \|} = \mathbf{n}_i \tag{8}$$

The detailed proof is in the Appendix.

Variational Shape Design. In the Powell-Sabin spline scheme, each vertex of the domain triangulation is associated with three control points. In the above spline construction step, we require the control points satisfying Equation (5) and (6). Therefore, there are still three degrees of freedom remaining. We can use these free variables for variational shape design. For example, we can fair the spline surface by minimizing the following energy functional subject to the interpolation constraints:

$$\begin{aligned} \min \ & \alpha \iint_M (\mathbf{s}_u^2 + \mathbf{s}_v^2) du dv + \beta \iint_M (\mathbf{s}_{uu}^2 + 2\mathbf{s}_{uv}^2 + \mathbf{s}_{vv}^2) du dv \\ \text{subject to } & \mathbf{V}_i = \sum_{j=1}^{3} \alpha_{ij} \mathbf{C}_{ij} \\ & \langle \mathbf{C}_{i1} - \mathbf{C}_{i2}, \mathbf{n}_i \rangle = 0 \\ & \langle \mathbf{C}_{i2} - \mathbf{C}_{i3}, \mathbf{n}_i \rangle = 0, \ \text{ for each vertex } \mathbf{V}_i \in M, \end{aligned} \tag{9}$$

where $\langle , \rangle$ is the inner product, u and v are parameters on the local charts. The objective function is the standard thin-plate energy with membrane terms, which can be written

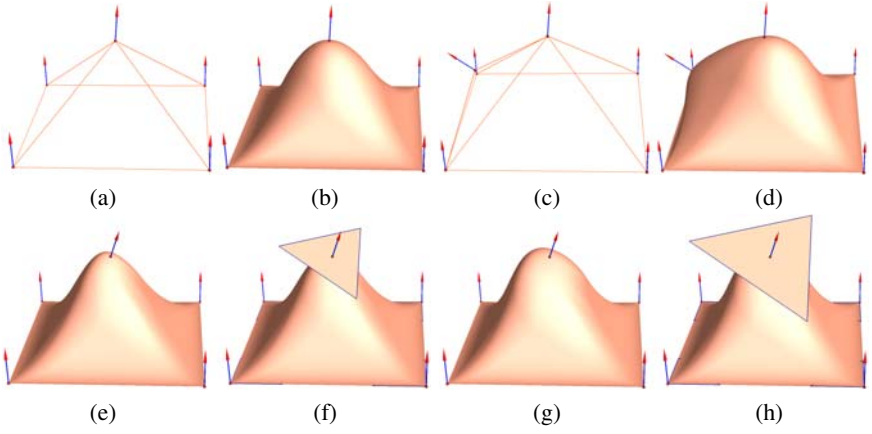


Fig. 6. Manipulation of manifold Powell-Sabin spline: The input is a triangular mesh M with normal information as shown in (a). We construct a manifold Powell-Sabin spline $\mathbf{S}$ to interpolate both the positions and normals of M (shown in (b)). We insert a new vertex v in the original mesh and assign a normal to v (shown in (c)). The corresponding spline is shown in (d). We can also change the normal but fix the positions, the spline and control triangles are shown in (e) and (f), respectively. We can even fix the vertices positions and their normals but change the size of the control triangles without violating the interpolation property. In (h), we enlarge the control triangle of the top-most vertex and get a new surface shown in (g). Note that the new spline still interpolates the positions and normals.

as a quadratic form of control points. Therefore, the above optimization problem can be solved efficiently using the Lagrange multiplier method.

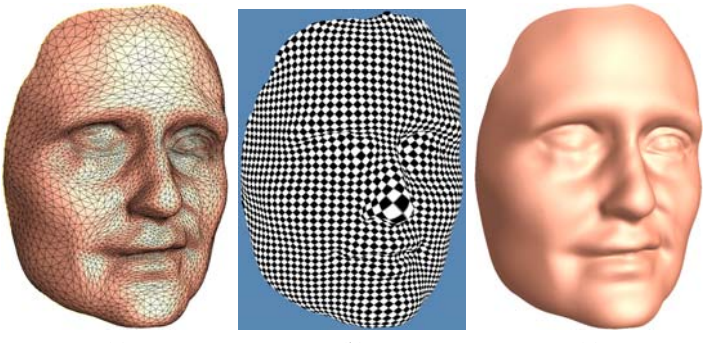
Handling the Singular Points. In [19], Gu et al. showed the manifold splines must have singular points if the domain manifold is closed and not a torus. The number of singular points is no more than $2g-2$ for a genus g domain manifold M. The singular points Z can be automatically detected from the conformal structure of M by checking the winding number. Then the molecule of Z is removed from M. No spline patches are defined on the molecule of Z. Therefore, there exist holes in the spline surface. For each hole, we compute a minimal surface spanning the hole such that it satisfies the given boundary condition.

3.4 Properties

The proposed globally interpolatory spline (based on Powell-Sabin spline over the planar domain) exhibits the following features:

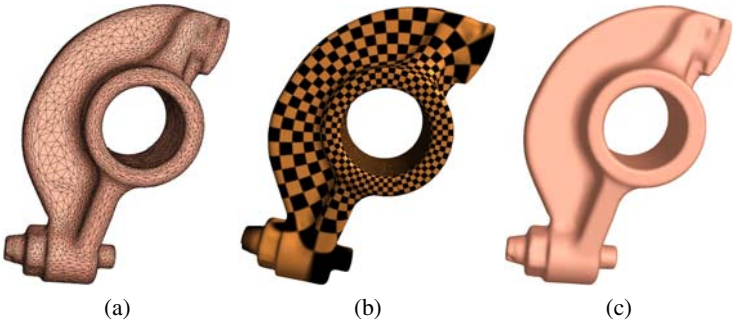
1. *Piecewise polynomial.* The global spline surface is a quadratic piecewise polynomial defined on the manifold M which has arbitrary triangulation. It is globally C^1-continuous and very efficient to evaluate.
2. *Local support.* It has local support since the basis functions $B_i^j(\mathbf{u})$ vanish outside the molecule of $\mathbf{v}_i$.

3. *Tangent plane control/Interpolation/Local shape modification.* The control triangle $(\mathbf{C}_{i0}, \mathbf{C}_{i1}, \mathbf{C}_{i2})$ is tangent to the spline surface $\mathbf{s}$ at $\mathbf{V}_i$. Thus, by manipulating the control triangle, the spline surface can interpolate both positions and normals. Furthermore, besides interpolation of the positions and normals, the control triangle still has three degrees of freedom which can be used for local shape modification and variational shape design.
4. *Convex hull.* The polynomial surface is inside the convex hull of the control points.
5. *Local adaptive refinement.* Since there is no restriction on the triangulation of M, the spline surface can be locally refined by knot insertion, e.g., inserting a new vertex inside the existing triangle, or splitting any edge.
6. *Minimal number of singular points.* The number of singular points depends only on the topology of the manifold M, i.e., no more than $2g - 2$ singular points for a genus g domain manifold.



(a) (b) (c)

Fig. 7. Example of a genus-0 open surface: (a) The face model with 4K vertices; (b) Global conformal parameterization; (c) The globally interpolatory spline



(a) (b) (c)

Fig. 8. Example of a genus-1 surface: (a) The rockerarm model with 10K vertices; (b) Global conformal parameterization; (c) The globally interpolatory spline

4 Results

We have implemented a prototype system on a 3GHz Pentium IV PC with 1GB RAM. Figure 6 illustrates the various properties of manifold Powell-Sabin spline which is useful in computer aided geometric design. We perform experiments on several models of various topological types, i.e., a genus-0 face (Figure 7), a genus-1 rockerarm (Figure 8), a genus-2 bottle (Figure 4), and a genus-6 Happy Buddha (Figure 1). The overall computational procedure requires about $6 \sim 30$ minutes for our test models.

5 Conclusion

In this paper, our goal is to seek a global spline solution that will allow us to interpolate all vertices and their normals using one-piece spline representation without any cutting and stitching operations. Founded upon the Powell-Sabin spline, we have developed a new globally interpolatory spline which is truly one-piece formulation without generating any seams when crossing triangular edges on its domain mesh. The interpolation property is valuable for the reverse engineering task that can effectively convert point-cloud raw data to the compact spline formulation. Our globally interpolatory spline is also relevant to surface modeling, variational design, and interactive editing.

Acknowledgements

This research was partially supported by the NSF grant ACI-0328930, the ITR grant IIS-0326388, and the Alfred P. Sloan Fellowship.

References

1. Powell, M.J.D., Sabin, M.A.: Piecewise quadratic approximations on triangles. ACM Trans. Math. Softw. **3** (1977) 316–325
2. Shi, X., Wang, S., Wang, W., R.H., W.: The C^1 quadratic spline space on triangulations. Technical Report Report 86004, Department of Mathematics, Jilin University (1996)
3. Dierckx, P.: On calculating normalized powell-sabin b-splines. Computer Aided Geometric Design **15** (1997) 61–78
4. Dierckx, P., Van Leemput, S., Vermeire, T.: Algorithms for surface fitting using Powell-Sabin splines. IMA Journal of Numerical Analysis **12** (1992) 271–299
5. Willemans, K., Dierckx, P.: Smoothing scattered data with a monotone Powell-Sabin spline surface. Numerical Algorithms **12** (1996) 215–232
6. Manni, C., Sablonniere, P.: Quadratic spline quasi-interpolants on powell-sabin partitions. (2004) submitted.
7. Windmolders, J., Dierckx, P.: Subdivision of uniform Powell-Sabin splines. Computer Aided Geometric Design **16** (1999) 301–315
8. Vanraes, E., Windmolders, J., Bultheel, A., Dierckx, P.: Automatic construction of control triangles for subdivided Powell-Sabin splines. Computer Aided Geometric Design **21** (2004) 671–682
9. Dyn, N., Levine, D., Gregory, J.A.: A butterfly subdivision scheme for surface interpolation with tension control. ACM Trans. Graph. **9** (1990) 160–169

10. Zorin, D., Schröder, P., Sweldens, W.: Interpolating subdivision for meshes with arbitrary topology. In: Proceedings of SIGGRAPH '96. (1996) 189–192
11. Nürnberger, G., Zeilfelder, F.: Developments in bivariate spline interpolation. J. Comput. Appl. Math. **121** (2000) 125–152 Numerical analysis in the 20th century, Vol. I, Approximation theory.
12. Hahmann, S., Bonneau, G.P.: Triangular G^1 interpolation by 4-splitting domain triangles. Computer Aided Geometric Design **17** (2000) 731–757
13. Hahmann, S., Bonneau, G.P.: Polynomial surfaces interpolating arbitrary triangulations. IEEE Trans. Vis. Comput. Graph **9** (2003) 99–109
14. Nürnberger, G., Zeilfelder, F.: Lagrange interpolation by bivariate C^1-splines with optimal approximation order. Adv. Comput. Math. **21** (2004) 381–419
15. Grimm, C.M., Hughes, J.F.: Modeling surfaces of arbitrary topology using manifolds. In: Proceedings of SIGGRAPH '95. (1995) 359–368
16. Cotrina, J., Pla, N.: Modeling surfaces from meshes of arbitrary topology. Computer Aided Geometric Design **17** (2000) 643–671
17. Cotrina, J., Pla, N., Vigo, M.: A generic approach to free form surface generation. In: Proceedings of ACM symposium on Solid modeling and applications. (2002) 35–44
18. Ying, L., Zorin, D.: A simple manifold-based construction of surfaces of arbitrary smoothness. ACM Trans. Graph. **23** (2004) 271–275
19. Gu, X., He, Y., Qin, H.: Manifold splines. In: Proceedings of ACM Symposium on Solid and Physical Modeling. (2005) 27–38
20. Dahmen, W., Micchelli, C.A., Seidel, H.P.: Blossoming begets B-spline bases built better by B-patches. Mathematics of Computation **59** (1992) 97–115
21. Vanraes, E., Dierckx, P., Bultheel, A.: On the choice of the PS-triangles. Report TW 353, Department of Computer Science, K.U.Leuven (2003)
22. Gu, X., Yau, S.T.: Global conformal surface parameterization. In: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing. (2003) 127–137

Appendix: Proof of the Interpolation Property

We prove that our global spline (based on Powell-Sabin spline) interpolates the domain manifold M and its normals, i.e., for a vertex $\mathbf{V}_i \in M$, $\mathbf{s}(\phi_i(\mathbf{V}_i)) = \mathbf{V}_i$ and $\mathbf{n}(\phi_i(\mathbf{V}_i)) = \mathbf{n}_i$ where $\phi_i : U_i \to \mathbb{R}^2$ maps the molecule of $\mathbf{V}_i$ to the planar domain.

The basis functions of vertices $\mathbf{V}_k$ have local support, i.e., they vanish outside the molecule of $\phi_k(\mathbf{V}_k)$. Therefore,

$$\mathbf{s}(\phi_i(\mathbf{V}_i)) = \sum_{i=1}^{N_v} \sum_{j=1}^{3} \mathbf{C}_{ij} B_i^j(\phi_i(\mathbf{V}_i)) = \sum_{j=1}^{3} \mathbf{C}_{ij} B_i^j(\phi_i(\mathbf{V}_i)) = \sum_{j=1}^{3} \mathbf{C}_{ij} \alpha_{ij} = \mathbf{V}_i.$$

The last equation results from the fact that α_{ij}, $j = 1,2,3$ are also the Barycentric coordinate of $\mathbf{V}_i$ with respect to $(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \mathbf{C}_{i3})$. Similarly, the normal $\mathbf{n}(\phi(\mathbf{V}_i))$ can be calculated as

$$\begin{aligned}
&\mathbf{n}(\phi_i(\mathbf{V}_i)) \propto \mathbf{s}_u(\phi_i(\mathbf{V}_i)) \times \mathbf{s}_v(\phi_i(\mathbf{V}_i)) = (\sum_{j=1}^{3} \mathbf{C}_{ij} \beta_{ij}) \times (\sum_{j=1}^{3} \mathbf{C}_{ij} \gamma_{ij}) \\
&= \lambda(\mathbf{C}_{i1} \times \mathbf{C}_{i2} + \mathbf{C}_{i2} \times \mathbf{C}_{i3} + \mathbf{C}_{i3} \times \mathbf{C}_{i1}) \\
&= \lambda(\mathbf{C}_{i1} - \mathbf{C}_{i2}) \times (\mathbf{C}_{i1} - \mathbf{C}_{i3}) \propto \mathbf{n}_i,
\end{aligned}$$

where $\lambda = \beta_{i1}\gamma_{i2} - \beta_{i2}\gamma_{i1} = \beta_{i3}\gamma_{i1} - \beta_{i1}\gamma_{i3} = \beta_{i2}\gamma_{i3} - \beta_{i3}\gamma_{i2}$. Therefore, the control triangle $(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \mathbf{C}_{i3})$ is tangent to the surface $\mathbf{s}$ at vertex $\mathbf{V}_i$.

Solving PDEs on Manifolds with Global Conformal Parametrization

Lok Ming Lui, Yalin Wang, and Tony F. Chan

Mathematics Department, UCLA
{malmlui, ylwang, chan}@math.ucla.edu

Abstract. In this paper, we propose a method to solve PDEs on surfaces with arbitrary topologies by using the global conformal parametrization. The main idea of this method is to map the surface conformally to 2D rectangular areas and then transform the PDE on the 3D surface into a modified PDE on the 2D parameter domain. Consequently, we can solve the PDE on the parameter domain by using some well-known numerical schemes on $\mathbb{R}^2$. To do this, we have to define a new set of differential operators on the manifold such that they are coordinates invariant. Since the Jacobian of the conformal mapping is simply a multiplication of the conformal factor, the modified PDE on the parameter domain will be very simple and easy to solve. In our experiments, we demonstrated our idea by solving the Navier-Stoke's equation on the surface. We also applied our method to some image processing problems such as segmentation, image denoising and image inpainting on the surfaces.

1 Introduction

Image processing on the surface has become more and more important in medical imaging, computer graphics and computer vision. Many image processing techniques involve solving a partial differential equation (PDE) on the surface. In 2D image processing, variational approaches have been widely used. The minimization procedure can be reformulated as a partial differential equation, using the Euler-Lagrange equation. In order to extend the 2D image processing techniques to 3D, we therefore need to formulate a technique to solve PDEs on surfaces with arbitrary topologies.

In this paper, we propose to solve PDEs on surfaces by using the global conformal parametrization. The main idea is to map the surface conformally to the 2D rectangles with the minimum number of coordinates patches. The problem can then be solved by solving a modified PDE on the 2D parameter domain. To do this, we have to define a new set of differential operators on the manifold. Once a PDE on the 3D surface is reformulated to the corresponding PDE on the 2D domain, we can solve the PDE on 2D by using some well-known numerical schemes. Since the Jacobian of the conformal mapping is simply a multiplication of the conformal factor, the modified PDE on the parameter domain will be very simple and easy to solve.

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 307–319, 2005.

Recently, some level set based PDE solving approaches have been proposed ([5,6]). Compared with the level set based approaches, we explicitly describe the manifold by the conformal parametrization, instead of the implicit representation of the level set function. We use a new set of differential operators on the manifold, without doing a projection of the Euclidean differential operators. Our method considers maps which are defined only on the manifold so we do not need to extend maps to a narrow band of the surface.

2 Previous Work

Several research groups have reported works on solving PDEs on the surface. Turk [1] proposed to generate textures on arbitrary surfaces using reaction-diffusion, which require to solve PDE on the surface. Dorsey et al. [2] propose to solve PDEs on the surface for virtual weathering. Both of them solved the PDE directly on the triangulated surface, which involve the discretization of the equations in general polygonal grid. Stam [3] proposed to simulate fluid flow on the surface via solving the Navier-Stokes equation. He achieved this by combining the two dimensional stable fluid solver with an atlas of parametrizations of a Catmull-Clark surface. Clarenz et al. [4] has proposed an algorithm for solving finite element based PDEs on point surfaces. They constructed a number of local FE matrices that represent the surface properties over small point neighborhoods. These matrices are next assembled in a single matrix that allows PDE discretization and solving on complete surface. Sapiro et al. [5] [6] implemented a framework for solving PDEs on the surface via the level set method. They represented the surface implicitly by the zero-level set of an embedding function and extend the data on the surface to the 3D volume. This allowed them to perform all the computation on the fixed Cartesian grid.

3 Mathematical Theory

3.1 Computation of Conformal Parameterization

A diffeomorphism $f : M \rightarrow N$ is a *conformal mapping* if it preserves the first fundamental form up to a scaling factor (the conformal factor). Mathematically, this means that $ds_M^2 = \lambda f^*(ds_N^2)$, where ds_M^2 and ds_N^2 are the first fundamental form on M and N respectively and λ is the conformal factor. (See [7]) For a diffeomorphism between two genus zero surfaces, a map is conformal if and only if it minimizes the harmonic energy,$E_{harmonic}$. However, this is not true for surfaces with genus equal to one or higher.

For high genus surfaces, Gu et. al [8] has proposed an efficient approach to parameterize surfaces conformally to the 2D rectangles. This approach is based on the homology group theory, the cohomology group theory and the Hodge theory. We can summarize the algorithm with the following five steps. For details, please refer to [8].

Step 1: Given a high genus surface, find the homology basis $\{\xi_1, ..., \xi_{2g}\}$ of its homology group.
Step 2: Given the homology basis $\{\xi_1, ..., \xi_{2g}\}$, compute its dual basis $\{w_1, ..., w_{2g}\}$ which is called the cohomology basis.
Step 3: Diffuse the cohomology basis elements to harmonic 1-forms. This can be done by solving the following simultaneous equations:
(1) $dw = 0$ (*closedness*) (2) $\Delta w = 0$ (*harmonity*) (3) $\int_{\xi_i} w_j = \delta_{ij}$ (*duality*)
The existence of solution is guaranteed by Hodge theory.
Step 4: Compute the Hodge star conjugate $\{^*w_1, ..., ^* w_{2g}\}$ of $\{w_1, ..., w_{2g}\}$
Step 5: Integrate the holomorphic 1-form and get the conformal mapping: $f(x) = \int_\gamma w + i^*w$, where $w = \Sigma\lambda_i w_i$

The above five steps allow us to compute a conformal parametrization from the surface onto the 2D domain. (See Figure 1)

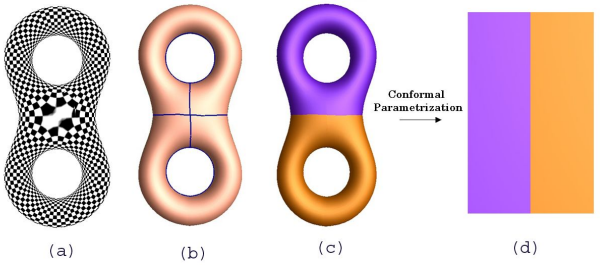


Fig. 1. Conformal parametrization of a high genus surface onto the 2D rectangles

3.2 Differential Operators on Manifolds

Many physical phenomenon can be explained via PDEs. In image processing, variational approaches are often used, which induces PDE solving. Therefore, it is important to define a set of partial differential operators on general manifolds. In this section, the partial differential operators on manifolds and the covariant differentiation on tensor fields will be discussed.

Let M be a manifold and $\phi : \mathbb{R}^2 \to M$ be the global conformal parametrization of M. With the conformal parametrization, we can do calculus on surfaces similar to what we do on $\mathbb{R}^2$. Suppose $f : M \to \mathbb{R}$ is a smooth map. We will firstly define partial derivative, $D_{x_i} f$, of f. On $\mathbb{R}^2$, we usually define the partial derivative, $\frac{\partial g}{\partial x_i}$, by taking limit. For example, $\frac{\partial g}{\partial x} = lim_{\triangle x \to 0} \frac{f(x+\triangle x, y) - f(x,y)}{\triangle x}$. With the conformal parametrization, we can define the partial derivative on scalar functions in the same manner. Because of the stretching effect, we have to modify the denominator in the limit a little bit. Specifically, we define (1):

$$D_x f = lim_{\triangle x \to 0} \frac{f \circ \phi(x+\triangle x, y) - f \circ \phi(x,y)}{dist(x+\triangle x, x)} = lim_{\triangle x \to 0} \frac{f \circ \phi(x+\triangle x, y) - f \circ \phi(x,y)}{\sqrt{\lambda} \triangle x} = \frac{1}{\sqrt{\lambda}} \frac{\partial f \circ \phi}{\partial x},$$

where λ is the conformal factor. $D_y f$ is defined similarly.

Now, the gradient of a function f, $\nabla_M f$, is characterized by: $df(Y) =< \nabla_M f, Y >$. Simple checking gives us: $\nabla_M f = \sum_{i,j} g^{ij}\partial_i f \partial_j$, where (g^{ij}) is the inverse of the Riemannian metric (g_{ij}).

With the conformal parametrization, we can define the *gradient* of f similar to the definition on $\mathbb{R}^2$. Namely, (2):

$$\begin{aligned}
&\nabla_M f = D_x f \mathbf{i} + D_y f \mathbf{j} \text{ where} \\
&(\mathbf{i}, \mathbf{j}) = (\tfrac{\partial}{\partial x}/\sqrt{< \tfrac{\partial}{\partial x}, \tfrac{\partial}{\partial x} >} \ , \ \tfrac{\partial}{\partial y}/\sqrt{< \tfrac{\partial}{\partial y}, \tfrac{\partial}{\partial y} >}) \\
&\qquad = \tfrac{1}{\lambda}[\tfrac{f\circ\phi}{\partial x}\tfrac{\partial}{\partial x} + \tfrac{f\circ\phi}{\partial y}\tfrac{\partial}{\partial y}]
\end{aligned}$$

Suppose $h : M \to \mathbb{R}$ is a smooth function. With this definition of gradient, we still have the following useful fact as in $\mathbb{R}^2$:

$$\begin{aligned}
\text{Length of } h^{-1}(0) &= \textstyle\int_M \delta(h)\sqrt{< \nabla_M h, \nabla_M h >} dS \\
&= \textstyle\int_M \sqrt{< \nabla_M H(h), \nabla_M H(h) >} dS \\
&= \textstyle\int_{\mathbb{C}} \delta(h\circ\phi)\sqrt{\lambda}\ ||\nabla h \circ \phi|| dxdy \\
&= \textstyle\int_{\mathbb{C}} \sqrt{\lambda}\ ||\nabla H(h\circ\phi)|| dxdy \qquad (3)
\end{aligned}$$

where H is the Heaviside function. (See Appendix)

Next, we need to give a well-defined definition of differential operator on vector field. This is based on the tensor calculus [9]. In Euclidean space, we conventionally differentiate the vector field $(x_1(t), ..., x_n(t))$ on a curve pointwisely to get $(x_1'(t), ..., x_n'(t))$. However, pointwise differentiation does not work for general manifolds because it is not coordinate invariant. For example, consider the parameterized circle in the plane given in Euclidean coordinate $(x(t), y(t)) = (cos\ t, sin\ t)$. Its acceleration at time t is $(-cos\ t, -sin\ t)$. However, in polar coordinates, the same curve is described as $(r(t), \theta(t)) = (1, t)$ and the acceleration is $(0, 0)$.

In order to differentiate a vector field $\overrightarrow{V}(t)$ along a curve, we have to write a difference quotient involving $\overrightarrow{V}(t)$ and $\overrightarrow{V}(t_0)$ which live on two different tangent spaces. Therefore, it is not appropriate to subtract. Secondly, even if we can differentiate the vector field pointwise, it is not guaranteed that the "derivative" is a tangent vector on the manifold.

We therefore need to define a differential operator on the vector field, which is coordinate invariant. This can be done by covariant differentiation $\nabla_X Y$, where X is called the direction of the differentiation. To do so, we need to develop a way to compare tangent vectors at different points. On $\mathbb{R}^2$, we usually parallelly translate the vectors and subtract. But on general manifolds, we do not have the concept of parallel translation. We say that a vector field $\overrightarrow{V}(\gamma(t))$ along a curve $\gamma(t)$ is *parallel* if: $D_t\overrightarrow{V}(\gamma(t))$ = orthogonal projection of $\frac{d}{dt}\overrightarrow{V}(\gamma(t))$ onto the tangent space = 0. We have the following important fact:

Parallel Translation : Given a curve $\gamma : I \to M$ and a vector $\overrightarrow{V}_0 \in T_{\gamma(t_0)}M$, there exists a unique parallel vector field $\overrightarrow{V}$ along γ with $\overrightarrow{V}(t_0) = V_0$.

With the parallel translation along a curve γ, we can define an operator: $P^{\gamma}_{t_0t_1} : T_{\gamma(t_0)}M \to T_{\gamma(t_1)}M$ by setting $P^{\gamma}_{t_0t_1}(\overrightarrow{V}_0) = V(t_1)$ where V is the parallel vector field along γ with $\overrightarrow{V}(0) = \overrightarrow{V}_0$. This is clearly an linear isomorphism.

Now, we can define $\nabla_X Y|_p$ as follows: let $\gamma : [0,1] \to M$ be a curve such that $\gamma(0) = p$ and $\gamma'(0) = Y|_p$. We define (4): $\nabla_X Y|_p = lim_{t\to 0}\frac{P^{\gamma}_{0t}{}^{-1}Y(\gamma(t))-Y(p)}{t}$

The covariant derivative satisfies the following properties:

(P1) $\nabla_{fX_1+gX_2}Y = f\nabla_{X_1}Y + g\nabla_{X_2}Y$ for $f, g \in C^{\infty}(M)$
(P2) $\nabla_X(aY_1 + bY_2) = a\nabla_X Y_1 + b\nabla_X Y_2$, $a, b \in \mathbb{R}$
(P3) $\nabla_X(fY) = f\nabla_X Y + (Xf)Y$ for $f \in C^{\infty}(M)$.

The above properties will determine the expression of the covariant derivative. Given a Riemannian manifold (M, g) where $g = (g_{ij})$ is the Riemannian metric. Suppose $\{\partial_i\}$ is the coordinate basis of the vector field. A simple verification will tell us the covariant derivative can be calculated by the following formula:

$$< \nabla_{\partial_i}\partial_j, \partial_l >= 1/2(\partial_i g_{jl} + \partial_j g_{li} - \partial_l g_{ij})$$

Simple calculation gives (5):

$$\nabla_{\partial_i}\partial_j = \Gamma^m_{ij}\partial_m \ where \quad \Gamma^m_{ij} = 1/2\ g^{ml}(\partial_i g_{jl} + \partial_j g_{li} - \partial_l g_{ij})$$

Suppose now the parametrization is conformal. The Riemannian metric (g_{ij}) is simply $(g_{ij}) = \lambda I$, where λ, I are the conformal factor and the identity matrix respectively. We then have the following formula (6):

$$\nabla_{\partial_x}\partial_x = \tfrac{1}{2\lambda}\ \tfrac{\partial\lambda}{\partial x}\ (\partial_x - \partial_y);\ \nabla_{\partial_y}\partial_y = \tfrac{1}{2\lambda}\ \tfrac{\partial\lambda}{\partial y}\ (-\partial_x + \partial_y);\ \nabla_{\partial_x}\partial_y = \tfrac{1}{2\lambda}\ (\tfrac{\partial\lambda}{\partial y}\partial_x + \tfrac{\partial\lambda}{\partial x}\partial_y)$$

With this formula and the above properties (P1)-(P3), we can calculate $\nabla_X Y$ easily. Thus for example:

$$\begin{aligned}\nabla_{a\frac{\partial}{\partial x}+b\frac{\partial}{\partial y}}\tfrac{\partial}{\partial y} &= a\nabla_{\frac{\partial}{\partial x}}\tfrac{\partial}{\partial y} + b\nabla_{\frac{\partial}{\partial y}}\tfrac{\partial}{\partial y} = \tfrac{a}{2\lambda}(\tfrac{\partial\lambda}{\partial y}\partial_x + \tfrac{\partial\lambda}{\partial x}\partial_y) + \tfrac{b}{2\lambda}(-\tfrac{\partial\lambda}{\partial y}\partial_x + \tfrac{\partial\lambda}{\partial y}\partial_y)\\ &= \tfrac{1}{2\lambda}\tfrac{\partial\lambda}{\partial y}(a-b)\partial_x + \tfrac{1}{2\lambda}(a\tfrac{\partial\lambda}{\partial x} + b\tfrac{\partial\lambda}{\partial y})\partial_y\end{aligned}$$

With the definition of covariant derivative, we can define the divergence of a vector field $\sum_{i=1}^{2} X_i\frac{\partial}{\partial x_i}$. The idea is to take the covariant derivative of X_i with respect to x_i and sum them up, we then get a scalar which is called the *divergence* of the vector field. For conformal parametrization, we have (7):

$$div_M(\Sigma_{i=1}^{2} X_i\tfrac{\partial}{\partial x_i}) = \textstyle\sum_{i=1}^{2}\frac{1}{\lambda}\partial_i(X_i\lambda)$$

If we calculate the divergence of $\nabla_M f$, we get the Laplacian of f:

$$\triangle_M f = \sum_{j=1}^{2}(1/\lambda)\ \partial_j\partial_j f \qquad (8)$$

Interestingly, with the above definitions, we still have the integration by part formula and the Green's formula:

$\int_M < \nabla_M u, X > dV = -\int_M u div_M X dV + \int_{\partial M} u < X, \overrightarrow{N} > d\widetilde{V}$, $\overrightarrow{N}$ is the unit normal vector. (Integration by part) (9)

$\int_M (u\triangle_M v - v\triangle_M u)dV = \int_{\partial M}(u\nabla_M v \cdot \overrightarrow{N} - v\nabla_M u \cdot \overrightarrow{N})d\widetilde{V}$
(Green's Theorem) (10)

Also, suppose C is a curve represented by the zero level set of $\phi : M \to \mathbb{R}$. We have the following useful property, similar to that on $\mathbb{R}^2$:

Geodesic curvature of $C = div_M(\frac{\nabla_M \phi}{||\nabla_M \phi||})$ (11) (See Appendix)

4 Navier-Stokes Equation on Surfaces

In this section, we will illustrate our idea by solving the Navier-Stokes equation on surfaces with arbitrary topologies. The idea is to parameterize the Riemann surface conformally onto the rectangular parameter domain based on the holomorphic differential one forms (Section 3.1). We then use the stable fluid solver [10] on the 2-D domain to solve the problem.

On $\mathbb{R}^2$, fluid flow is governed by the Navier-Stokes equation. For incompressible fluid flow, we have the following (*):

$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \bigtriangledown)\mathbf{u} + v\nabla^2\mathbf{u} + \mathbf{f}$ and $\nabla \cdot \mathbf{u} = 0$ (**imcompressibility**) (12)

where $\mathbf{u} = (u^1, u^2)$ is the fluid's velocity, v is the viscosity and $\mathbf{f} = (f^1, f^2)$ are external forces.

We can simulate the fluid flow as follow: we first use the stable fluid solver to solve (*). Then update the position of the fluid by $\mathbf{x}^{new} = \mathbf{x}^{old} + \mathbf{u}dt$, where $\mathbf{x}^{new}$ = updated position of the fluid particle and $\mathbf{x}^{old}$= previous position of the fluid particle.

To simulate fluid flow on the Riemann surface, we have to modify the 2D Navier-Stokes equation by the manifold version of gradient and lapacian. Replacing the gradient and laplacian by the manifold version of gradient and laplacian, we get the corresponding Navier-Stokes equation for the Riemann surface M:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla_M)\mathbf{u} + v\triangle_M \mathbf{u} + \mathbf{f} \quad (13)$$

Let ϕ be the conformal parametrization of M and $\mathbf{w} = \mathbf{u} \circ \phi$. We have:

$$\frac{\partial \mathbf{w}}{\partial t} = -\frac{1}{\lambda}(\mathbf{w} \cdot \bigtriangledown)\mathbf{w} + \frac{1}{\lambda}v\triangle\mathbf{w} + \mathbf{f} \quad (14)$$

Note that it is really the governing equation for fluids on the manifold — it is the same physics that we know. For detail, see Aris's book. [11]

We can next use the Stable Fluid Solver introduced by Stam to solve the Navier-Stokes equation. We describe the algorithm as follow:

Step 1: (Adding force) We solve: $\frac{\partial \mathbf{w}_1}{\partial t} = \mathbf{f}$. The iterative scheme is: $\mathbf{w}_1 = \mathbf{w}_0 + dt\mathbf{f}$
Step 2: (diffusion equation) We solve: $\frac{\partial \mathbf{w_2}}{\partial t} = \frac{1}{\lambda}v\triangle\mathbf{w}_1$. We use a simple implicit solver to get the iterative scheme: $(I - dt\frac{1}{\lambda}v\triangle)\mathbf{w}_2 = \mathbf{w}_1$.

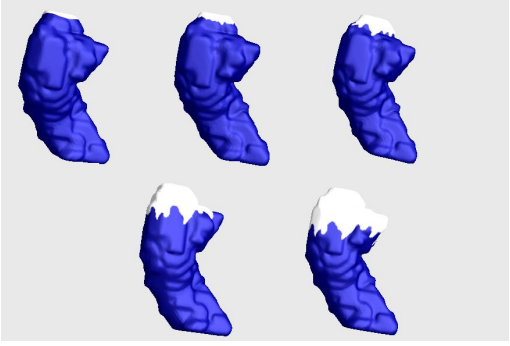


Fig. 2. Simulation of snow flowing down the surface

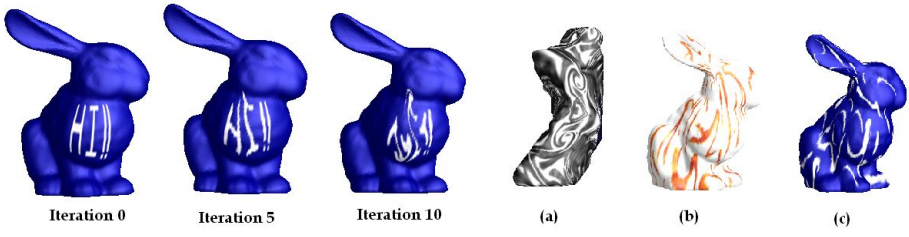


Fig. 3. Fluid flow on the surface in (A). Navier-Stoke's equation for surface decoration in (B).

Step 3: (advection equation) We solve: $\frac{\partial \mathbf{w}_3}{\partial t} = -\frac{1}{\lambda}(\mathbf{w}_2 \cdot \bigtriangledown)\mathbf{w}_3$. We use a semi-Lagrangian to get an iterative scheme: $\mathbf{w}_3 = \mathbf{w}_2(\mathbf{x} - dt\frac{1}{\lambda}\mathbf{w_2}(\mathbf{x}))$

Step 4: (projection) We project $\mathbf{w}$ onto its imcompressible (divergence free) component. For this, we first solve the Poisson equation: $\triangle\varphi = \nabla \cdot \mathbf{w}_3$
We then update: $\mathbf{w}_4 = \mathbf{w}_3 - \frac{1}{\lambda}\nabla\varphi$. Update $\mathbf{w} = \mathbf{w}_4$.

Step 5: (Update fluid position) Update $\mathbf{x}$ by $\mathbf{x}^{new} = \mathbf{x}^{old} + \mathbf{w}dt$

As an example, we simulate the snow flowing down the surface based on the Navier-Stokes equation in Figure 2. In Figure 3 (A), we simulate fluid flow on a bunny surface by adding a S-shaped force. In Figure 3 (B), we simulate fluid flow on surfaces for surface decoration.

5 Image Processing on Surfaces

5.1 Chan-Vese Segmentation on Surfaces

Segmentation is an important technique in image processing to extract useful region. One commonly used technique is the Chan-Vese (CV) segmentation technique, which is based on the level set method [12]. Here, we will extend the CV segmentation on $\mathbb{R}^2$ to arbitrary Riemann surface M.

Let $\phi : \mathbb{R}^2 \to M$ be the conformal parametrization of the surface M. We propose to minimize the following energy functional: (15)

$F(c_1, c_2, \psi) = \int_M (u_0 - c_1)^2 H(\psi) dS + \int_M (u_0 - c_2)^2 (1 - H(\psi)) dS + \nu length\ of\ \psi^{-1}(\{0\})$
$= \int_M (u_0 - c_1)^2 H(\psi) dS + \int_M (u_0 - c_2)^2 (1 - H(\psi)) dS + \nu \int_M |\nabla_M H(\psi)|_M dS,$

where $\psi : M \to \mathbb{R}$ is the level set function and $|\cdot|_M = \sqrt{<\cdot, \cdot>}$.

With the conformal parametrization, we have:

$F(c_1, c_2, \psi) = \int_{\mathbb{R}^2} \lambda (u_0 \circ \phi - c_1)^2 H(\psi \circ \phi) dxdy + \int_{\mathbb{R}^2} \lambda (u_0 \circ \phi - c_2)^2 (1 - H(\psi \circ \phi)) dxdy$
$+ \nu \int_{\mathbb{R}^2} \sqrt{\lambda} |\nabla H(\psi \circ \phi)| dxdy,$

For simplicity, we let $\zeta = \psi \circ \phi$ and $w_0 = u_0 \circ \phi$. Fixing ζ, we must have:

$$c_1(t) = \frac{\int_\Omega w_0 H(\zeta(t,x,y)) \lambda dxdy}{\int_\Omega H(\zeta(t,x,y)) \lambda dxdy} \qquad (16)$$

$$c_2(t) = \frac{\int_\Omega w_0 (1 - H(\zeta(t,x,y)) \lambda dxdy}{\int_\Omega (1 - H(\zeta(t,x,y))) \lambda dxdy} \qquad (17)$$

Fixing c_1, c_2, the Euler-Lagrange equation becomes:

$$\frac{\partial \zeta}{\partial t} = \lambda \delta(\zeta) [\ \nu \frac{1}{\lambda} \bigtriangledown \cdot (\sqrt{\lambda} \frac{\nabla \zeta}{||\nabla \zeta||}) - (w_0 - c_1)^2 + (w_0 - c_2)^2] \qquad (18)$$

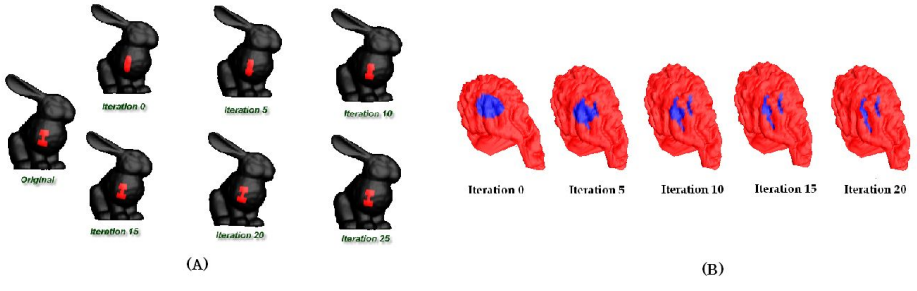


Fig. 4. CV segmentation on surface in (A). CV segmentation on surface for sulci extraction on the cortical surface in (B).

In Figure 4(A), we illustrate the CV segmentation on the bunny surface. As shown in the figure, the initial contour evolves to the original image in a few iterations. One application of CV segmentation is to extract the sulci position on the cortical surface. The sulci position is usually the high curvature region. We can consider the intensity as a function of curvatures, such as Mean curvatures and Gaussian curvatures. In Figure 4 (B), we illustrate how we can extract the sulci position on the cortical surface using CV segmentation. Here, we consider the mean curvature as the intensity.

5.2 Image Denoising on Surfaces

One important task of surface processing is the restoration or reconstruction of a true image u from an observed image f. In many applications, the measure image is polluted by noise or blur. The distorted image need to be denoised to understand the useful part of the image. On $\mathbb{R}^2$, Rudin, Osher and Fatemi (ROF) has proposed the following model [13]:

$$\mathbf{inf}_u F(u) = \int_\Omega |\nabla u| + \nu |f-u|^2 dxdy \qquad (19)$$

We proceed to extend the ROF on 2D to any surface M with arbitrary topologies. Let ϕ be the conformal parametrization of M and $\zeta = u \circ \phi$. Following the 2D ROF model, we propose to minimize the following energy:

$\mathbf{inf}_u F(u) = \int_M |\nabla_M u|_M + \nu |f-u|^2 dS \qquad (20)$ or

$\mathbf{inf}_u F(\zeta) = \int_{\mathbb{R}^2} \sqrt{\lambda} |\nabla \zeta| + \lambda \nu |f-\zeta|^2 dxdy \qquad (21)$

We can minimize the above energy by solving the Euler-Lagrange equation:

$\frac{\partial u}{\partial t} = 2\nu(f-u) + div_M(\frac{\nabla_M u}{|\nabla_M u|_M}) \qquad (22)$
or

$\frac{\partial \zeta}{\partial t} = 2\nu(f-\zeta) + \frac{1}{\lambda} div(\sqrt{\lambda}\frac{\nabla \zeta}{|\nabla \zeta|})$ on the rectangular parameter domain. (23)

and $\frac{\partial \zeta}{\partial \vec{n}} = 0$ on the boundary (24).

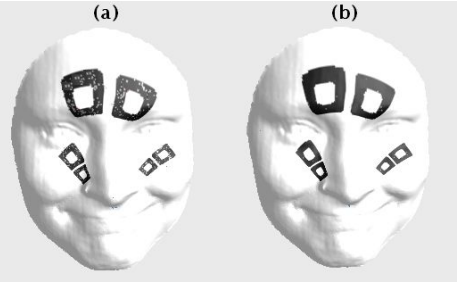


Fig. 5. ROF denoising on the human face

As an example, we use the ROF model to denoise the dirty scar on the human face in Figure 5. It is observed that the image can be significantly improved.

5.3 Image Inpainting on Surfaces

Inpainting, originally an artist's work, is the process of filling in the missing or desired image information where it is unavailable. (see Figure 6). Such "defect" domain may be introduced by the aging of the canvas and oil of an ancient

painting, and the occlusion by undesired objects in front of a scene of interest. For 2D images, Chan & Shen has introduced an inpainting model via curvature driven diffusion (CDD) [14]. We are going to extend this model to 3D Riemann surfaces.

Suppose Ω is the domain of the image on $\mathbb{R}^2$. Let D be the inpainting domain (the occluded region). The CDD model reads:

$$\frac{\partial u}{\partial t} = \nabla \cdot [\frac{g(|\kappa|)}{|\nabla u|}\nabla u], \qquad x \in D \qquad (25)$$

and $u = u^0, \qquad x \in D^c \qquad (26)$

Here κ denotes the curvature, and g(s) is defined to be zero if $s = 0$ and equal to infinity if $s = \infty$.

The curvature κ at a pixel x is the scalar curvature of the isophote through it and is given by: $\kappa = \nabla \cdot (\frac{\nabla u}{|\nabla u|}) \qquad (27)$

Suppose now Ω is the image domain on a Riemann surface M. $D \subset M$ is the inpainting domain. Let ϕ be the conformal parametrization of the surface and let $\zeta = u \circ \phi$. Replacing the gradient and divergence by the manifold version of gradient and divergence, we get the CDD inpainting model for the Riemann surface M:

$$\frac{\partial u}{\partial t} = div_M \cdot [\frac{g(|\kappa|)}{|\nabla_M u|_M}\nabla_M u] = \frac{1}{\lambda}\nabla \cdot [\frac{\sqrt{\lambda} g(|\kappa|)}{|\nabla \zeta|}\nabla \zeta], \quad x \in \phi^{-1}(D) \quad (29)$$

and $\zeta = \zeta^0, \qquad x \in \phi^{-1}(D^c) \quad (30)$

The curvature κ at a pixel x is given by:

$$\kappa = div_M \cdot (\frac{\nabla_M u}{|\nabla_M u|_M}) = \frac{1}{\lambda}\nabla \cdot (\sqrt{\lambda}\frac{\nabla \zeta}{|\nabla \zeta|}) \qquad (31)$$

In Figure 6, we illustrate the image inpainting on the human face. In (a), some region of the image is occluded. In (b), the image is effectively restored using the curvature driven diffusion inpainting technique.

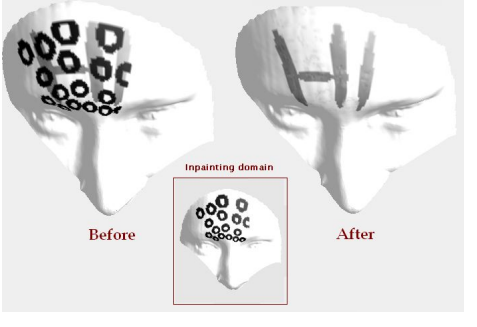


Fig. 6. Curvature driven diffusion inpainting on the human face

6 Conclusion and Future Work

In this paper, we propose a method to solve partial differential equations on surface with arbitrary topologies. The idea is to map the surface conformally onto

a simple parameter domain, namely, the 2D rectangle. We can next transform the PDE on the surface into a modified PDE on the 2D domain. We can then solve the PDE with the well-developed numerical schemes on $\mathbb{R}^2$. With the conformal parametrization, the differential operators defined on the surface closely resemble to the usual Euclidean counterpart, except for a multiplication of the conformal factor. Also, the parametrization of the surface using holomorphic 1-form allows us to parametrize (high genus)surface with the minimum number of coordinate chart. Thus, less boundary adjustment are needed when solving the PDEs on the surface. Finally, unlike the conventional way that projects the differential operators on $\mathbb{R}^3$ onto the surface, we directly define differential operators on the parameter domain without the need of doing projection. We thus avoid the complicated projection operation in our algorithm. We have illustrated our method by solving the Navier-Stokes equation on the surface. We also tested our method by solving some PDE-based surface processing problems, such as surface segmentation and surface denoising. In the future, we will look for more applications of solving PDEs on the surface.

References

1. Turk, G.: Generating textures on arbitrary surfaces using reaction-diffusion. Computer Graphic **25** (1991) 289–298
2. Dorsey, J., Hanrahan, P.: Digital materials and virtual weathering. Scientific American **282** (2000) 46–53
3. Stam, J.: Flows on surfaces of arbitrary topology. Proceedings of ACM SIGGRAPH 2003 **22** (2003) 724–731
4. Clarenza, U., Rumpfa, M., Teleaa, A.: Surface processing methods for point sets using finite elements. Computers and Graphics **28** (2004) 851–868
5. Bertalmio, M., Cheng, L.T., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. Journal of Computational Physics **174** (2001) 759 – 780
6. Memoli, F., Sapiro, G., Thompson, P.: Implicit brain imaging. Neuroimage **23** (2004) 179–188
7. Schoen, R., Yau, S.T.: Lectures on Harmonic Maps. International Press (1997)
8. Gu, X., Yau, S.T.: Global conformal surface parameterization. ACM Symposium on Geometry Processing 2003 (2003)
9. Syngen, J., Schild, A.: Tensor Calculus. Dover Publication (1949)
10. Stam, J.: Stable fluids. Proceedings of ACM SIGGRAPH 1999 (1999) 121–128
11. Aris, R.: Vectors, Tensors and the Basic Equations of Fluid Mechanics. Dover Publication (1989)
12. Vese, L.A., Chan, T.F.: Multiphase level set framework for image segmentation using the mumford and shah model. International Journal of Computer Vision **50** (2002) 271–293
13. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60** (1992) 259–268
14. Chan, T.F., Shen, J.: Non-texture inpainting by curvature-driven diffusions (cdd). J. Visual Comm. Image Rep. **12** (2001) 436–449

Appendix

Claim : Suppose $h : M \to \mathbb{R}$ is a smooth function. Then:
$Length\ of\ h^{-1}(0) = \int_M \sqrt{< \nabla_M H(h), \nabla_M H(h) >} dS = \int_{\mathbb{C}} \sqrt{\lambda}\ ||\nabla H(h \circ \phi)|| dxdy$

Proof :

Recall that the Co-area formula reads:
$\int_{\Omega \subset \mathbb{R}^2} f(x,y)|\nabla u| dxdy = \int_{\mathbb{R}} \int_{\{u(x,y)=r\}} f(x,y) d\mathcal{H} dr$
where $\mathcal{H}$ is the Hausdorff measure.

Let ϕ be the conformal parametrization of the surface M and $\zeta = u \circ \phi$.
Then,
$\int_M |\nabla_M H(u)|_M dS = \int_{\mathbb{R}^2} \delta(\zeta)|\nabla \zeta| \sqrt{\lambda} dxdy = \int_{\mathbb{R}} \int_{\{\zeta(x,y)=r\}} \sqrt{\lambda}\delta(\zeta) ds dr = \int_{\{\zeta(x,y)=0\}} ds$
$= \int_0^1 \sqrt{\lambda}|\mathbf{c}'(t)| dt = \int_0^1 \sqrt{\lambda}|\phi \circ \mathbf{c}'(t)| dt = length\ of \{u = 0\}$
where $\mathbf{c}(t)$ is the parametrization of $\zeta(x,y) = 0$ **Q.E.D.**

Claim : Let $\phi : M \to \mathbb{R}$. The geodesic curvature κ of $\phi^{-1}(\{0\}) = div_M(\frac{\nabla_M \phi}{\sqrt{<\nabla_M \phi, \nabla_M \phi>}})$

Proof :

Recall that the geodesic curvature of of a curve $\overrightarrow{\gamma}$

$= \frac{\sqrt{<D_t \dot{\overrightarrow{\gamma}}, D_t \dot{\overrightarrow{\gamma}}>}}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>} - \frac{<D_t \dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>^{3/2}} = \frac{<\dot{\overrightarrow{\gamma}}, D_t \dot{\overrightarrow{\gamma}}^{\perp}>}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>^{3/2}}$

Let the parametrization of the zero level set of ϕ be $\overrightarrow{\gamma} = (X(t), Y(t))$. Then $\phi(X(t), Y(t)) = 0$.

This implies (1): $< \nabla_M \phi, \dot{\overrightarrow{\gamma}} >= 0$

and (2): $< D_t(\nabla_M \phi), \dot{\overrightarrow{\gamma}} > + < D_t \dot{\overrightarrow{\gamma}}, \nabla_M \phi >= 0$

Now, $D_t \overrightarrow{V}(t) = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 (\dot{V}_k + \Gamma_{ij}^k \gamma_i V_j) \partial_k$

Thus, for conformal parametrization we have (3):

$D_t \dot{\overrightarrow{\gamma}} = (\ddot{X} + (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial x})(\dot{X}^2 - \dot{Y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial y}\dot{X}\dot{Y})\ ,\ \ddot{Y} - (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial y})(\dot{X}^2 - \dot{Y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial x})\dot{X}\dot{Y})$

and (4):

$D_t(\nabla_M \phi) = (\dot{\phi_x} + (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial x})({\phi_x}^2 - {\phi_y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial y}\phi_x \phi_y)\ ,\ \dot{\phi_y} - (\frac{1}{2\lambda}\frac{\partial \lambda}{\partial y})({\phi_x}^2 - {\phi_y}^2) - (\frac{1}{\lambda}\frac{\partial \lambda}{\partial x})\phi_x \phi_y\ )$

Combining (1), (2), (3), (4), we have: $\dot{X}^2 + \dot{Y}^2 = (1 + (\phi_x/\phi_y)^2)\dot{X}^2$ and $\frac{<D_t \dot{\overrightarrow{\gamma}}^{\perp}, \dot{\overrightarrow{\gamma}}>}{<\dot{\overrightarrow{\gamma}}, \dot{\overrightarrow{\gamma}}>^{3/2}} = \lambda(\dot{X}\ddot{Y} - \dot{Y}\ddot{X})$

$$= -\frac{\lambda}{\phi_y}[\phi_{xx}\dot{X}^2 + 2\phi_{xy}\dot{X}\dot{Y} + \phi_{yy}\dot{Y}^2]\dot{X} - \dot{X}(\dot{X}^2 + \dot{Y}^2)(\frac{1}{2\lambda}\frac{\partial\lambda}{\partial y})$$

$$+\dot{Y}(\dot{X}^2 + \dot{Y}^2)(\frac{1}{2\lambda}\frac{\partial\lambda}{\partial x})$$

So, $\kappa = \frac{<\vec{\dot{\gamma}}, D_t\vec{\dot{\gamma}}^{\perp}>}{<\vec{\dot{\gamma}}, \vec{\dot{\gamma}}>^{3/2}} = \frac{\lambda(\dot{X}\ddot{Y} - \dot{Y}\ddot{X})}{\lambda^{3/2}(\dot{X}^2+\dot{Y}^2)^{3/2}}$

$= \frac{1}{\sqrt{\lambda}}(\frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}) + \frac{1}{2\lambda^{3/2}}(\phi_x\frac{\partial\lambda}{\partial x} + \phi_y\frac{\partial\lambda}{\partial x})$

$= \frac{1}{\sqrt{\lambda}}\nabla\cdot(\frac{\nabla\phi}{|\nabla\phi|}) + \frac{1}{\lambda^{3/2}}\nabla\phi\cdot\nabla\lambda = \frac{1}{\lambda}\nabla\cdot(\lambda(\frac{1/\lambda\nabla\phi}{\sqrt{\lambda|\nabla\phi|^2}}))$

$= \frac{1}{\lambda}\nabla\cdot(\lambda(\frac{\nabla_M\phi}{\sqrt{<\nabla_M\phi, \nabla_M\phi>}})) = div_M(\frac{\nabla_M\phi}{\sqrt{<\nabla_M\phi, \nabla_M\phi>}})$ **Q.E.D.**

Fast Marching Method for Generic Shape from Shading

Emmanuel Prados and Stefano Soatto

Dept. of Computer Science, University of California, Los Angeles, CA 90095, USA

Abstract. We develop a fast numerical method to approximate the solutions of a wide class of equations associated to the Shape From Shading problem. Our method, which is based on the control theory and the interfaces propagation, is an extension of the "Fast Marching Method" (FMM) [30,25]. In particular our method extends the FMM to some equations for which the solution is not systematically decreasing along the optimal trajectories. We apply with success our one-pass method to the Shape From Shading equations which are involved by the most relevant and recent modelings [22,21] and which cannot be handled by the most recent extensions of the FMM [26,8].

1 Introduction

The *Shape From Shading* (SFS) problem is to compute the three-dimensional shape of a surface from a single black and white image of that surface. The field was pioneered by Horn who was the first to pose the problem as that of finding the solution of a nonlinear first-order partial differential equation (PDE) called the brightness equation [11]. The first explicit PDE considered in this field is the Eikonal equation

$$|\nabla u| = r(x), \quad \forall x \in \Omega \tag{1}$$

(modeling based on a single far frontal light source and orthographic camera). Ω is an open subset of $\mathbb{R}^2$ which represents the image domain, e.g., the rectangle $]0, X[\times]0, Y[$. $r : \Omega \rightarrow \mathbb{R}$ is a non-negative function directly related to the brightness image. From the work of Horn [11], a number of explicit PDEs corresponding to different modelings have been developed and studied [15,24,27,20,6]. By introducing the "generic" SFS equation $H_g(x, \nabla u(x)) = 0, \forall x \in \Omega$ (H_g being described below), Prados and Faugeras [22,21] have recently unified a number of these explicit equations and thus their associated models. The associated "generic" SFS Hamiltonian is defined by

$$H_g(x, p) = \kappa_x \sqrt{|D_x \; R_x p + \mathbf{v}_x|^2 + K_x^2} + \mathbf{w}_x \cdot p + c_x, \tag{2}$$

where $\kappa_x, K_x, \mu_x, \nu_x, c_x, \mathbf{v}_x, \mathbf{w}_x$ depend on the chosen modeling, see [22,17]. For example, the classical Rouy/Tourin Hamiltonian [24]

$$H_{R/T}(x, p) = I(x)\sqrt{1 + |\nabla u|^2} + \mathbf{l} \cdot \nabla u - \gamma \tag{3}$$

(orthographic camera + single far light source) is a particular case of the generic SFS Hamiltonian H_g; $\mathbf{L} = (\mathbf{l}, \gamma)$ represents the direction of the light source; $I(x)$ is the

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 320–331, 2005.

brightness of the image at pixel x. Here, let us remark that $H_{R/T}(x,0) = I(x) - \gamma$ can be strictly negative.

In addition to the modeling, Prados and Faugeras [17,21] have also unified a number of theoretical results and of SFS numerical methods based on PDEs. Nevertheless they just consider iterative methods. Also, the iterative methods suffer of some optimality weaknesses since they use alternating raster scans similarly as the ones proposed [9,29]. In this paper, we get rid of these weaknesses by proposing a single pass method based on front propagation and Fast Marching techniques.

The single pass methods related to front propagation like the level set method [16] and Fast Marching Methods [25] have already been applied to the SFS problem [12,25,13,31,28] (see [5,25] and references therein for other applications). In particular, Sethian [25] was the first who applies the "Fast Marching Method"for solving the SFS problem. This first work deals only with the simplest version of the SFS problem associated to the Eikonal equation (1) (orthographic camera; single, far and front light source). Recently, Kimmel and Sethian [13] have proposed an adaptation of this initial algorithm in order to deal with far oblique light source. This upgraded method seems very delicate, requires a change of variable and does not seems adaptable to more general modelings. Roughly speaking, most of the authors [13,28,6] use various techniques (e.g. changes of variables, introductions of new unknowns) in order to get back to the Eikonal equation and then to use the initial tools developed by Sethian. As opposed to [28], Yuen et al. [31] propose a real adaptation of the FMMs to the perspective SFS problem. Nevertheless, this work is reduced to frontal light source, also in that case the subjacent cost function is non negative. Contrary to all the previous work, the fast method we propose in this paper allows to solve in one-pass the main PDEs associated to the most recent, realistic and relevant modelings of the SFS literature. In particular, it allows to solve all the SFS equations collected in [22,21] (many of which have cost functions of arbitrary sign). It is completely generic and contrary for example to [13], we do not need any change of variables.

More generally, the most recent work on "Fast Marching Methods" only allows to solve equations of type

$$H(x, \nabla u(x)) = 0 \quad \forall x \in \Omega, \tag{4}$$

such that $H(x,0) = 0$ (with H convex with respect to ∇u). Here, we relax this key assumption which is strongly related to the *causality principle* (see Section 3).

2 Control Formulation of the Problem

In this section we reformulate the problem using tools from control theory. The reader unfamiliar with these tools can refer to the comprehensive book [1]. First, we use the Legendre transform [14] to rewrite equation (4) as: $\forall\, x \in \Omega$,

$$\sup_{a \in A} \{-f(x,a) \cdot \nabla u(x) - l(x,a)\} = 0. \tag{5}$$

The functions l and f are respectively called the *cost function* and the *dynamics*; $a \in A$ is an admissible value of the *control*. For example, we can rewrite the generic SFS Hamiltonian H_g as such a supremum with

$$f(x,a) = -\left[\, \kappa_x R_x^T D_x R_x . a + \mathbf{w}_x \,\right],$$
$$l(x,a) = -\left[\, K_x \kappa_x \sqrt{1-|a|^2} + \kappa_x (R_x^T \mathbf{v}_x) \cdot a + c_x \,\right]$$

with A the closed unit ball of $\mathbb{R}^2$; R_x^T being the matrix transpose of R_x; see [17,22] for all details. This kind of equation must be complemented by boundary conditions in order to be well posed. We therefore add the following constraints:

$$u(x) = \varphi(x) \qquad \forall x \in \mathcal{T}, \tag{6}$$
$$u(x) = +\infty \qquad \forall x \in \partial\Omega \setminus \mathcal{T}, \tag{7}$$

where the *target* $\mathcal{T}$ is a nonempty closed subset of $\overline{\Omega}$ and $\varphi : \mathcal{T} \to \mathbb{R}$ defines a boundary condition of Dirichlet type. Also equations (5) and (4) are now considered on $\Omega \setminus \mathcal{T}$ instead of Ω. Let V be the value function

$$V(x) = \inf_{\alpha \in \mathcal{A}} \left\{ \int_0^{\tau_x(\alpha)} l(y_x(\tau,\alpha),\alpha(\tau)) d\tau \;\; + \varphi(y_x(\tau_x(\alpha),\alpha)) \right\}. \tag{8}$$

$\mathcal{A}$ is the set of the *admissible controls* (a set of the measurable functions of $t \in [0,+\infty[$ with value in A, the latter being a closed bounded subset of $\mathbb{R}^M$). $y_x(.,\alpha)$ is a *trajectory* controlled by α starting from x; this is the solution of the dynamical system $y'(\tau) = f(y(\tau),\alpha(\tau))$, $\tau > 0$, $y(0) = x$; $\tau_x(\alpha)$ is the smallest time τ such that $y_x(\tau)$ reaches[1] $\mathcal{T} \cup \partial\Omega$. It is well known [14,1] that V is the unique viscosity solution[2] of equation (5)-(6)-(7) and that V verifies the *dynamical programming principle* [14,1].

Finally, we denote α_x^*, τ_x^* and y_x^* the optimal control, the optimal time and the *optimal trajectory* associated to (8) (i.e. for which the $\inf$ of (8) is reached).

For example, in the particular case of the Eikonal equation (1), the optimal trajectories correspond to the gradient lines. Nevertheless, as shown in [26], this is not true for any equation ($\nabla u(x) \neq y_x^{*\prime}(0) = f(x,\alpha_x^*(0))$).

3 Approximation Scheme and Causality

For simplicity, in this paper we deal with a regular mesh. For an extension to the irregular meshes we refer the reader to [17,23]. The reader unfamiliar with the notion of approximation schemes can refer to [2]. Let us just recall that, following [2], an approximation scheme is a functional equation of the form

$$S(\rho, x, u(x), u) = 0 \quad \forall x \in \overline{\Omega},$$

which "approximates" the considered PDE. S is defined on $\mathcal{M} \times \overline{\Omega} \times \mathbb{R} \times B(\overline{\Omega})$ into $\mathbb{R}$, $\mathcal{M} = \left(\mathbb{R}^+\right)^N$ and $\rho = (h_1, ..., h_N) \in \mathcal{M}$ defines the size of the mesh that is used in the

[1] For notations simplicity, we extend φ on $\mathcal{T} \cup \partial\Omega$, and we define $\varphi(x) = +\infty, \forall x \in \partial\Omega \setminus \mathcal{T}$.

[2] More accurately, we must consider the notion of "Singular Discontinuous Viscosity Solutions" (SDVS) of (5)-(6)-(7) instead of the classical notion of viscosity solutions [17].

corresponding numerical algorithms. $B(D)$ is the space of bounded functions defined on a set D. In order to obtain a consistent scheme, the function S is obtained by approximating ∇u by finite differences and then by substituting ∇u by its approximation in the initial equation. The main property allowing to ensure the convergence of the computed numerical solution toward the viscosity solution [7,14,1,17] is the *monotonicity* of the scheme (i.e. the function $u \mapsto S(\rho, x, t, u)$ is nonincreasing) [2].

In most of the FMMs the gradient ∇u is discretized by

$$\nabla u(x) \simeq \frac{t - u(x + s_i h_i e_i)}{-s_i h_i}$$

Above, t corresponds to $u(x)$; we replace $u(x)$ by t in order to emphasize that this is the *update value*. $(e_1, .., e_n)$ is the canonical basis; $\forall i,\ s_i \in \{\pm 1\}$. With the exception of the recent work [26], the simplex $\{x, x + s_1 h_1 e_1, ..., x + s_n h_n e_n\}$ (in practice the sign vector $(s_1, .., s_n)$) is chosen in such a way that "it contains $-\nabla u$", i.e. such that

$$\forall i = 1..n,\ \ s_i \text{ is the opposite of the sign of } [\nabla u]_i. \tag{9}$$

In the Eikonal case, the control formulation of equation (1) is $\sup_{a \in \overline{B}(0,1)} \{a \cdot \nabla u(x) - r(x)\} = 0$ and the optimal control is always $a_x = \frac{\nabla u(x)}{|\nabla u(x)|}$. Thus the scheme

$$\begin{aligned} S_{eiko}(\rho, x, t, u) &= \left| \frac{t - u(x + s_i h_i e_i)}{-s_i h_i} \right| - r(x) \\ &= \sup_{a \in \overline{B}(0,1)} \left\{ a \cdot \left(\frac{t - u(x + s_i h_i e_i)}{-s_i h_i} \right) - r(x) \right\}, \end{aligned} \tag{10}$$

where s_i defined by (9), is clearly monotonic. Nevertheless, this is generally false for any equation of the form (4). In effect, in the general case, the scheme

$$\begin{aligned} S_c(\rho, x, t, u) &= H\left(x, \left(\frac{t - u(x + s_i h_i e_i)}{-s_i h_i} \right)\right) \\ &= \sup_{a \in A} \{-f(x, a) \cdot \left(\frac{t - u(x + s_i h_i e_i)}{-s_i h_i} \right) - l(x, a)\} \end{aligned} \tag{11}$$

with s_i defined by (9), is not monotonic anymore as soon as there exists $i \in [1..n]$ s.t. $sign[f(x, a_x)]_i \neq s_i\ (= -sign[\nabla u]_i)$, where a_x is the optimal a of (11) [t being the root of $S_c(\rho, x, t, u) = 0$]. For obtaining a monotonic scheme, we must change the choice of the simplex (i.e. the choice of the s_i). In fact, we must choose the simplex with respect to the dynamics of the optimal control (and not to the gradient). If we define $s_i(x, a) = sign f_i(x, a)$, the scheme[3]

$$S(\rho, x, t, u) = \sup_{a \in A} \{-f(x, a) \cdot \left(\frac{t - u(x + s_i(x, a) h_i e_i)}{-s_i(x, a) h_i} \right) - l(x, a)\} \tag{12}$$

is naturally monotonic, and the "good" one. In other words, this scheme is obtained from (4) by using only the backward and forward approximations of the partial derivatives and by choosing the good one *according to the dynamic of the optimal control*.

[3] Scheme already suggested by Prados and Faugeras [20,17].

Because of space limitations we omit the details of the implementation of this scheme and refer the reader to [23,17]. Let us just note here that the approximation scheme proposed by Sethian and Vladimirsky is different from the one presented here: [26] is based on the use of more simplexes than those contained in the immediate neighbors of the considered site, unlike the scheme (12). Moreover, our scheme is consistent (see [17]).

Our scheme suggests then that the reconstruction must follow (track) the optimal trajectories, if we want to be able to recover the viscosity solution with a one-pass algorithm. Sethian and Vladimirsky [26] demonstrate this fundamental aspect intuitively and experimentally. As opposed to the first work [30,25] in which the causality was directly related to the gradient lines, in [26] the causality property is based on optimal trajectories. Nevertheless, Sethian and Vladimirsky's causality is based on the fact that the solution is strictly decreasing along the optimal trajectories. This property is verified for the equations considered in [26]

$$\sup_{a \in \mathcal{A}} -\mathsf{f}(x,a)a \cdot \nabla u - 1 = 0, \quad \forall x \in \Omega, \tag{13}$$

since for these equations the cost function is $l(x,a) = 1 > 0$. Nevertheless, for any equation of type (4) such that the cost function l takes some negative values, this monotonicity property does not hold: the solution can arbitrary and alternatively increase and decrease along the optimal trajectories. Also, in the general case the causality property used by Sethian and Vladimirsky [26] does not apply.

Finally, let us emphasize that as it was shown in [17], most of the Shape From Shading equations have generally cost functions of arbitrary sign. This is the case for example for the classical Rouy/Tourin Hamiltonian $H_{R/T}$ (where, $l_{R/T}(x,a) = I(x)\sqrt{1+|a|^2} - \gamma$) and for the perspective SFS Hamiltonian $H^{pers}_{P/F}$ [20] which fit in the class of Hamiltonians given by equation (4).

In the following, we slightly reinterpret the FMMs; we generalize and weaken the principle of causality. Later on we propose a new practical causality property which extends the one used by [26] to any equation of type (4).

4 Update Ordering for "Single Pass" Method

4.1 Related Work and Basic Ideas

For the moment, let us assume that *we know the optimal trajectories* y^*_x. We can then recover directly the solution of equation (5)-(6)-(7) by going back up these curves:

$$V(x) = \int_0^{\tau_x(\alpha^*_x)} l(y^*_x(\tau), \alpha^*_x(\tau))d\tau \ + \varphi(y^*_x(\tau_x(\alpha^*_x))). \tag{14}$$

Then from the numerical point of view, we can reconstruct the solution by a direct integration along the optimal trajectories. This idea corresponds roughly to the method of *characteristic strips* introduced in the Shape from Shading literature by Horn [10] for solving the Eikonal equation. In this work, the knowledge of the optimal trajectories was implicitly replaced by Neumann conditions. For removing some "resticking"

problems and for improving the stability of the method, we can go back up all the optimal trajectories simultaneously and not separately. More precisely, the idea is then the following. We assume that the solution is known inside a closed curve $\mathcal{C}_t$ ($t \geq 0$) which propagates along the optimal trajectories. Basically, the curves $\mathcal{C}_t$ must verify: for all x in Ω, the optimal trajectory y_x^* always intersects the curve $\mathcal{C}_t$, but only once for each $t \geq 0$. Moreover, if $t_1 < t_2$ then $\Omega_{t_1} \subset \Omega_{t_2}$ (where Ω_t is the open subset such that $\partial\Omega_t = \mathcal{C}_t$). This idea corresponds exactly to the one introduced by Bruckstein [3] for solving the Eikonal SFS problem. To alleviate some instability and topological problems [16], the best way is then to use the "level set" method introduced by Osher and Sethian [16] or Fast Marching techniques [25].

4.2 Curve Propagations and Associated Costs

Now, we assume that *we do not know the optimal trajectories*. This is usually the case in most of the applications, in particular in Shape From Shading. In this case, we must reconstruct simultaneously the solution u and the curves $\mathcal{C}_t$. In this way, we are sure that the updates of the values of the solution propagate in the same way as the optimal trajectories. The idea is then the following: let us assume that we already know $\mathcal{C}_t$ and the values of the solution on Ω_t. Now, if we want to compute the values of the solution u on Ω_{t+dt}, we need to compute explicitly these values, but also we need to compute the new curve $\mathcal{C}_{t+dt}$. In order to *practically handle the curves $\mathcal{C}_t$, we define them as the level sets* (i.e. the isocontours) of a *"cost"* $\mathsf{C}(x)$ defined on Ω into $\mathbb{R}$. For example, in the basic "Fast Marching Method" for the Eikonal equation [30,25], the chosen cost C is equal to u. Also, the curves $\mathcal{C}_t$ correspond to the isocontours of the solutions. In other respects, in a sense the function $\tilde{z}$ introduced in [13] could be consider as such a cost, (nevertheless let us note that $\tilde{z}$ does not depend on x but on $\tilde{x}$ [which himself depends on x and $u(x)$]; see [13]).

In order to reach our goal, let us remind an important (but too many times neglected) difficulty encountered when we deal with equations of type (5)-(6)-(7): the optimal trajectories y_x^* and the solution V significantly depend on the set Ω. Therefore, in the sequel, when it is relevant, we indicate the associated set by specifying it by an index as follow: $\alpha_x^* = \alpha_{x(\Omega)}^*$, $y_x^* = y_{x(\Omega)}^*$, $\tau_x^* = \tau_{x(\Omega)}^*$, $V = V_\Omega$.
Finally, to be completely rigorous, if the optimal trajectories depend on the set on which we work, then the curves $\mathcal{C}_t$ and their associated cost C must also depend on it. Also, we will use the same type of notations as above for the cost C (nevertheless, in the sequel we use the notations $\mathcal{C}_t$ and Ω_t associated with the iso-curves only in reference to Ω and equation (5)-(6)-(7)).

The above remark is mainly motivated by the following idea. First, let us remind that we do not know the cost C_Ω on $\Omega \setminus \mathcal{C}_t$ (since this is equivalent to knowing all the propagating closed curves). Moreover, it seems reasonable to assume that we are not able to directly compute C_Ω even on a very close neighborhood $\mathcal{N}$ of Ω_t ($\mathcal{N} \subset \Omega$). In effect, any point $x \in \mathcal{N} \setminus \overline{\Omega}_t$ (even if it is extremely close to $\mathcal{C}_t$) can have an optimal trajectory $y_{x(\Omega)}^*$ which goes away very far from $\mathcal{C}_t$ before coming back to it. Nevertheless it is reasonable to assume that we are able to compute (an approximation of) $\mathsf{C}_\mathcal{N}$ on a close neighborhood $\mathcal{N}$ of Ω_t (let us remind that $\mathsf{C}_\mathcal{N}$ is the cost associated to the equation $(5_\mathcal{N})$-$(6_\mathcal{N})$-$(7_\mathcal{N})$).

For example in [26], the neighborhood $\mathcal{N}$ (considered above) corresponds with the set of the "*Considered*" points. In the classical methods, the cost C corresponds with the solution of the considered equation. Also, the values $V(x)$ computed for all the "*Considered*" points x are (numerical approximation of) the cost $\mathsf{C}_{\mathcal{N}}(x)$.
Later on, from the knowledge of the cost $\mathsf{C}_{\mathcal{N}}$ on the neighborhood $\mathcal{N}$ of Ω_t, we would like to find $\mathcal{C}_{t+dt}$ such that $\mathcal{C}_t \subset \mathcal{C}_{t+dt} \subset \mathcal{N}$. Also, if the costs verify the hypotheses

(H1) $\mathsf{C}_{\mathcal{W}}(x)$ is decreasing along the optimal trajectories $y^*_{x(\mathcal{W})}$, i.e. : for all x and $\tau > 0$ (small enough) $\mathsf{C}_{\mathcal{W}}(y^*_{x(\mathcal{W})}(\tau)) \leq \mathsf{C}_{\mathcal{W}}(x)$,
(H2) it is decreasing with respect to $\mathcal{W}$, i.e.:
if $\Omega_1 \subset \Omega_2$, then $\mathsf{C}_{\Omega_1}(x) \geq \mathsf{C}_{\Omega_2}(x)$,
(H3) let $x_1 \in \Omega_1 \subset \Omega_2$. If $y^*_{x_1(\Omega_2)}(.)$ stays in Ω_1 then $\mathsf{C}_{\Omega_2}(x_1) = \mathsf{C}_{\Omega_1}(x_1)$,

then we have the following

Proposition 1. *Let us assume that the hypotheses* (H1)*,* (H2) *and* (H3) *hold. Let* $\mathcal{C}_{t+dt}$ *be such that* $\mathcal{C}_t \subset \mathcal{C}_{t+dt} \subset \mathcal{N}$*. Therefore for all* x *in* Ω_{t+dt}*,*

$$\mathsf{C}_{\Omega}(x) = \mathsf{C}_{\mathcal{N}}(x).$$

In other words, $\mathsf{C}_{\mathcal{N}}$ coincide with C_{Ω} on Ω_{t+dt}.

Proof. Let x in Ω_{t+dt}. By the hypothesis (H1), the optimal trajectory $y^*_{x(\Omega)}$ stays in Ω_{t+dt}. We then apply the hypothesis (H3) with $\Omega_1 = \Omega_{t+dt}$ and $\Omega_2 = \Omega$. We have therefore $\mathsf{C}_{\Omega}(x) = \mathsf{C}_{\Omega_{t+dt}}(x)$. Since we have $\mathcal{C}_{t+dt} \subset \mathcal{N} \subset \Omega$, then by the hypothesis (H2), $\mathsf{C}_{\Omega}(x) = \mathsf{C}_{\Omega_{t+dt}}(x) = \mathsf{C}_{\mathcal{N}}(x)$. □

Now, let us consider the problem from the discret point of view. We assume that the set $\overline{\Omega}$ is "covered" by a grid of points. We divide the set of the grid points into three classes (as for the classical "Fast Marching Methods" [26]): *Accepted* $\mathbf{A}$, *Considered* $\mathbf{C}$, *Far* $\mathbf{F}$. The *Accepted* points are the ones in Ω_t (i.e. inside $\mathcal{C}_t$); we therefore suppose that we already know the values of the solution for all the grid points in $\mathbf{A}$. The set of *Considered* points $\mathbf{C}$ is the set of the adjacent points to the *Accepted* points. The union $\mathbf{C} \cup \mathbf{A}$ is the discrete version of the neighborhood $\mathcal{N}$ of $\overline{\Omega}_t$. The set of the *Far* points corresponds to the other points of the grid.

Let us remind that we know the values of the solution of (5)-(6)-(7) on Ω_t and that we want to compute the ones on Ω_{t+dt}. Also, this requires the preliminary computation of the new curve $\mathcal{C}_{t+dt}$. From the practical and discrete point of view, we want to extend Ω_t to Ω_{t+dt} in such a way that $\Omega_{t+dt} \setminus \Omega_t$ contains a single point of the grid. This is therefore equivalent to finding the point x_0 of $\mathbf{C}$ which has the lowest cost C_{Ω} ($\mathsf{C}_{\Omega}(x_0) = \min_{x \in \mathbf{C}} \mathsf{C}_{\Omega}(x)$) and then we transfer it to the set of *Accepted* points $\mathbf{A}$ (which is then enlarged). Also, if the costs C verify the hypotheses (H1), (H2) and (H3), then the following proposition allows to find this point x_0 (assuming that we know the costs $\mathsf{C}_{\mathcal{N}}(x)$ for all x in $\mathcal{N}$).

Proposition 2. *Let us assume that the costs* C *verify* (H1)*,* (H2) *and* (H3)*. The point* $x_0 \in \mathbf{C}$ *which has the smallest cost* $\mathsf{C}_{\Omega}(x_0)$ *is the point* x *of* $\mathbf{C}$ *which has the smallest cost* $\mathsf{C}_{\mathcal{N}}(x)$*; Also,* $\mathsf{C}_{\Omega}(x_0) = \mathsf{C}_{\mathcal{N}}(x_0)$*.*

Proof. Let x_0 be the point in $\mathbf{C}$ which has the less cost $\mathsf{C}_{\Omega}(x_0)$. Let $\mathcal{C}_{t+dt}$ be the level set of the cost C_{Ω} associated to the value $\mathsf{C}_{\Omega}(x_0)$. Since $\mathcal{C}_t \subset \mathcal{C}_{t+dt} \subset \mathcal{N}$, By Proposition 1 we have $\mathsf{C}_{\Omega}(x_0) = \mathsf{C}_{\mathcal{N}}(x_0)$.

Let x be any point in $\mathbf{C}$. By hypothesis (H2), we have $\mathsf{C}_{\mathcal{N}}(x) \geq \mathsf{C}_{\Omega}(x)$. By definition of x_0, we have $\mathsf{C}_{\Omega}(x) \geq \mathsf{C}_{\Omega}(x_0)$. Therefore, $\mathsf{C}_{\mathcal{N}}(x) \geq \mathsf{C}_{\mathcal{N}}(x_0)$. □

Thus if (H1), (H2) and (H3) hold, then from the numerical point of view, for extending Ω_t to Ω_{t+dt}, we have just to compute the costs $\mathsf{C}_{\mathcal{N}}(x)$ for all x in the $\mathbf{C}$ and to search the point x_0 in $\mathbf{C}$ which has the smallest cost $\mathsf{C}_{\mathcal{N}}$.

4.3 Proposed Costs

At this stage, let us remind that the method we propose here needs the explicit computations of some approximations of the cost C and of the solution V. Since our final objective is only to compute some approximations of V, the computations of the approximations $\mathsf{C}_{\mathcal{N}}$ can appear useless and expensive, and so decrease the interest of the method. It seems therefore reasonable and quite relevant to search and use some costs $\mathsf{C}_{\mathcal{W}}$ which are directly related to the values of the solution $V_{\mathcal{W}}$, as in the Eikonal case and the classical FMMs.

Example 1. In [26], Sethian and Vladimirsky consider equations of type (13), i.e. some equations (5) with $f(x,a) = \mathsf{f}(x,a)a$ and $l(x,a) = 1$. So the viscosity solution V of (13) (complemented by some adequate boundary conditions) is $V(x) = \tau_x^* + \varphi(y_x^*(\tau_x^*))$. Thus, if the boundary condition φ imposed on $\mathcal{T}$ is a constant function ($\varphi(x) = c \in \mathbb{R}$ for all x in $\mathcal{T}$) then $V(x) = \tau_x^* + c$. In this case, it is therefore judicious and reasonable to choose $\mathsf{C}_{\mathcal{W}}(x) = \tau^*_{x(\mathcal{W})}$. $V_{\mathcal{W}}$ and the $\mathsf{C}_{\mathcal{W}}$ are thus directly related; Consequently the curves $\mathcal{C}_t$ correspond with the isocontours of the solution of (13). Also, the costs $\mathsf{C}_{\mathcal{W}}$ verify trivially the hypotheses (H1), (H2) and (H3).

In the particular case considered by Sethian and Vladimirsky [26] (Example 1), the cost directly related to the values of the solution is trivial. Nevertheless in the general case, in particular for HJB equations with a cost fonction $l(x,a)$ depending on a or with an arbitrary sign, we need to work a little more.

Let us remind that generally the isocontours of the solution cannot play the role of the curves $\mathcal{C}_t$ since the values of the solution u are *not decreasing* along the optimal trajectories (i.e. for all x, the function $t \mapsto u(y_x^*(t))$ is not decreasing). A second basic idea could be to choose systematically the cost $\mathsf{C}(x) = \tau_x^*$ since this cost verifies naturally the hypotheses (H1), (H2) and (H3). Nevertheless, computing this cost can be very difficult, and generally, it is really not related with the solution u of the considered equation. In order to define an adequate cost in the general case, let us introduce the following definitions.

Let ψ be a subsolution of (4), i.e.

$$H(x, \psi(x)) \leq 0, \quad \forall x \in \Omega.$$

In [17], Prados and Faugeras describe the subsolutions of the main classical SFS equations. For example, for the classical Rouy/Tourin Hamiltonian $H_{R/T}$ (equation (3)) $\psi(x) = -\frac{1}{\gamma}\mathbf{1} \cdot x$ is subsolution.

Let $\mathcal{Z}(x)$ be the multivalued map [4] on $\overline{\Omega}$ defined as:

$$\mathcal{Z}(x) = \{p \in \mathbb{R}^N : H(x,p) \leq 0\}.$$

Let $\delta : \overline{\Omega} \times \mathbb{R}^N \rightarrow \mathbb{R}$ be the support function [4] of the set $\tilde{\mathcal{Z}}(x) = \mathcal{Z}(x) - \nabla\psi(x)$, i.e.:

$$\delta(x,p) = \max\{pq : q \in \tilde{\mathcal{Z}}(x)\}.$$

Finally, for all x_1, x_2 in $\overline{\mathcal{W}}$, let us denote

$$L_{\mathcal{W}}(x_1,x_2) = \inf_{\xi \in \Xi_{x_1,x_2}} \left\{ \int_0^1 \delta(\xi(t), -\dot{\xi}(t))dt \right\} \tag{15}$$

where Ξ_{x_1,x_2} is the set of $\xi(t) \in W^{1,\infty}([0,1], \overline{\mathcal{W}})$ s.t. $\xi(0) = x_1$ and $\xi(1) = x_2$. A complete study and description on these notions can be found in [17,4,19,18]. In particular, in [17], it is proved that

$$V(x) = \psi(x) + \min\{L_{\Omega\setminus\mathcal{T}}(x,z) + \varphi(z) - \psi(z) \mid z \in \mathcal{T}\}. \tag{16}$$

is the unique Singular Discontinuous Viscosity Solution (SDVS) of (5)-(6)-(7), see [17]. Let us emphasize here that $\delta(.,.)$ and so $L_{\mathcal{W}}$ are non-negative. Also $L_{\mathcal{W}}(x,z)$ is a semi-distance.

These results and notations in hand, we can now define an appropriate cost for our "generic" HJB equation:

$$\mathsf{C}_{\mathcal{W}}(x) = L_{\mathcal{W}}(x, z_x^*),$$

where z_x^* is the optimal z of (16). This cost is trivially decreasing along the optimal trajectories and it verifies naturally the assumptions (H1)-(H2)-(H3). Finally, if the boundary condition φ verifies "$\varphi - \psi$ is constant on the target $\mathcal{T}$" then the viscosity solution $V_{\mathcal{W}}$ is directly related to $\mathsf{C}_{\mathcal{W}}$: it coincides with $\mathsf{C}_{\mathcal{W}} + \psi$ (up to the addition of a constant). Also, the isocontour of the cost $\mathsf{C}(x)$, and so $\mathcal{C}_t$, corresponds with the isocontour of the $V - \psi$.

5 Proposed Algorithm

The grid points are divided into the three classes (as in the basic "Fast Marching Method"): *Accepted* $\mathbf{A}$, *Considered* $\mathbf{C}$, *Far* $\mathbf{F}$ (see above). ψ is a subsolution of $H(x, \nabla u) = 0$ as defined in Section 4.3. U defined on the whole grid is the approximation of the solution of equation (5)-(6)-(7). $\tilde{U}$ defined on the *Considered* points is the approximation of the solution of equation $(5_{\mathcal{N}})$-$(6_{\mathcal{N}})$-$(7_{\mathcal{N}})$, where $\mathcal{N}$ is the neighborhood associated to $\mathbf{A} \cup \mathbf{C}$.

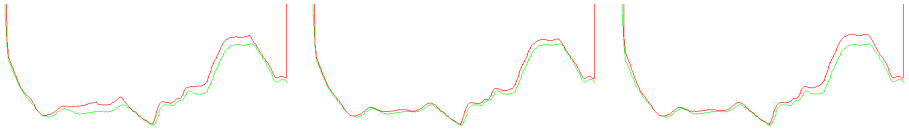
Algorithm 1.

1. *Start with all the grid points in* Far.
2. *Move the grid points on the boundary $\partial\Omega$ and on the target $\mathcal{T}$ to the* Accepted*. For all these points $x \in \mathcal{T}$, put $U(x) = \varphi(x)$, for the other points put $U(x) = +\infty$.*
3. *Move all the grid points $x \in \Omega$ adjacent to the* Accepted *points into* Considered *and for each of these points, evaluate $\tilde{U}(x)$ by using the update scheme* (12)*.*
4. *Find the grid point $x_0 \in \mathbf{C}$ with the smallest value $\tilde{U}(x) - \psi(x)$. Move x_0 from* Considered *to* Accepted*; Put $U(x_0) = \tilde{U}(x_0)$.*
5. *Move from* Far *into* Considered*, all the* Far *points which are adjacent to x_0.*
6. *Re-evaluate $\tilde{U}$ by using the scheme* (12)*, for all the* Considered *points adjacent x_0.*
7. *If the set* $\mathbf{C}$ *is not empty, return to item 4.*

The complete proof of the convergence of the computed numerical solution toward the viscosity solution is postponed to a following paper.

a) b) c) d)

Fig. 1. a) original surface (groundtruth); b) image synthesized from a); c) surface reconstructed by using the classical causality; d) surface reconstructed by our algorithm



a) b) c)

Fig. 2. a) vertical section of Fig.1-c); b) vertical section of Fig.1-d); c) vertical section of the surface reconstructed by the iterative algorithm after three complete iterations

6 Numerical Experiments

We have implemented our method for the generic SFS Hamiltonian H_g (2). As a consequence our algorithm applies to any modeling considered in [21,22].

Because of space and since the contributions of this paper concern mainly some numerical improvements, here we illustrate our results only with the classical Rouy/Tourin Hamiltonian $H_{R/T}$ (3). We also only deal with synthetic images; more exactly with the classical example of the Mozarts face [32]. Moreover as the theory suggests [19], in our experiments we assume that we know and we use the values of the solutions at all its local minima.

Mainly, the differences between our new method and the previous FMMs are two fold. 1) We propose a new *causality principle* (the updating order is related to the level sets of $u - \psi$ instead of the ones of u; u being the solution of the considered equation). 2) We propose also to use an approximation scheme (already described in [17]) which is different from the previous ones proposed in the FMM literature. Here, we focus on the improvement due to the change of the causality. Figure 1 shows: a) the original surface; b) the image synthesized from (a) with a single far light source ($\mathbf{l} = (-0.3, -0.3)$; see page 320) and by using an orthographic projection; c) the surface reconstructed by our algorithm [our scheme + our causality]; d) the surface reconstructed by using the classical causality [our scheme + replacing ψ by 0]. Let us note that due to the black shadows, we are not able to completely recover the original surface. Nevertheless, the improvement involves by the change of the causality is quite visible. To better show the differences between the surfaces 1-c) and 1-d), we display vertical sections of them in

Figures 2-a) and 2-b). In Figure 2, the red curves represent the sections of the computed approximations and the green curve is the section of the groundtruth. In other respects, Figure 2-c) displays a vertical section of the surface reconstructed by the corresponding iterative algorithm [22,17] after three complete alternating raster scans similar to those used in [9,29]. In this example, the number of pixels considered is around 20200. With our FMM method, the computation of the solution requires around 40500 updates. With the iterative version, around 76000 updates are required for computing an approximation of the solution with an error of the same order of magnitude. About computational time, our FMM method returns this result after 6 seconds (computer: Intel Celeron 1.5GHz; Let us note that we do not have tried to optimize our code yet). The iterative version requires approximatively the same computational time (7 seconds). Many more numerical results and qualitative and quantitative comparison tests can be found in our forthcoming associated papers [23]. Also, we postpone to [23] the comparisons of the results obtained with Prados and Faugeras's scheme with those obtained with the other ones.

7 Conclusion

In this article, we revisit the classical "Fast Marching Methods" [30,25,26,8] and we extend them to a wide class of HJ equations. In particular, our method can deal with HJB equations with arbitrary signs cost functions when the previous methods just deal with positive cost functions. Also, in the cases where the solution does not decrease along the optimal trajectories, we correct the causality property by using a subsolution. Finally, our method is generic and it applies indifferently to all the SFS equations obtained by the most recent and relevant modelings of this problem [17].

References

1. M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhauser, 1997.
2. G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4:271–283, 1991.
3. A. M. Bruckstein. On shape fron shading. *Computer. Vision Graphics Image Process*, 44:139–154, 1988.
4. F. Camilli and A. Siconolfi. Nonconvex degenerate Hamilton-Jacobi equations. *Mathematische Zeitschrift*, 242:1–21, 2002.
5. L. Cohen. Minimal paths and fast marching methods for image analysis. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Mathematical Models in Computer Vision: The Handbook*, chapter 7. Springer, 2005.
6. F. Courteille, A. Crouzil, J-D. Durou, and P. Gurdjos. Towards shape from shading under realistic photographic conditions. In *Proceedings of ICPR'04*, 2004.
7. M.G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton–Jacobi equations. *Trans. AMS*, 277:1–43, 1983.
8. E. Cristiani and M. Falcone. Fast Marching Semi-Lagrangian methods for the Eikonal equation. In *SIMAI 2004*, pages 20–24, Venezia, Italy, September 2004.
9. P.-E. Danielsson. Euclidean Distance Mapping. *Computer Graphics and Image Processing*, 14(3):227–248, 1980.

10. B.K. Horn. *Robot Vision*. MIT Press, 1986.
11. B.K.P. Horn. Obtaining shape from shading information. In P.H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.
12. R. Kimmel and A.M. Bruckstein. Tracking level sets by level sets : A method for solving the shape from shading problem. *CVIU*, 62(2):47–58, July 1995.
13. R. Kimmel and J.A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(2):237–244, May 2001.
14. P.-L. Lions. *Generalized Solutions of Hamilton–Jacobi Equations*. Number 69 in Research Notes in Mathematics. Pitman Advanced Publishing Program, 1982.
15. J. Oliensis and P. Dupuis. Direct method for reconstructing shape from shading. *SPIE*, 1570:116–128, 1991.
16. S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: algorithms based on the Hamilton–Jacobi formulation. *Journal of Comput. Physics*, 79:12–49, 1988.
17. E. Prados. *Application of the theory of the viscosity solutions to the Shape From Shading problem*. PhD thesis, Univ. of Nice-Sophia Antipolis, 2004.
18. E. Prados, F. Camilli, and O. Faugeras. A unifying and rigorous shape from shading method adapted to realistic data and applications. *Accepted to the Journal of Mathematical Imaging and Vision (JMIV)*, 2005.
19. E. Prados, F. Camilli, and O. Faugeras. A viscosity solution method for shape-from-shading without boundary data. *Submitted to ESAIM Mathematical Modelling and Numerical Analysis*, 2005.
20. E. Prados and O. Faugeras. "Perspective Shape from Shading" and viscosity solutions. In *Proceedings of ICCV'03*, volume 2, pages 826–831, October 2003.
21. E. Prados and O. Faugeras. Unifying approaches and removing unrealistic assumptions in Shape from Shading: Mathematics can help. In *Proceedings of ECCV'04*, 2004.
22. E. Prados and O. Faugeras. A generic and provably convergent shape-from-shading method for orthographic and pinhole cameras. *accepted to IJCV*, 2005.
23. E. Prados and S. Soatto. Fast Marching Method for Hamilton-Jacobi-Bellman equations and applications. Technical report, UCLA, to appear in 2005; Journal version in preparation.
24. E. Rouy and A. Tourin. A Viscosity Solutions Approach to Shape-from-Shading. *SIAM Journal of Numerical Analysis*, 29(3):867–884, June 1992.
25. J.A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
26. J.A. Sethian and A. Vladimirsky. Ordered upwind methods for static hamilton–jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis*, 41(1):325–363, 2003.
27. A. Tankus, N. Sochen, and Y. Yeshurun. A new perspective [on] Shape-from-Shading. In *Proceedings of ICCV'03*, volume 2, pages 862–869, October 2003.
28. A. Tankus, N. Sochen, and Y. Yeshurun. Perspective Shape-from-Shading by Fast Marching. In *Proceedings of CVPR'04*, 2004.
29. Y.-H. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM J. Numer. Anal.*, 41(2):673–694, 2003.
30. J.N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control*, 40:1528–1538, 1995.
31. S.Y. Yuen, Y.Y. Tsui, Y.W. Leung, and R.M.M. Chen. Fast marching method for shape from shading under perspective projection. In *Proceedings of VIIP'02*, pages 584–589, 2002.
32. R. Zhang, P.-S. Tsai, J.-E. Cryer, and M. Shah. Shape from Shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999.

A Gradient Descent Procedure for Variational Dynamic Surface Problems with Constraints

Jan Erik Solem and Niels Chr. Overgaard

Applied Mathematics Group,
School of Technology and Society,
Malmö University, Sweden
{jes, nco}@ts.mah.se

Abstract. Many problems in image analysis and computer vision involving boundaries and regions can be cast in a variational formulation. This means that m-surfaces, e.g. curves and surfaces, are determined as minimizers of functionals using e.g. the variational level set method. In this paper we consider such variational problems with constraints given by functionals. We use the geometric interpretation of gradients for functionals to construct gradient descent evolutions for these constrained problems. The result is a generalization of the standard gradient projection method to an infinite-dimensional level set framework. The method is illustrated with examples and the results are valid for surfaces of any dimension.

1 Introduction

Variational formulations have been successfully used by many researchers to solve a wide variety of problems within computer vision and image analysis. Benefits of using variational formulations are among others; a solid mathematical framework, well developed numerical techniques, and the fact that a variational formulation clearly and unambiguously shows the type of solutions that are sought.

From a variational formulation one can determine what the solutions look like. If they cannot be computed directly from the Euler-Lagrange equations a gradient descent procedure can be used, cf. [1]. This paper deals with gradient descent for variational problems involving dynamic m-dimensional surfaces and interfaces with side conditions. The side conditions appear in the form of constraints on the solutions. Constraints can appear in "finite" form such as boundary conditions for curves and surfaces and in "infinite" form, defined by functionals. Here the latter type is considered.

Variational problems with such constraints appear naturally in many vision applications such as overlapping of regions in multi-phase segmentation [2,3] and as obstacles for surface fitting problems [4].

This paper introduces mathematical techniques that give a geometric interpretation and introduces an infinite-dimensional gradient projection method as an extension of the finite-dimensional theory, cf. [5]. A projected gradient is

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 332–343, 2005.

introduced as the orthogonal projection on a manifold determined by the constraint functional and we derive the corresponding gradient descent procedure. In doing so, we show that a gradient descent Lagrange method is in fact a projection. Furthermore, we give some illustrative examples and treat issues related with practical implementations.

The paper is organized as follows; background material on finite-dimensional constrained optimization, the level set method and infinite-dimensional gradient descent is covered in Section 2. Our contributions are described in Sections 3 to 5. In Section 3 we derive the projected gradient on the constraint manifold and in Section 4 we give some illustrative examples. Finally, we comment on the practical implementation in Section 5.

2 Background

As a courtesy to the reader, the necessary background on finite dimensional constrained optimization, the level set method, and the geometric gradient interpretation for variational m-surface problems is briefly recalled here.

2.1 Gradient Projection for Finite Dimensional Problems

Suppose we are asked to find the minimum of $f(\mathbf{x})$ subject to the side condition $g(\mathbf{x}) = 0$, where $f, g : \mathbf{R}^{m+1} \to \mathbf{R}$ are a pair of differentiable functions. Set $N = \{\mathbf{x} \in \mathbf{R}^{m+1} : g(\mathbf{x}) = 0\}$ and assume for simplicity that $\nabla g(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in N$. Then N is a differentiable m-dimensional surface, by the implicit function theorem, and the above minimization problem becomes that of finding $\mathbf{x}^* \in N$ such that

$$f(\mathbf{x}^*) = \min_{\mathbf{x} \in N} f(\mathbf{x}). \tag{1}$$

The classical method for solving such a problem is to use Lagrange multipliers: If $\mathbf{x}^*$ minimizes f on N, then there exists a constant $\lambda^* \in \mathbf{R}$ such that

$$\nabla f(\mathbf{x}^*) - \lambda^* \nabla g(\mathbf{x}^*) = 0.$$

Thus, $(\mathbf{x}^*, \lambda^*)$ is to be found among the stationary points of the Lagrange function $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$. However, computing the stationary points of $L(\mathbf{x}, \lambda)$ is generally a highly non-trivial matter, so in practice one tries to find $\mathbf{x}^*$ directly by using some kind of descent method.

Let us describe the construction of a gradient descent procedure for the minimization problem (1). First, let $\mathbf{x} \in N$ and define the *gradient* $\nabla_N f(\mathbf{x})$ *of* f *on* N *at* $\mathbf{x} \in N$ by

$$\nabla_N f(\mathbf{x}) = \nabla f(\mathbf{x}) - \frac{\nabla g(\mathbf{x}) \cdot \nabla f(\mathbf{x})}{\|\nabla g(\mathbf{x})\|^2} \nabla g(\mathbf{x}), \tag{2}$$

where $\mathbf{v} \cdot \mathbf{w}$ denotes the usual scalar product between vectors $\mathbf{v}, \mathbf{w} \in \mathbf{R}^{m+1}$, and $\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$ is the corresponding norm. Notice that the N-*gradient* $\nabla_N f(\mathbf{x})$ is

simply the orthogonal projection of the usual gradient $\nabla f(\mathbf{x})$ onto the tangent space of N at $\mathbf{x}$: $T_{\mathbf{x}}N = \{\mathbf{v} \in \mathbf{R}^{m+1} : \nabla g(\mathbf{x}) \cdot \mathbf{v} = 0\}$, hence the term: *the gradient projection method*, cf. Figure 1.

Next, given a point $\mathbf{x}_0 \in N$ then the gradient descent motion for (1) is the solution of the initial value problem:

$$\dot{\mathbf{x}}(t) = -\nabla_N f(\mathbf{x}(t)), \qquad \mathbf{x}(0) = \mathbf{x}_0. \tag{3}$$

If the limit $\mathbf{x}^\circ = \lim_{t\to\infty} \mathbf{x}(t)$ exists then $\nabla_N f(\mathbf{x}^\circ) = 0$, hence $\mathbf{x}^\circ$ is a stationary point of f's restriction to the manifold N. Hopefully the gradient descent motion solves the minimization problem (1), i.e. $\mathbf{x}^\circ = \mathbf{x}^*$, but $\mathbf{x}^\circ$ may of course turn out to be some other stationary point.

The purpose of this work is to generalize this procedure to variational problems for curves and surfaces, where both the objective function and the side conditions are given by functionals.

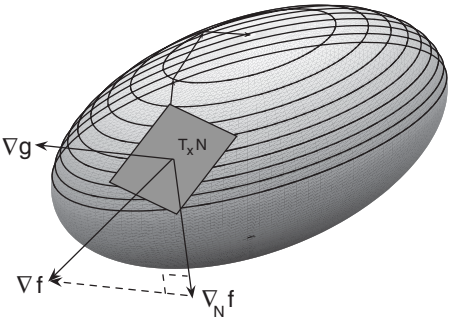


Fig. 1. Projected gradient descent on an ellipsoid-shaped manifold defined as the set $N = \{\mathbf{x} \in \mathbf{R}^{m+1} : g(\mathbf{x}) = 0\}$. The gradient is projected on the tangent space using the gradient of the constraint, ∇g.

2.2 The Kinematics of Dynamic Surfaces

A regular m-surface in $\mathbf{R}^{m+1}$ has codimension equal to one and can be represented implicitly as the zero set of a differentiable function $\phi : \mathbf{R}^{m+1} \to \mathbf{R}$, the *level set function*, as

$$\Gamma = \{\mathbf{x} \ : \ \phi(\mathbf{x}) = 0\} \ . \tag{4}$$

The sets $\Omega = \{\mathbf{x} : \phi(\mathbf{x}) < 0\}$ and $\{\mathbf{x} : \phi(\mathbf{x}) > 0\}$ are called the *inside* and the *outside* of Γ, respectively. Using this convention, the outward unit normal $\mathbf{n}$ and the mean curvature κ of Γ are given by (c.f. [6])

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad \text{and} \quad \kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \ . \tag{5}$$

The implicit representation introduced above can be used to define a *dynamic surface* (or *surface evolution*), $t \to \Gamma(t)$, by adding a time dependence, $\phi =$

$\phi(\mathbf{x}, t)$, to the level set function, where $\phi : \mathbf{R}^{m+1} \times I \to \mathbf{R}$ is a sufficiently differentiable function. The dynamic surface is then given by

$$t \mapsto \Gamma(t) := \{\mathbf{x} \ : \ \phi(\mathbf{x}, t) = 0\} \ . \tag{6}$$

Let us recall from [1] the notion of the *normal velocity* of the surface evolution (6). Suppose a particle moves along with the dynamic surface $\Gamma(t)$. If the motion of the particle is given by the parameterized curve $\boldsymbol{\alpha} : I \to \mathbf{R}^{m+1}$ with $\boldsymbol{\alpha}(0) = \mathbf{x}_0$, then the equality $\phi(\boldsymbol{\alpha}(t), t) = 0$ holds identically at all times t. Differentiation of this identity yields

$$\dot{\boldsymbol{\alpha}}(0) \cdot \mathbf{n} = -\frac{\partial\phi(\mathbf{x}_0, 0)/\partial t}{|\nabla\phi(\mathbf{x}_0, 0)|} \ , \tag{7}$$

where the left-hand side is the normal component of the velocity $\dot{\boldsymbol{\alpha}}(0)$ of the particle at $t = 0$. This normal component is an intrinsic property of the evolution since it does not depend on the choice of $\boldsymbol{\alpha}$ or $\phi(\mathbf{x}, t)$, cf. [1]. We can then define the *normal velocity* of the evolution $\Gamma(t)$ as the function

$$\dot{\Gamma}(t) = -\frac{\partial\phi(\mathbf{x}, t)/\partial t}{|\nabla\phi(\mathbf{x}, t)|} \qquad (\mathbf{x} \in \Gamma(t)) \ . \tag{8}$$

Using the notation $v = v(\Gamma) = -\dot{\Gamma}(t)$ we can rewrite this equation as

$$\frac{\partial\phi}{\partial t} = v|\nabla\phi| \ , \tag{9}$$

where we have dropped the dependence on $\mathbf{x}$ and t to simplify the notation. This is the well-known *level set equation* which is the basis for the level set method, introduced independently by [7] and [8] as a tool for evolving implicit surfaces.

2.3 Geometric Gradient Descent for Dynamic Surfaces

In this section we recall from [1] the construction of gradient descent evolutions for the minimization of functionals $E(\Gamma)$ defined on manifolds of admissible m-surfaces Γ. Here we are primarily concerned with functionals of the following types

$$E_\circ(\Gamma) = \int_\Gamma g(\mathbf{x})\, d\sigma \quad \text{or} \quad E_\bullet(\Gamma) = \int_\Omega g(\mathbf{x})\, d\mathbf{x} \ , \tag{10}$$

where $\Gamma = \partial\Omega$ is a closed m-surface, $d\sigma$ the Euclidean surface measure, and $g : \mathbf{R}^{m+1} \to \mathbf{R}$ is a given weight function.

As in [1], let M denote the (pre-)manifold of admissible m-surfaces Γ. If $\Gamma \in M$ then the *tangent space of* M *at* Γ is the set $T_\Gamma M$ of all functions $v : \Gamma \to \mathbf{R}$ such that v correspond to the normal velocity of some (regular) surface evolution through Γ. Each tangent space $T_\Gamma M$ of M is endowed with a scalar product $\langle \cdot, \cdot \rangle_\Gamma$ defined as the integral

$$\langle v, w \rangle_\Gamma = \int_\Gamma v(\mathbf{x}) w(\mathbf{x})\, d\sigma \qquad (v, w \in T_\Gamma M) \ . \tag{11}$$

If the norm of $v \in T_\Gamma M$ is defined by by $\|v\|_\Gamma = \sqrt{\langle v, v \rangle_\Gamma}$, then we have Schwarz' inequality:

$$|\langle v, w \rangle_\Gamma| \leq \|v\|_\Gamma \|w\|_\Gamma \qquad (v, w \in T_\Gamma M) \ . \tag{12}$$

Now, consider a functional $E : M \to \mathbf{R}$ and let $\Gamma \in M$ be fixed. The functional E is said to be Gâteaux-differentiable at Γ, if the derivative

$$dE(\Gamma)v = \frac{d}{dt} E(\Gamma(t))\Big|_{t=0} \tag{13}$$

exists for every $v \in T_\Gamma M$. Here $\Gamma(t)$ is a surface evolution which satisfies $\Gamma(0) = \Gamma$ and $\dot{\Gamma}(0) = v$. The linear functional on the left hand side of (13) is called the Gâteaux derivative (or the functional derivative) of E at Γ. There sometimes exists a vector $\nabla E(\Gamma) \in T_\Gamma M$ such that the following identity holds for all normal velocities $v \in T_\Gamma M$:

$$dE(\Gamma)v = \langle \nabla E(\Gamma), v \rangle_\Gamma \qquad \text{(Riesz)} \ . \tag{14}$$

If this is the case, then $\nabla E(\Gamma)$ is called the *gradient of E at Γ*, and it is uniquely determined by the property (14)[1].

The gradient descent for the variational problem $E(\Gamma^*) = \min_\Gamma E(\Gamma)$ is, analogous to (3), given by the solution of the following initial value problem:

$$\dot{\Gamma}(t) = -\nabla E(\Gamma(t)); \qquad \Gamma(0) = \Gamma_0 \ , \tag{15}$$

where Γ_0 is the initial m-surface.

As an example we apply this procedure to the two functionals in (10), and derive the corresponding gradient descent evolutions in the level set framework. First we notice that the Gâteaux derivatives of these functionals are $dE_\circ(\Gamma)v = \int_\Gamma (\nabla g \cdot \mathbf{n} + g\kappa) v \, d\sigma$ and $dE_\bullet(\Gamma)v = \int_\Gamma g v \, d\sigma$, respectively. The first of these derivatives, which is the classical *geodesic active contours*, is derived in [1,9,10], and the second derivative can be found in e.g. [1,11]. Using (14) we see that

$$\nabla E_\circ(\Gamma) = \nabla g \cdot \mathbf{n} + g\kappa \quad \text{and} \quad \nabla E_\bullet(\Gamma) = g \ . \tag{16}$$

Using the formula (8) for the normal velocity it follows from (15) that the gradient descent evolutions for the minimization of $E_\circ$ and $E_\bullet$ are

$$\frac{\partial \phi}{\partial t} = (\nabla g \cdot \mathbf{n} + g\kappa)|\nabla \phi| \quad \text{and} \quad \frac{\partial \phi}{\partial t} = g|\nabla \phi| \ , \tag{17}$$

respectively, where the initial level set function $\phi_0(\mathbf{x}) = \phi(\mathbf{x}, 0)$ must be specified.

[1] It would be more correct to use the notation $\nabla_M E$ for the gradient of E, as it is actually the gradient of E on the manifold M of admissible m-surfaces. However, always insisting on correct names ultimately leads to cumbersome notation, and since functionals on M are always denoted by upper case letters, we trust the reader understands that ∇E means the (functional) gradient in the infinite-dimensional setting of surfaces.

3 Gradient Projection for Variational Surface Problems

In this section we show how the notion of a functional gradient, defined in Section 2.3, can be used to give a geometric interpretation of descent evolutions for variational level set problems with constraints. Let $F, G : M \to \mathbf{R}$ be two Gâteaux-differentiable functionals of either of the forms in (10). Define N as the sub-manifold of admissible m-surfaces Γ given by

$$N = \{\Gamma \in M : G(\Gamma) = 0\} \ ,$$

and consider the constrained variational problem of finding $\Gamma^* \in N$ such that

$$F(\Gamma^*) = \min_{\Gamma \in N} \ F(\Gamma) \ . \tag{18}$$

Assume that $\nabla G \neq 0$ on N. If Γ^* solves (18) then, according to the Lagrange multiplier method, there exists a number $\lambda^* \in \mathbf{R}$ such that the pair (Γ^*, λ^*) is a stationary point of the Lagrange function $L(\Gamma, \lambda) = F(\Gamma) - \lambda G(\Gamma)$, that is, (Γ^*, λ^*) solves the following system of equations:

$$\begin{cases} \nabla L(\Gamma, \lambda) = \nabla F(\Gamma) - \lambda \nabla G(\Gamma) = 0 \\ \\ \dfrac{\partial L}{\partial \lambda} L(\Gamma, \lambda) = G(\Gamma) = 0 \ . \end{cases} \tag{19}$$

In order to find (Γ^*, λ^*) we construct a gradient descent motion $t \mapsto (\Gamma(t), \lambda(t))$ for the Lagrange function L in such a way that the constraint $\partial L/\partial \lambda = 0$ is enforced at all times. This means that $t \to \Gamma(t)$ solves the initial value problem:

$$\dot{\Gamma}(t) = -\nabla L(\Gamma(t), \lambda(t)), \qquad \Gamma(0) = \Gamma_0 \ , \tag{20}$$

where it remains to determine the value of $\lambda = \lambda(t)$ in (20). To do this, we differentiate the identity $G(\Gamma(t)) = 0$, in the second equation of the system (19),

$$\begin{aligned} 0 = \frac{d}{dt} G(\Gamma) = dG(\Gamma)\dot{\Gamma} &= -\langle \nabla G(\Gamma), \nabla L(\Gamma, \lambda) \rangle_\Gamma \\ &= -\langle \nabla G(\Gamma), \nabla F(\Gamma) - \lambda \nabla G(\Gamma) \rangle_\Gamma \end{aligned}$$

then we see that

$$\lambda = \frac{\langle \nabla F(\Gamma), \nabla G(\Gamma) \rangle_\Gamma}{\|\nabla G(\Gamma)\|_\Gamma^2} \ .$$

It follows from this calculation that the right-hand side of (20) is $\nabla L(\Gamma, \lambda) = \nabla_N F(\Gamma)$, where

$$\nabla_N F(\Gamma) := \nabla F(\Gamma) - \frac{\langle \nabla F(\Gamma), \nabla G(\Gamma) \rangle_\Gamma}{\|\nabla G(\Gamma)\|_\Gamma^2} \nabla G(\Gamma) \tag{21}$$

is the *N-gradient of F at $\Gamma \in N$*, which is defined as the orthogonal projection (in $T_\Gamma M$) of ∇F onto the tangent space $T_\Gamma N = \{v \in T_\Gamma M : \langle \nabla G(\Gamma), v \rangle_\Gamma = 0\}$,

precisely as in the finite dimensional case in Section 2.1. In other words, the gradient for the Lagrange function is a projection. In the level set formulation, the N-gradient descent motion for (18) becomes

$$\frac{\partial \phi}{\partial t} = \nabla_N F \, |\nabla \phi| \ , \tag{22}$$

where $\nabla_N F$ is given by (21). This result is easily generalized to variational problems with several constraints.

Proposition 1. *Let $F, G_1, \ldots, G_n : M \to \mathbf{R}$ be Gâteaux differentiable functionals, and assume that the functional gradients $\nabla G_1(\Gamma), \ldots, \nabla G_n(\Gamma)$ are linearly independent on $N = \{\Gamma \in M : G_1(\Gamma) = \cdots = G_n(\Gamma) = 0\}$. Then the gradient descent motion for the minimization problem (18) is the solution of the initial value problem:*

$$\dot{\Gamma}(t) = -\nabla_N F(\Gamma(t)), \qquad \Gamma(0) = \Gamma_0 \ . \tag{23}$$

Here the N-gradient $\nabla_N F$ is given by

$$\nabla_N F(\Gamma) = \nabla F(\Gamma) - \sum_{i=1}^{n} \lambda_i \nabla G_i(\Gamma) \ ,$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ solves the following system of equations:

$$\sum_{j=1}^{n} \lambda_j \langle \nabla G_j, \nabla G_i \rangle_\Gamma = \langle \nabla F, \nabla G_i \rangle_\Gamma \qquad (i = 1, \ldots, n) \ . \tag{24}$$

That is, $\nabla_N F(\Gamma)$ is the orthogonal projection of F onto the tangent space $T_\Gamma N$.

Proof. The dynamic surface defined by (23) is the gradient descent motion for the Lagrange function $L(\Gamma, \boldsymbol{\lambda}) = F(\Gamma) - \sum_{i=1}^{n} \lambda_i G_i(\Gamma)$ which satisfies the conditions $\partial L / \partial \lambda_i(\Gamma(t)) = 0$ for $i = 1, \ldots, n$. The latter implies that the equalities $G_i(\Gamma(t)) = 0$, $i = 1, \ldots, n$ hold for all t. Differentiation of these identities gives the system (24) for the Lagrange multipliers, which is solvable, by the assumption on the functional gradients $\nabla G_1, \ldots, \nabla G_n$. The details are left to the reader.

4 Examples

In this section we will show some examples of how to apply the geometric analysis above. First we use a classical result from differential geometry as a pedagogical example to illustrate the theory in Section 3, then we give a practical case where such analysis can be used for surface fitting to 3D data.

4.1 Illustrative Example: The Isoperimetric Problem

Let us define the length and area functionals for a closed planar curve as

$$L(\Gamma) = \int_\Gamma d\sigma \quad \text{and} \quad A(\Gamma) = \int_\Omega d\mathbf{x} \ , \tag{25}$$

where, as above, Ω is the interior of Γ. Since in both cases these expressions correspond to $g(\mathbf{x}) = 1$ in (10), the gradients are simply

$$\nabla L = \kappa \quad \text{and} \quad \nabla A = 1 \ , \tag{26}$$

from (16).

The isoperimetric problem (IP) is to *find the curve with a fixed length that encloses the largest area*, or equivalently, *find the shortest curve which encloses a fixed area*. If we decide to use the second formulation we can define the manifold satisfying the constraint as

$$N = \{\Gamma \ : \ A(\Gamma) = A_0\} \ ,$$

for some constant A_0. The problem is then formulated as

$$IP: \quad \min_{\Gamma \in N} L(\Gamma) \ , \tag{27}$$

and the projected gradient is simply

$$\nabla_N L = \nabla L - \frac{\langle \nabla L, \nabla A \rangle_\Gamma}{||\nabla A||^2_\Gamma} \nabla A = \kappa - \frac{\langle \kappa, 1 \rangle_\Gamma}{||1||^2_\Gamma} 1 \ .$$

Remark: For a simple closed curve in the plane it is well known that $\langle \kappa, 1 \rangle_\Gamma = 2\pi$, cf. [12–p.37]. Also, the norm in the denominator is equal to the curve length, $||1||^2_\Gamma = \int_\Gamma d\sigma = L(\Gamma)$. Using Schwarz' inequality (12) it is clear that the constraints $L(\Gamma) = L_0$ and $A(\Gamma) = A_0$ always have nonzero gradient because $2\pi = \langle \kappa, 1 \rangle_\Gamma \leq ||\kappa||_\Gamma ||1||_\Gamma = ||\nabla L||_\Gamma ||\nabla A||_\Gamma$.

Using the projected gradient above, we can now state and prove the following classical result, which tells us what the extremals are.

Proposition 2. *For a simple closed curve in the plane*

$$\nabla_N L(\Gamma) = 0 \quad \Leftrightarrow \quad \Gamma = \text{circle} \ .$$

Proof. Recall that a circle with radius r has positive constant curvature $\kappa = 1/r$. If Γ is a circle with radius r, then

$$\nabla_N L = \kappa - \frac{\langle \kappa, 1 \rangle_\Gamma}{\langle 1, 1 \rangle_\Gamma} = \frac{1}{r} - \frac{1}{r} \frac{\langle 1, 1 \rangle_\Gamma}{\langle 1, 1 \rangle_\Gamma} = 0 \ .$$

Conversely, if $\nabla_N L = 0$, then

$$\nabla_N L = \kappa - \frac{2\pi}{L(\Gamma)} = 0 \ ,$$

which implies $\kappa = 2\pi / L(\Gamma) = 1/r > 0$, so Γ is a circle.

Figure 2 illustrates two examples of the projected gradient descent

$$\frac{\partial \phi}{\partial t} = (\kappa - \frac{2\pi}{L(\Gamma)}) |\nabla \phi| \ , \tag{28}$$

for the problem (27).

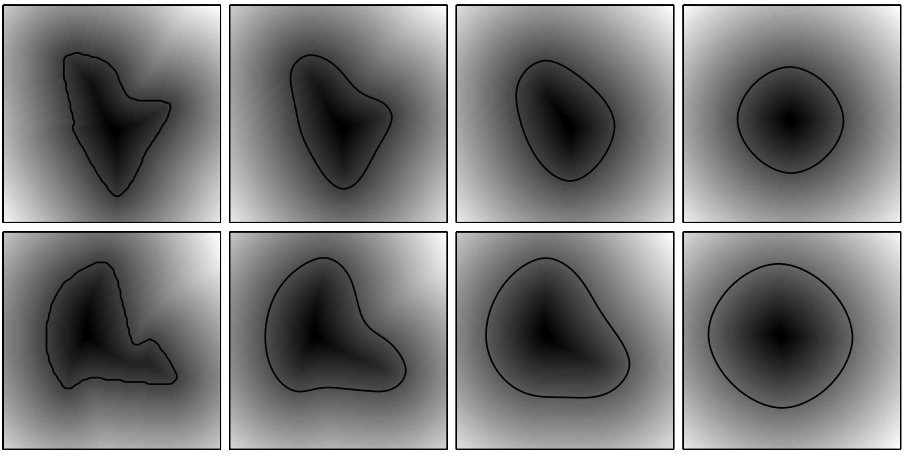


Fig. 2. Curve evolution for projected gradient descent of the isoperimetric problem. From left to right: initial shape, after 100 iterations, after 400 iterations and final shape for two randomly created planar curves. The value of the function ϕ is indicated in grayscale.

4.2 Practical Example: Visibility

This example concerns evolving surfaces in a surface fitting scheme such that they never violate visibility constraints cf. [4]. Some form of 3D data, e.g. unorganized points, is recovered from a sequence of images e.g. using structure from motion [13]. The goal is then to fit a surface to this data as a part of the scene reconstruction procedure. To do this the method from [14] can be used. In this case the functional is

$$F(\Gamma) = \int_{\Gamma} d(\mathbf{x})\, d\sigma \ , \tag{29}$$

where $d(\mathbf{x}) : \mathbf{R}^{m+1} \rightarrow \mathbf{R}$ is the distance potential to the data set. The gradient is $\nabla F = \nabla d \cdot \mathbf{n} + d\kappa$ from (16).

Let $W \subset \mathbf{R}^3$ denote the set corresponding to the forbidden regions in space determined from observations in the images. The following functional,

$$G(\Gamma) = \int_{\Omega} \chi_W \, d\mathbf{x} \ , \tag{30}$$

where Ω is the interior of Γ and χ_W is a characteristic function for the set W, was used in [4] to detect if the visibility condition is violated. The problem of evolving Γ such that no seen parts of the 3D data are occluded by the surface during evolution, leads to a variational problem of minimizing (29) under the constraint $G(\Gamma) = 0$.

The gradient of the constraint functional is $\nabla G = \chi_W$, by (16). This means that the projected gradient is

$$\nabla_N F = \nabla d \cdot \mathbf{n} + d\kappa - \frac{\langle \nabla d \cdot \mathbf{n} + d\kappa, \chi_W \rangle_{\Gamma}}{||\chi_W||^2_{\Gamma}} \chi_W \ ,$$

and the gradient descent evolution is

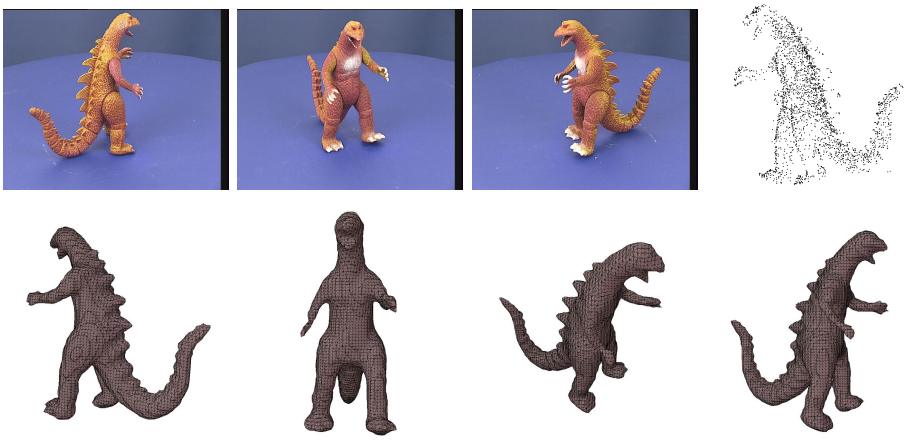


Fig. 3. Reconstruction from the Oxford dinosaur sequence. (top) Sample images recovered 3D points. (bottom) Four views of the reconstructed surface, using the visibility constraint.

$$\frac{\partial \phi}{\partial t} = (\nabla d \cdot \mathbf{n} + d\kappa - \frac{\langle \nabla d \cdot \mathbf{n} + d\kappa, \chi_W \rangle_\Gamma}{||\chi_W||^2_\Gamma} \chi_W) |\nabla \phi| \,.$$

An example[2] reconstruction is shown in Figure 3 where the forbidden set W is the complement of the visual hull.

5 Implementation Issues

This section deals with some practical considerations for implementing a projected gradient descent evolution, as described in Sections 3 and 4. There are some issues related to the fact that at the implementation stage we use finite resolution and numerical approximations.

When computing the projected gradient, one needs to compute surface (and volume) integrals such as the curve length $L(\Gamma)$ for the isoperimetric problem. This is not trivial to do in the implicit level set representation. Sometimes it is enough to compute these values with approximations using the Dirac and Heaviside functions as in e.g. [3,15]. If more accurate values are needed, methods like the marching cubes algorithm [16] can be used. Whatever the choice, there will be small errors in these computed values.

Another issue is that of satisfying the constraints. Only the gradient of the constraint functional appear in the evolution, not the value. In the isoperimetric example, the value of $A(\Gamma)$ never appear in the evolution equation (28). During the decent, finite step lengths and numerical errors in the calculations of the curve length $L(\Gamma)$ may introduce a "drift" in the value of $A(\Gamma)$, so that the constraint $G(\Gamma) = A(\Gamma) - A_0 = 0$ fails to hold after a while. One way to counter

[2] The example is from [4].

this drift is to add the gradient of a second order term, $G(\Gamma)^2/2$, to the projected gradient $\nabla_N F$, such that the evolution becomes $\dot{\Gamma}(t) = -v(\Gamma)$ with

$$v(\Gamma) = \nabla_N F + G\,\nabla G \ . \tag{31}$$

For the isoperimetric problem (27) this is

$$v(\Gamma) = \nabla_N L + (A(\Gamma) - A_0)1 \ . \tag{32}$$

The extra term in $v(\Gamma)$ is zero if the constraint $G(\Gamma) = 0$ holds, otherwise this term will try to restore the constraint.

Adding this term to the N-gradient of L does not change the extremals. In fact, the stationary points of the new augmented evolution are the same as for the one with $v(\Gamma) = \nabla_N L$.

Proposition 3. *$v(\Gamma) = 0$ in (32) if and only if $G(\Gamma) = A(\Gamma) - A_0 = 0$ and $\nabla_N L(\Gamma) = 0$.*

Proof. If $v(\Gamma) = 0$ then pairing with $\nabla G = 1$ yields

$$\begin{aligned} 0 = \langle v(\Gamma), 1\rangle_\Gamma &= \langle \kappa - \frac{2\pi}{L(\Gamma)} + (A(\Gamma) - A_0)1, 1\rangle_\Gamma \\ &= \langle \kappa - \frac{2\pi}{L(\Gamma)}, 1\rangle_\Gamma + (A(\Gamma) - A_0)\langle 1, 1\rangle_\Gamma = 0 + (A(\Gamma) - A_0)L(\Gamma) \ . \end{aligned}$$

Since $L(\Gamma) \neq 0$ this implies that $G(\Gamma) = A(\Gamma) - A_0 = 0$, and consequently $\nabla_N L(\Gamma) = 0$ by (32). In particular, Γ is a circle (by Proposition 2). The other direction is trivial.

This result is in fact true for the general case (31). The proof is a straight-forward adaption of the proof of Proposition 3 and is left to the reader.

6 Conclusions

In this paper we introduced a geometric infinite-dimensional gradient projection method for variational problems with constraints, as an extension of the finite-dimensional theory. Using a scalar product on the manifold of admissible m-surfaces we showed that gradient descent for the Lagrange method is equivalent to an orthogonal projection on the tangent space of the constraint manifold. We gave examples of how to use this theory in practice together with some useful ideas for implementation. This includes a way of stabilizing the evolution by augmenting the normal velocity. We also prove that this modified evolution still solves the original problem.

References

1. Solem, J.E., Overgaard, N.: A geometric formulation of gradient descent for variational problems with moving surfaces. In: The 5th International Conference on Scale Space and PDE methods in Computer Vision, Scale Space 2005, Hofgeismar, Germany, Springer (2005) 419–430

2. Damgaard Pedersen, U., Fogh Olsen, O., Holm Olsen, N.: A multiphase variational level set approach for modelling human embryos. In: IEEE Proc. Workshop on Variational, Geometric and Level Set Methods in Computer Vision. (2003) 25–32
3. Zhao, H., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. J. Computational Physics 127 (1996) 179–195
4. Solem, J.E., Kahl, F., Heyden, A.: Visibility constrained surface evolution. In: International Conference on Computer Vision and Pattern Recognition, San Diego, CA. (2005)
5. Rosen, J.: The gradient projection method for nonlinear programming: Part II, nonlinear constraints. J. Society for Industrial and Applied Mathematics **9** (1961) 514–532
6. Thorpe, J.A.: Elementary Topics in Differential Geometry. Springer-Verlag (1985)
7. Dervieux, A., Thomasset, F.: A finite element method for the simulation of Rayleigh–Taylor instability. In Rautman, R., ed.: Approximation Methods for Navier–Stokes Problems. Volume 771 of Lecture Notes in Mathematics. Springer, Berlin (1979) 145–158
8. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics **79** (1988) 12–49
9. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. Journal of Computer Vision (1997)
10. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces based object segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 394–398
11. Paragios, N., Deriche, R.: Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. International Journal of Visual Communication and Image Representation (2000)
12. do Carmo, M.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)
13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
14. Zhao, H., Osher, S., Merriman, B., Kang, M.: Implicit and non-parametric shape reconstruction from unorganized points using a variational level set method. In: Computer Vision and Image Understanding. (2000) 295–319
15. Osher, S.J., Fedkiw, R.P.: Level Set Methods and Dynamic Implicit Surfaces. Springer Verlag (2002)
16. Lorensen, W., Cline, H.: Marching cubes: a high resolution 3d surface reconstruction algorithm. Computer Graphics (Siggraph'87) **21** (1987) 163–169

Regularization of Mappings Between Implicit Manifolds of Arbitrary Dimension and Codimension

David Shafrir[1], Nir A. Sochen[1], and Rachid Deriche[2]

[1] University of Tel-Aviv, Ramat-Aviv, Tel-Aviv 69978, Israel
{davidsha, sochen}@post.tau.ac.il
[2] Odysee project, INRIA Sophia-Antipolis, 2004 Route des Lucioles, 06902 Sophia-Antipolis CEDEX, France
Rachid.Deriche@sophia.inria.fr

Abstract. We study in this paper the problem of regularization of mappings between manifolds of arbitrary dimension and codimension using variational methods. This is of interest in various applications such as diffusion tensor imaging and EEG processing on the cortex. We consider the cases where the source and target manifold are represented implicitly, using multiple level set functions, or explicitly, as functions of the spatial coordinates. We derive the general implicit differential operators, and show how they can be used to generalize previous results concerning the Beltrami flow and other similar flows.

As examples, We show how these results can be used to regularize gray level and color images on manifolds, and to regularize tangent vector fields and direction fields on manifolds.

1 Introduction

In many application the data of interest is defined on a manifold. Examples for this exist in areas like medical imaging, computer graphics and computational physics. In medical imaging, the data may be attached to a tissue, as in the case of cortex data processing. In computer graphics it appears in texture smoothing and synthesis, and in surface parameterization.

In much of the cases the data is noisy and needs to be regularized. Therefore there is a need to perform the regularization intrinsically on the manifold. The use of analytical methods, such as calculus of variations, and of efficient numerical schemes, such as the level-set technique, give rise to efficient and robust solutions for this task.

Let us consider first the more standard flat case. One of the main approaches for regularization is based on p-harmonic maps [1], in which the smoothness criterion of the data is based on the L_p norm of the data gradients.

The two most popular variants of harmonic maps are isotropic diffusion ($p = 2$) and Total Variation ($p = 1$). The isotropic diffusion, which has been studied and used extensively, proves to have major drawbacks when dealing with data

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 344–355, 2005.

that has discontinuities. As it is essentially linear, it is not edge preserving and blurs the data. The Total Variation denoising, first introduced by Rudin et al. [2], does preserve edges, but causes the infamous stair-casing affect and loss of texture.

Another problem regarding p-harmonic maps (and their generalized ϕ formulation [3]) is its extension to vector valued data. The problem is that the diffusion is done separately in each channel, and there is no coupling between the different channels. This may lead to loss of correlation between the channels. For example, in color images, chromaticity is not well preserved and color artifacts are observed [4].

The second approach for denoising, the Beltrami framework [5,6,7] is based on looking at the mapping to the data as a manifold in the spatial-feature space. The smoothing process is done by the evolution of that manifold under the volume minimizing flow. The edge-preserving property comes from the fact that only the feature space coordinates are modified. The Beltrami flow has been shown to interpolate (for scalar valued data) between the Total Variation diffusion and the isotropic diffusion. This allows more control of the tradeoff between the benefits and shortcomings of the two methods. Also, by representing a vector-valued mapping as single geometrical object, this method creates a strong coupling between the different channels, and therefore seems to be more natural for vector valued data.

1.1 The Extension to Non-flat Manifolds

The implicit level-set representation of manifolds has become a viable alternative to triangulated surfaces for solving variational problems [8,9,10]. It is much more natural for numerical computations, since the numerical behavior can be analyzed using standard methods to guarantee accuracy and robustness. The implementation is also much simpler and elegant, and there is no need to address the problems that arise due to the irregular structure of triangulated meshes. Harmonic maps have been recently extended for denoising data on implicit manifolds [10,11] defined by single level-set function (codimension 1).

Motivated by the need to denoise color images on surfaces, the case where the data is constrained to be on sphere was also studied from a level-set perspective in [12,13]. It was later extended to the case where the data is constrained to be on a more general target manifold [14]. While it has impressive results it still suffers from the same problems that are encountered in the flat domain situations: Either the over smoothness in the isotropic diffusion case, or the stair-case affect in the Total Variation based diffusion. The Beltrami framework has been formulated via an explicit parameterizations of the domain and target manifolds. Non-flat target manifolds were studied by this approach in [15,16]. It was mostly studied for flat domain manifolds, and only recently was studied for gray-level images on implicit surfaces [17].

In this paper, we explore the geometry of multiple level set functions, and show how it can be used to generalize pervious results to their full extent. We study the general problem of regularization of vector-valued data on a manifold

of arbitrary dimension and codimension, defined implicitly or explicitly, as a function of the spatial coordinates. This extension is of particular interest in the Beltrami case because of its advantages in higher dimensional data. We also study the case where the data is constrained to be on a manifold of arbitrary dimension and codimension. By this we show the general approach for solving the problem of regularization of mappings between general manifolds, represented implicitly or explicitly. We show, as examples, how the implicit Beltrami flow can be used to regularize gray level and color images over manifolds, and to regularize tangent and directional vector fields on manifolds.

2 The Implicit Beltrami Flow

We first consider the case of the Beltrami flow, from a variational approach. Let M be a $n-k$ dimensional manifold embedded in $\mathbb{R}^n$, represented by the intersection the zero level-sets of k implicit functions (constraints). Let $(f) : \mathbb{R}^n \to \mathbb{R}^m$ be the smooth extension to all of $\mathbb{R}^n$ of the mapping between the domain manifold to the vector data values. We look at the graph of the mapping, embedded in $\mathbb{R}^{m+n}$, and we describe it by an additional set of m constraints, describing the mapping. For example, the mapping $y = g(\boldsymbol{x})$ is transformed into the constraint $y - g(\boldsymbol{x}) = 0$. Thus the graph is described by a set of $k+m$ constraints. The regularization is performed by changing the mapping so that the graph manifold evolves under the volume minimization flow.

Let us consider the volume minimization flow from an implicit approach: Given a set of constraints, we wish to find the variation of the volume of the solution manifold with respect to the constraints functions. For start, we need to express the volume of the solution as a function of the set of constraints. We have the following theorem:

Proposition 1. *Let $\{f_i\}_{i=1}^k$ be k functions from $\mathbb{R}^n$ to $\mathbb{R}$, and let G be a $n \times k$ matrix with the i'th column equal to ∇f_i, and assume that it is of maximal rank. Let S be the intersection manifold of the function's zero level sets:*

$$S = \{\overrightarrow{x} | f_i(\overrightarrow{x}) = 0 : i = 1, \ldots, k\} \ . \tag{1}$$

Then the volume of S is equal to:

$$Vol(S) = \int_{\mathbb{R}^n} \prod_i \delta(f_i(\boldsymbol{x})) \sqrt{\det(G^T G)}\, d\boldsymbol{x} \ . \tag{2}$$

Where δ is Dirac's delta function.

The matrix G is simply the Jacobian matrix of the transformation defined by the k implicit functions, and this is how we refer to it.

In the next sections we will make use of the projection operator. Let G be a $n \times k$ matrix. The projection operator P_G to the orthogonal complement of the column space of G is equal to:

$$P_G = I - G(G^T G)^{-1} G^T \ . \tag{3}$$

The reason this operator is useful in our context is because of the following fact: Let $f_1, \ldots, f_k$ are k scalar functions in $\mathbb{R}^n$, and let G be the corresponding Jacobian matrix at point x. Then the projection operator P_G projects a vector v onto the intersection manifold of the level sets of the functions $f_1, \ldots, f_k$ that passes through $\boldsymbol{x}$. This follows immediately from the fact the gradient of the function at point $\boldsymbol{x}$ is orthogonal to the level set of the function that passes through that point. Using this observation and some additional computational effort we derive in the appendix the gradient descent equation for the volume functional: Let G be the gradient matrix of $f_1, \ldots, f_k$, and let G' be the matrix G without the row ∇f_i. Then

$$\frac{df_i}{dt} = \frac{1}{\sqrt{\det(G^T\, G)}} \nabla \cdot \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla f_i}{\|P_{G'} \nabla f_i\|^2} \right) \ . \tag{4}$$

Let the manifold M, embedded in $\mathbb{R}^n$, be the intersection of the zero level sets of the functions $\psi_1, \ldots, \psi_k$, and let $u_1, \ldots, u_m$ be the feature functions. Let $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ be the coordinate system of the combined spatial-feature space.

The graph of the mappings is described by the following set of implicit functions:

$$\begin{aligned}
& f_1 = y_1 - \beta\, u_1(x_1, \ldots, x_n) = 0 \\
& \vdots \\
& f_m = y_m - \beta\, u_m(x_1, \ldots, x_n) = 0 \\
& \psi_1(x_1, \ldots, x_n) = 0 \\
& \vdots \\
& \psi_k(x_1, \ldots, x_n) = 0 \ .
\end{aligned}$$

Where β is a parameter for the embedding. Let G be the Jacobian matrix of the above implicit functions, and G' be G without the column ∇f_i. We can use equation (4) to derive the gradient descent equation of u_i:

$$\frac{du_i}{dt} = \frac{1}{\sqrt{\det(G^T\, G)}} \nabla \cdot \left(\frac{\sqrt{\det(G'^T\, G')}\, P_{G'} \nabla f_i}{-\beta \|P_{G'} \nabla f_i\|} \right) \ . \tag{5}$$

3 The Geometry of Multiple Level-Set Functions

In this section we will show a different derivation of the flow equations by deriving the implicit differential operators in the situation of multiple level set functions.

We look at a set of k implicit functions $f_1, \ldots, f_k$ in $\mathbb{R}^n$. The k functions define a transformation $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^k$, and let G the Jacobian of this transformation. We assume that G is of maximal rank. Let p be a point in $\mathbb{R}^n$ and let S be the intersection of level sets that passes through it:

$$S = \{\boldsymbol{x} |\ f_i(\boldsymbol{x}) = f_i(p) \ \ i = 1, \ldots, k\} \ . \tag{6}$$

The tangent space $T_p\mathbb{R}^n$ at p can be decomposed into 2 linear subspaces: The tangent space T_pS of S at p and its orthogonal complement $T_p^{\perp}S$. The differential transformation df restricted to $T_p^{\perp}S$ is one to one, so the transformation $\mathcal{F}$ induces a metric on $T_p^{\perp}S$ which is the pullback of the Euclidean metric by $\mathcal{F}$. We consider the case where the metric on T_pS is simply the Euclidean metric. The combination of the two restrictions defines a metric g on $\mathbb{R}^n$.

This is the natural metric induced by the choice of the implicit functions, In which the level sets of the functions looks locally equally spaced and orthogonal. In this metric the length of the tangent vector ds is:

$$\|ds\|^2 = \|P_G(ds)\|^2 + \sum_i (df_i(ds))^2 = \|P_G(ds)\|^2 + \sum_i (\nabla f_i \cdot ds)^2 \ . \quad (7)$$

The volume element of this metric is the the volume element of the transformation $\mathcal{F}$ restricted to $T_p^{\perp}S$, which is $g = \sqrt{\det(G^T G)}$.

It is interesting to note that this metric coincides with the usual Euclidean metric when the functions are distance fields and the level sets of the different functions are orthogonal at every point. In this case G is an orthogonal matrix. Since we are interested in the restriction of this metric to S, vectors on the tangent space should be first projected to $T_p^{\perp}S$. This metric can be used to compute the implicit differential operators on S.

3.1 The Intrinsic Differential Operators

The intrinsic gradient is computed by simply projecting it on S. Since the metric on the tangent space is Euclidean, the gradient is simply

$$\nabla_g f = P_G \nabla f \ . \quad (8)$$

The divergence operator with respect to this metric is:

$$\nabla_g \cdot (V) = \frac{1}{\sqrt{g}} \nabla \cdot (\sqrt{g} P_G V) \ . \quad (9)$$

We combine the last 2 results to calculate the Laplace-Beltrami operator of a scalar function h:

$$\Delta_g h = \frac{1}{\sqrt{g}} \nabla \cdot (\sqrt{g} P_G \nabla h) \ . \quad (10)$$

The Laplace-Beltrami operator can be used to derive the Beltrami flow. We will use the same notation as defined in (5).

The Beltrami flow for the function u_i is simply the Laplace-Beltrami operator of this function on the graph manifold [5]. We extend the function $u_i(x_1, \ldots, u_n)$ to be a function of R^{n+m} by the equation $u_i = -\frac{y_i}{\beta}$.

$$\frac{du_i}{dt} = \Delta_g u_i = \frac{1}{\sqrt{g}} \nabla \cdot (\sqrt{g} P_G \nabla u_i) = \frac{1}{-\beta \sqrt{g}} \nabla \cdot (\sqrt{g} P_G e_i) \ . \quad (11)$$

Where $(e_1, \ldots, e_m)$ is the last m vectors in natural basis of $\mathbb{R}^{n+m}$, which span the feature subspace. It is easy to show that this equation is equivalent to (5).

We can use the implicit operators to extend any other flow equation to the case of multiple level set functions. For example, in the case of p-Harmonic maps, Let G be the Jacobian matrix of the implicit functions defining the manifold. Then:

$$v_t = \nabla_g \cdot \left(\frac{\nabla_g v}{\|\nabla_g v\|^{2-p}}\right) = \frac{1}{\sqrt{g}} \nabla \cdot \left(\frac{\sqrt{g}\, P_G \nabla v}{\|P_G \nabla v\|^{2-p}}\right) \quad . \tag{12}$$

This is a generalization of the single level-set function case in [10].

3.2 Non-flat Color Images Regularization

We consider smoothing a color image defined on a surface. The domain manifold embedded in $\mathbb{R}^3$ is specified by an implicit function $\psi(x, y, z) = 0$, and let (r, g, b) be the coordinate system of the feature space. Let u_1, u_2, u_3 be the 3 function of the red, green and blue channels. Their embedding in $\mathbb{R}^6$ is:

$$\begin{aligned} r &= \beta\, u_1(x, y, z) \\ g &= \beta\, u_2(x, y, z) \\ b &= \beta\, u_3(x, y, z) \quad . \end{aligned}$$

Therefore the set of constraints describing the graph is:

$$\begin{aligned} f_1 &= r - \beta\, u_1(x, y, z) = 0 \\ f_2 &= g - \beta\, u_2(x, y, z) = 0 \\ f_3 &= b - \beta\, u_3(x, y, z) = 0 \\ f_4 &= \quad \psi(x, y, z) \quad = 0 \quad . \end{aligned}$$

We can now use (5) to calculate the variation of the graph surface are with respect to u_1, u_2 and u_3. Let G be the Jacobian matrix corresponding to the functions $f_1, \ldots, f_4$

$$G = \begin{pmatrix} -\beta \nabla u_1 & 1 & 0 & 0 \\ -\beta \nabla u_2 & 0 & 1 & 0 \\ -\beta \nabla u_3 & 0 & 0 & 1 \\ \nabla \psi & 0 & 0 & 0 \end{pmatrix}^T$$

where ∇ is the gradient operator with respect to the spatial-coordinates (x, y, z). The G' matrix is the matrix G without the column ∇f_i. Using (5) we get the gradient descent equation:

$$\frac{du_i}{dt} = \frac{1}{\sqrt{g}} \nabla \cdot \left(\frac{\sqrt{g}\, P_{G'} \nabla u_i}{\|P_{G'} \nabla f_i\|^2}\right) \quad . \tag{13}$$

Where the divergence is on the spatial coordinates only. The well-posedeness and extremum principle for this type of flow are discussed in [18,19].

4 Constrained Regularization

In constrained regularization, the feature space has a shape of a general manifold, which can depend or not depend on the spatial coordinates. Just like the domain manifold, we represent it implicitly by a set of implicit functions $h_1, \ldots, h_l$. Explicit constraints of the form $w = f(\boldsymbol{x})$ can be transformed to the implicit constraint by $h(\boldsymbol{x}, w) = w - f(\boldsymbol{x}) = 0$.

We calculate the gradient descent equation using the following two steps process: We first calculate the unconstrained variation, and then enforce the constraints by projecting it to the tangent space of the constraints manifold (which can be point dependant). This method is based on the gradient projection method due to Rosen [20]. It has been also been used to some extent in [14]. The projection is performed by calculating the gradients matrix C of the functions $h_1, \ldots, h_l$ and using projection operator P_C of (3). Alternatively, a series of projections can be used, as explained in Sect. 5.

4.1 Explicit/Implicit Constraints

Although we transform both explicit and implicit constraints into implicit functions, there is a difference in the actual implementation of the two cases. Let the feature space be of dimension m and let $h_1, \ldots, h_l$ be the constraints. If the constraints are given explicitly, It is possible to hold only $m - l$ of the coordinates, and calculate the other from them. The projected gradient descent step can be used to update these coordinates only, and by this we ensure that the data values are always on the constraining manifold.

A specific coordinate system may not cover the whole feature space in a numerically stable way, so there might be a need to switch coordinate systems for different regions [7,16]. When the constraints are given implicitly, all the coordinate must be kept. Due to numerical errors, the coordinates values may be taken out of the manifold during the evolution. Therefore there is a need to project the values back onto the manifold after each iteration, and this can be done using the Newton-Raphson method.

4.2 Regularization of Tangent Vector Fields

We consider regularization of tangent vector fields as an example to constrained regularization (see [10,21] for other approaches). Let M be a manifold embedded in $\mathbb{R}^n$, and let $\nu(x)$ be a tangent vector field of M. We look at $\nu(x)$ as an n dimensional vector field, and therefore the graph of the mapping between the manifold to the vector field can be embedded in $\mathbb{R}^{2n}$. We can use, as before, the Beltrami flow to smooth that graph, but with an additional constraint - we should keep the vector field tangent to the manifold.

As an example, we consider a vector filed on a surface in $\mathbb{R}^3$, which is described by the implicit function $\psi(x, y, z)$. The vector field coordinates (u, v, w) are specified using the functions $\nu_1(x, y, z), \nu_2(x, y, z), \nu_3(x, y, z)$ which are the extension of ν to all of $\mathbb{R}^3$. The gradient descent step $\tilde{u_t}$ of the unconstrained

version is exactly the same as in the case of color images. The gradient of the single constraint on the feature space is simply $\nabla\psi(x,y,z)$, as it is normal to the tangent plane in the point (x,y,z). Therefore the gradient descent equation is:

$$u_t = P_{\nabla\psi}\tilde{u_t} = \tilde{u_t} - \frac{\tilde{u_t}\cdot\nabla\psi}{\|\nabla\psi\|^2} \ . \tag{14}$$

For direction fields, the vector is constrained to be of a unit norm, and therefore there is an additional constraint:

$$\nu_1(x,y,z)^2 + \nu_2(x,y,z)^2 + \nu_3(x,y,z)^2 = 1 \ . \tag{15}$$

The feature space constraints matrix is:

$$C = \begin{pmatrix} \nabla\psi \\ 2\,\nu_1 \ 2\,\nu_2 \ 2\,\nu_3 \end{pmatrix}^T$$

And the gradient descent equation is therefore:

$$u_t = P_C(\tilde{u_t}) \ . \tag{16}$$

Where $\tilde{u_t}$ is the unconstrained gradient descent step.

5 Implementation Details

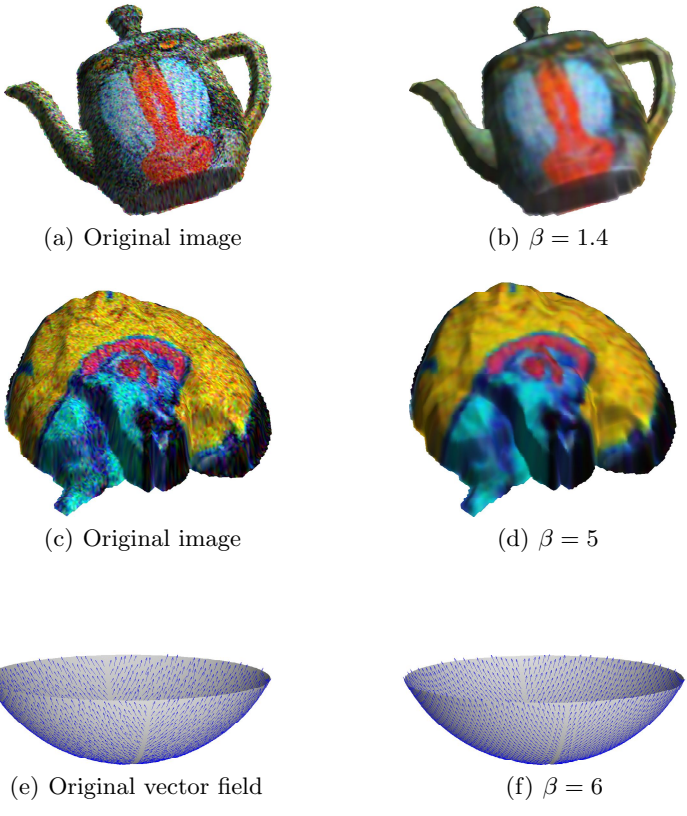
In order to reduce time and space complexity when working with manifolds of high dimension and codimension, the local level set method [22,23] can be used. This way solving PDE's on manifolds of high codimension becomes traceable. It is important to notice, however, that in the case of the implicit Beltrami flow, although the graph manifold in embedded in R^{m+n}, the actual computations are performed on the original n-dimensional space.

When working in higher dimensions, calculating the projection operator and the determinants can be quite cumbersome. There is however a way to calculate both these values in an iterative way. We use the Gram-Shmidt process, and we utilize the following two properties of it:

Lemma 1. *Let $u_1,\ldots,u_k$ be k vectors in $\mathbb{R}^n$ $(n \geq k)$ and let $v_1,\ldots,v_k$ be the k vectors that are the result of applying the Gram-Shmidt process on $u_1,\ldots,u_k$. We mark by V the matrix whose columns are $v_1,\ldots,v_k$ and by U the matrix whose columns are $u_1,\ldots,u_k$. Then:*

1. $\det(U^TU) = \det(V^TV) = \prod_i(\|v_i\|)$.
2. $v_i = P_L(u_i)$, *where L is the matrix whose columns are $u_1,\ldots,u_{i-1}$.*

We can use these facts to calculate $\sqrt{\det(G^T\,G)}$ and $P_{G'}(u_i)$ by ordering the vectors $\{u_1,\ldots,u_k\}$ so that u_i is last. Using The result of the Gram- Schmidt process, v_k is the needed projected vector, and the determinant can be computed by multiplying the lengths of the resultant vectors.



(a) Original image (b) $\beta = 1.4$

(c) Original image (d) $\beta = 5$

(e) Original vector field (f) $\beta = 6$

Fig. 1. Color images and vector fields on surfaces

6 Experimental Results

We implemented the color image and vector field regularization flows in Matlab. For color images on a surface, we created a three-dimensional level-set matrix for the surface, and three matrices to hold the red, green and blue channels. We tested the regularization for various values of β. In Fig. 1 we see the baboon image projected on the teapot, to which we added Gaussian noise. We used a low β value - $\beta = 0.5$ for the smoothing process, and we can see that the resultant image in Fig. 1 is somewhat over smoothed. This is expected since for $\beta \to 0$, (5) behaves like isotropic diffusion.

In Fig. 1 we added Gaussian noise to the brain image, and used a higher β value of 6 for the smoothing. We can see in Fig. 1 that the edges are preserved. This stems from the fact that in the case of scalar functions, the Beltrami flow behaves like the Total Variation flow when $\beta \to \infty$. For vector data, the Beltrami flow aligns the channels gradients so that edges are preserved.

Lastly, we created a piecewise smooth vector field on the half sphere. It has a marked discontinuity in the middle, where the direction of the vectors

jumps sharply. We added Gaussian noise to the vector field, and used $\beta = 6$ for the regularization. We can see in Fig. 1 that indeed the vector field is nicely regularized while the discontinuity is preserved.

7 Summary and Conclusions

In this article we derived the Beltrami flow from an implicit approach, where we modeled the domain manifold, the mapping and the target manifold using a set of constraints. This enabled us to extend the Beltrami framework to implicit manifolds that are given as an intersection of multiple level-sets. The resultant framework turns out to be very simple and easy to implement. It also enables to mix explicit and implicit constraints in a straightforward way.

Among the advantages of the Beltrami flow are its natural extension to vector valued data, and the ability to control the edge-preserving behavior tuning the β parameter. We have also shown how to deal with the case of arbitrary codimension of the domain and target manifolds, and this general technique can also be used in harmonic maps and related flows. Future research directions include applications to medical data processing and computer graphics, comparing the Beltrami approach to the more standard ones.

References

1. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press (January 2001)
2. Rudin, L., S.Osher, E.Fatemi: Nonlinear total variation based noise removal algorithms. Physica D **60** (1992) 259–268
3. Deriche, R., Faugeras, O.: Les edp en traitement des images et vision par ordinateur. Traitement du Signal **13** (1996)
4. Tschumperlé, D.: PDE's Based Regularization of Multivalued Images and Applications. PhD thesis, Universite de Nice-Sophia Antipolis (2002)
5. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Trans. on Image Processing **7** (1998) 310–318
6. Kimmel, R., Malladi, R., Sochen, N.: Images as embedded maps and minimal surfaces: Movies, color, texture, and volumetric medical images. Int. J. Comput. Vision **39** (2000) 111–129
7. Kimmel, R., Sochen, N.A.: Using beltrami framework for orientation diffusion in image processing. In: IWVF-4: Proceedings of the 4th International Workshop on Visual Form, Springer-Verlag (2001) 346–355
8. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging,Vision,and Graphics. Springer-Verlag New York, Inc. (2003)
9. Osher, S., Fedkiw, R.P.: Level set methods: an overview and some recent results. J. Comput. Phys. **169** (2001) 463–502
10. Bertalmio, M., Sapiro, G., Cheng, L.T., Osher, S.: Variational problems and pde's on implicit surfaces. In: VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01), IEEE Computer Society (2001) 186
11. Chan, T., Shen, J.: Variational restoration of nonflat image features: models and algorithms. SIAMJ Appl. Math. **61** (2000) 1338–1361

12. Tang, B., Sapiro, G., Caselles, V.: Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case. Int. J. Comput. Vision **36** (2000) 149–161
13. Brezis, H., Coron, J.M., Lieb, E.H.: Harmonic maps with defects. Commun. Math. Phys. **107** (1986) 649–705
14. Memoli, F., Sapiro, G., Osher, S.: Solving variational problems and partial differential equations mapping into general target manifolds. J. Comput. Phys. **195** (2004) 263–292
15. Sochen, N., Zeevi, Y.Y.: Representation of colored images by manifolds embedded in higher dimensional non-euclidean space. In: ICIP '98: Proceedings of the International Conference on Image Processing. (1998)
16. Kimmel, R., Sochen, N.A.: Orientation diffusion or how to comb a porcupine? Journal of Visual Communication and Image Representation **13** (2001) 238–248
17. Sochen, N., Deriche, R., Perez, L.L.: The beltrami flow over implicit manifolds. In: ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision, IEEE Computer Society (2003) 832
18. Dascal, L., Sochen, N.: On the maximum principle of the color beltrami flow. SIAM Journal on Applied Mathematics **65** (2005) 1615–1632
19. Dascal, L., Kamin, S., Sochen, N.: Variational inequality for discontinuous solutions of degenerated parabolic equations. Technical Report **School of Mathematical Sciences, Tel-Aviv University** (2005)
20. Rosen: Two-phase algorithm for non linear constraint problems. Non Linear Programming **3** (1978) 97–124
21. Chefd'hotel, C., Tschumperlé, D., Deriche, R., Faugeras, O.: Regularizing flows for constrained matrix-valued images. J. Math. Imaging Vis. **20** (2004) 147–162
22. Min, C.: Local level set method in high dimension and codimension. J. Comput. Phys. **200** (2004) 368–382
23. Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A pde-based fast local level set method. J. Comput. Phys. **155** (1999) 410–438

Appendix

We will need the following lemma for the volume variation calculation:

Lemma 2. *Let G be a $n \times k$ matrix, with the vector $(v) = (x_1, \ldots, x_n)$ of variables in the first column and the other columns constant. Let $g = \sqrt{\det(G^T G)}$. Let G' be the matrix G without the vector (v). Then*

$$\nabla g = \frac{\sqrt{\det(G^T G)}\, P_{G'}(v)}{\|P_{G'}(v)\|^2} \quad . \tag{17}$$

Where $P_{G'}$ is the projection operator to the orthogonal complement of the space spanned by the vectors of G'.

Let $u_1, \ldots, u_k, v$ be $k+1$ implicit function in R^n, and let G be the Jacobian matrix of these functions. The volume of their zero level sets intersection is:

$$Vol(S) = \int_{\mathbb{R}^n} \Pi_i \delta(u_i(\boldsymbol{x}))\, \delta(v)\, \sqrt{\det(G^T G)}\, d\boldsymbol{x} \tag{18}$$

Our goal is to calculate the variation of the volume functional with respect to the function v. Let G' be the matrix G without the column ∇v. Let S' be the intersection manifold of the function $u_1, \ldots, u_k$, and $P_{G'}$ be the projection operator to it.

$$\frac{d}{dt} Vol(S)(v + t\,\mu) = \int_{\mathbb{R}^n} \delta'(v)\, \Pi_i \delta(u_i) \sqrt{\det(G^T\, G)}\, \mu \tag{19}$$

$$+ \int_{\mathbb{R}^n} \Pi_i \delta(u_i) \delta(v) \frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2}\, \nabla \mu \tag{20}$$

$$= \ldots - \int_{\mathbb{R}^n} \nabla \cdot \left(\Pi_i \delta(u_i) \delta(v) \frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \right) \mu \tag{21}$$

$$= \int_{\mathbb{R}^n} \delta'(v) \Pi_i \delta(u_i) \sqrt{\det(G^T\, G)}\, \mu \tag{22}$$

$$- \int_{\mathbb{R}^n} \nabla \cdot \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \Pi_i \delta(u_i) \right) \delta(v)\, \mu \tag{23}$$

$$- \int_{\mathbb{R}^n} \Pi_i \delta(u_i) \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \right) \cdot \nabla v\, \delta'(v)\, \mu \tag{24}$$

$$= - \int_{\mathbb{R}^n} \nabla \cdot \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \Pi_i \delta(u_i) \right) \delta(v)\, \mu \tag{25}$$

Since $\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \cdot \nabla v = \sqrt{\det(G^T\, G)}$.

$$= - \int_{\mathbb{R}^n} \nabla \cdot \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \Pi_{i>1} \delta(u_i) \right) \delta(u_1) \delta(v)\, \mu \tag{26}$$

$$- \int_{\mathbb{R}^n} \frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v \cdot \nabla u_1}{\|P_{G'} \nabla v\|^2} \Pi_{i>1} \delta(u_i) \delta'(u_1) \delta(v)\, \mu \quad . \tag{27}$$

But $P_{G'} \nabla v \cdot \nabla u_i = 0$. Thus, repeating the same argument

$$= - \int_{\mathbb{R}^n} \nabla \cdot \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \right) \Pi_i \delta(u_i) \delta(v)\, \mu \tag{28}$$

$$= - \int_S \frac{1}{\sqrt{\det(G^T\, G)}} \nabla \cdot \left(\frac{\sqrt{\det(G^T\, G)}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|^2} \right) \mu \quad . \tag{29}$$

And this should hold for every μ. By naturally extending the variation to all of $\mathbb{R}^n$ we get the gradient descend equation for the Beltrami flow:

$$\frac{dv}{dt} = \frac{1}{\sqrt{\det(G^T\, G)}} \nabla \cdot \left(\frac{\sqrt{\det(G'^T\, G')}\, P_{G'} \nabla v}{\|P_{G'} \nabla v\|} \right) \quad . \tag{30}$$

Lens Distortion Calibration Using Level Sets

Moumen T. El-Melegy and Nagi H. Al-Ashwal

Electrical Engineering Department,
Assiut University, Assiut 71516, Egypt
moumen@aun.edu.eg

Abstract. This paper addresses the problem of calibrating camera lens distortion, which can be significant in medium to wide-angle lenses. Our approach is based on the analysis of distorted images of straight lines. We use a PDE-based level set method to find the lens distortion parameters that straighten these lines. One advantage of this method is that it integrates the extraction of image distorted lines and the computation of distortion parameters within one energy functional which is minimized during level set evolution. Some experiments to evaluate the performance of this approach are reported.

1 Introduction

Variational methods and partial differential equations (PDEs) are being more and more used to analyze, understand and exploit properties of images in order to design powerful application techniques, see for example [2, 3, 4]. Variational methods formulate an image processing or computer vision problem as an optimization problem depending on the unknown variables (which are functions) of the problem. When the optimization functional is differentiable, the calculus of variations provides a tool to find the extremum of the functional leading to a PDE whose steady state gives the solution of the imaging or vision problem. A very attractive property of these mathematical frameworks is to state well-posed problems to guarantee existence, uniqueness and regularity of solutions [3].

In this work we use a PDE-based level set method to calibrate camera lens distortion, which can be significant in medium to wide-angle lenses. Applications that require 3-D modeling of large scenes, (e.g., [8, 9, 10]) or image compositing over a large scene area, (e.g., [11, 12, 13]) typically use cameras with such wide fields of view. In such instances, the camera distortion effect has to be removed by calibrating the camera's lens distortion and subsequently undistorting the input image.

The lens distortion parameters are most often estimated along with all (extrinsic and intrinsic) parameters of the camera model (see for example [5,6]). This is done using a set of 3D-to-2D correspondences extracted with the help of a calibration object of known structure. The problem with these methods is the fact that there is some kind of coupling between internal parameters, including distortion parameters, and external parameters that result in high errors on the camera internal parameters [7]. Moreover obtaining accurate coordinates of 3D scene points is sometimes demanding or impossible (e.g. in case of snapshots already recorded).

N. Paragios et al. (Eds.): VLSM 2005, LNCS 3752, pp. 356 – 367, 2005.

In contrast, another family of nonmetric methods have been proposed, which do not rely on known scene points [7], [19], and [20]. Instead, these methods rely on the fact that straight lines in the scene must always perspectively project to straight lines in the image. This means that curvature of lines in the image is due to lens distortion. Using this principle, distortion parameters that map distorted image curves to straight lines can be estimated. Once estimated, the images can be undistorted by applying the inverse of the distortion function to the entire image or image features.

We will focus in this paper on recovering the radial component of lens distortion, as it is often the most prominent in images. We propose a variant of a level set formulation of the well-known Mumford-Shah functional in a way that permits the simultaneous segmentation of linear image objects and estimation of radial lens distortion coefficients in a calibration image. Our method thus belongs to the group of nonmetric calibration techniques. The computed coefficients can then be applied to undistort any images taken by the same camera. One key feature of our method is that it integrates the extraction of image features needed for calibration and the computation of distortion parameters within one energy functional, which is minimized during level set evolution. Thus our approach, unlike most other nonmetric calibration methods, avoids the propagation of errors in feature extraction onto the computation stage. This results in a more robust computation even at high noise levels.

The closest work to ours is that of Kang [14]. He used the traditional snake to calculate the radial lens distortion parameters. However, his method is sensitive to the location of the initial contour, so the user should specify the position of the initial contour. In contrast, our level-set based method has some global convergence property that makes it not sensitive to the initial level set.

The organization of this paper is as follows: In Section 2, we briefly review a level set formulation of the piecewise-constant Mumford-Shah functional, as proposed in [1]. In Section 3, we show the lens distortion model. Section 4 presents a variational framework to extract the linear objects and compute lens distortion coefficient from the image. We present the experimental results in Section 5. Our conclusions and future work are presented in Sections 6.

2 Region-Based Segmentation with Level Sets

In several papers [1, 15, 16], Chan and Vese detailed a level set implementation of the Mumford-Shah functional, which is based on the use of the Heaviside function as an indicator function for the separate phases.

The Chan-Vese method used a piecewise constant, region-based formulation of the functional, which allows the contour to converge to the final segmentation over fairly large distances, while local edge and corner information is well preserved. It can detect cognitive contours (which are not defined by gradients), and contours in noisy images. This model does not contain a balloon term, which induces a bias favoring either contraction or expansion; however it can detect objects of interest inside or outside the initial contour.

According to the level-set framework a contour, C, is embedded in a single level set function $\phi : \Omega \to \Re$ such that:

$$\begin{cases} C = \{(x, y) \in \Omega : \phi(x, y) = 0\}, \\ inside\,(C) = \{(x, y) \in \Omega : \phi(x, y) > 0\}, \\ outside(C) = \{(x, y) \in \Omega : \phi(x, y) < 0\}. \end{cases} \tag{1}$$

In the Mumford-shah model, a piecewise constant segmentation of an input image f is given by [1]:

$$\begin{aligned} E_{seg}(c_1, c_2, \phi, f) &= \mu \int_\Omega \left|\nabla H_\varepsilon(\phi)\right| dxdy + \nu \int_\Omega H_\varepsilon(\phi)\, dxdy \\ &+ \lambda_1 \int_\Omega \left|f - c_1\right|^2 H_\varepsilon(\phi)\, dxdy + \lambda_2 \int_\Omega \left|f - c_2\right|^2 (1 - H_\varepsilon(\phi))\, dxdy, \end{aligned} \tag{2}$$

where c_1 and c_2 are the mean values of the image f inside and outside the curve defined as a zero-level set of ϕ, respectively, and $\mu, \nu, \lambda_1, \lambda_2$ are regularizing parameters to be estimated or chosen *a priori*. H_ε is the regularized Heaviside function defined in [1] as

$$H_\varepsilon(s) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{s}{\varepsilon}\right)\right). \tag{3}$$

so

$$\delta_\varepsilon(s) = \frac{dH_\varepsilon}{ds} = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2 + s^2}. \tag{4}$$

The regularized H_ε and δ_ε having a discretization with a support larger than zero permits the detection of interior contours – for example if one wants to segment a ring-like structure, starting from an initial contour located outside the ring.

The Euler-Lagrange equation for this functional is implemented by the following gradient descent [1]:

$$\frac{\partial \phi}{\partial t} = \delta_\varepsilon(\phi)\left[\mu\, div\left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \nu - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2\right], \tag{5}$$

where the scalars c_1 and c_2 are updated with the level set evolution and given by:

$$c_1 = \frac{\int f(x, y) H(\phi)\, dxdy}{\int H(\phi)\, dxdy}. \tag{6}$$

$$c_2 = \frac{\int f(x, y)(1 - H(\phi))\, dxdy}{\int (1 - H(\phi))\, dxdy}. \tag{7}$$

3 Camera Distortion Model

The standard model for the radial and decentering distortion [5] is mapping from the observable, distorted image coordinates, (x, y), to the unobservable, undistorted image plan coordinates, (x^u, y^u), using the equation:

$$\begin{aligned} x^u &= x + \hat{x}(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \ldots) + \quad \left[\mathrm{P}_1(r^2 + 2\hat{x}^2) + 2\mathrm{P}_2 \hat{x}\hat{y}\right]\left[1 + \mathrm{P}_3 r^2 + \ldots\right], \\ y^u &= y + \hat{y}(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \ldots) + \quad \left[\mathrm{P}_1(r^2 + 2\hat{y}^2) + 2\mathrm{P}_2 \hat{x}\hat{y}\right]\left[1 + \mathrm{P}_3 r^2 + \ldots\right]. \end{aligned} \tag{8}$$

where

$\hat{x} = x - c_x$, $\hat{y} = y - c_y$, $r^2 = \hat{x}^2 + \hat{y}^2$, and $\kappa_1, \kappa_2, \kappa_3$ are the coefficients of radial distortion and P_1, P_2, P_3 are the coefficients of the decentering distortion. r $\psi$is the radius of an image point from the distortion center, defined as (c_x, c_y) above.

In this work we seek to recover, κ_1, as it has the most dominating effect. We assume the distortion center to be the image center whereas all other coefficients are neglected. So the distortion model becomes:

$$\begin{aligned} x^u &= x + \hat{x}(\kappa_1 r^2), \\ y^u &= y + \hat{y}(\kappa_1 r^2). \end{aligned} \tag{9}$$

4 Our Approach

Our goal here is to extend the energy functional (2) in order to force the level set to segment linear, or should-to-be-linear, objects from the image and simultaneously solve for the lens distortion parameters. Let us assume that an image f consists of multiple, disjoint objects each of them is contained in a level sets $\{\phi_i\}$. To realize this assumption, we run the evolution of the Chan-Vese level set formulation following (5-7). After a predefined number of iterations, the level set will be sufficiently close to the boundaries of the different objects in the image. This provides an initial separation of the image objects. At this point, the level set function ϕ is partitioned to L level set functions, $\{\phi_i\}$, where L is the number of initially detected objects in the image. The separation of the level set functions is done by applying connected components labeling to $H(\phi)$. Each separated level set function can be then reinitialized to a distance function [2]. Note that the overall level set can be given by $\phi = \max(\phi_1, \phi_2, \cdots, \phi_L)$. Similarly, the image f is partitioned into L images, f_i, each supposedly contains one object: $f_i = f \times H(\phi_i)$.

Having done this, the energy functional we then consider is

$$E = \sum_i E_{seg}(c_1^i, c_2^i, \phi_i, f_i) + \alpha E_{Line}(\rho_i, \theta_i, \phi_i), \tag{10}$$

where

$$E_{Line}(\rho_i,\theta_i,\phi_i)=\int_\Omega\left(\rho_i - x^u\cos\theta_i - y^u\sin\theta_i\right)^2 H(\phi_i)\,dxdy. \tag{11}$$

Here α is a nonnegative parameter controlling the emphasize on the second term E_{Line}, which measures how well a level set presents a line in the undistorted image coordinates (x^u, y^u), with θ_i being the orientation of the normal to the line, and ρ_i being the distance to the line from the origin. Note that the undistorted coordinates are related to the given distorted image coordinates $(x\ , y\ )$ via the distortion parameter κ_1 as in (9). Note also in this formulation, we allow a level set to have mean values inside (represented by c_1^i) and outside (represented by c_2^i) its zero-level contour different from the other level sets. As such, the method is able to segment objects of different intensities. This is in contrast to the original two-phase Chan-Vese formulation. Although multi-phase level sets [17] can achieve this, it is less convenient for our purpose and work in this paper.

To find the minimizer of (10), we start with finding closed-form solutions for (ρ_i,θ_i) of each line by setting $\frac{\partial E}{\partial\rho_i}=\alpha\frac{\partial E_{Line}(\rho_i,\theta_i,\phi_i)}{\partial\rho_i}=0$, and $\frac{\partial E}{\partial\theta_i}=\alpha\frac{\partial E_{Line}(\rho_i,\theta_i,\phi_i)}{\partial\theta_i}=0$, which leads to

$$\rho_i=\bar{x}\cos\theta_i+\bar{y}\sin\theta_i, \tag{12}$$

where

$$\bar{x}=\frac{\int_\Omega x^u\,H(\phi_u)dxdy}{\int_\Omega H(\phi_i)\,dxdy},\ \bar{y}=\frac{\int_\Omega y^u H(\phi_i)dxdy}{\int_\Omega H(\phi_i)dxdy}, \tag{13}$$

$$\theta_i=\frac{1}{2}\arctan\left(\frac{a_2}{a_1-a_3}\right), \tag{14}$$

with

$$a_1=\int_\Omega(x^u-\bar{x})^2 H(\phi_i)dxdy\,, \tag{15}$$

$$a_2=2\int_\Omega(x^u-\bar{x})(y^u-\bar{y})H(\phi_i)dxdy\,, \tag{16}$$

$$a_3=\int_\Omega(y^u-\bar{y})^2 H(\phi_i)dxdy\,. \tag{17}$$

Similarly, the scalars $\{c_1^i\}$ and $\{c_2^i\}$ are given by:

$$c_1^i = \frac{\int f_i(x,y)H(\phi_i)\,dxdy}{\int H(\phi_i)\,dxdy}, \tag{18}$$

$$c_2{}^i = \frac{\int f_i(x,y)(1-H(\phi_i))\,dxdy}{\int (1-H(\phi_i))\,dxdy}. \tag{19}$$

As for κ_1 that minimizes (10), we start with an initial guess κ_1^0 (in our implementation, we take it 0). Introducing an artificial time, t, κ_1 is then updated according to the gradient decent rule $\frac{\partial \kappa_1}{\partial t} = -\frac{\partial E}{\partial \kappa_1}$, where

$$\frac{\partial E}{\partial \kappa_1} = 2\alpha \sum_{i=1}^{L} \int_\Omega (x^u \cos\theta_i + y^u \sin\theta_i - \rho_i)\left[(x-c_x)r^2\cos\theta_i + (y-c_y)r^2\sin\theta_i\right]H(\phi_i)\,dxdy. \tag{20}$$

Finally we consider the minimization with respect to ϕ_i, which is done by deducing the associated Euler-Lagrange equation for ϕ_i:

$$\begin{aligned}\frac{\partial \phi_i}{\partial t} = -\frac{\partial E}{\partial \phi_i} = \delta_\varepsilon(\phi_i)\Bigg[\mu\, div\left(\frac{\nabla\phi_i}{|\nabla\phi_i|}\right) - v - \lambda_1(f_i - c_1{}^i)^2 + \lambda_2(f_i - c_2{}^i) \\ - \alpha(\rho_i - x^u\cos\theta_i - y^u\sin\theta_i)^2\Bigg],\end{aligned} \tag{21}$$

where the scalars ρ_i, θ_i, $c_1{}^i$, $c_2{}^i$, and κ_1 are updated with the level set evolution according to (12, 14, 17, 18, 19). The weights have the same meaning as before. In the steady state the value of κ_1 is the required lens distortion coefficient. In our implementation, we choose the regularizing parameters as follows: First half of the iterations (total = 200), we choose a small α to decrease the effect of line parameters allowing all objects in the image be segmented. We use the values: $\lambda_1 = \lambda_2 = 10$, $\mu = 0.001\times 255^2$, $v = \alpha = 0$. In the second half of iterations, we increase the effect of the line parameters and the area parameter to detect lines objects and ignore other objects in the image. The values are, $v = 1\times 10^6$, $\alpha = 600$. For highly noisy images we choose the value of $\mu = 0.5\times 255^2$ so that the zero level set of ϕ ignores the small objects (due to noise) during the evolution.

One point still needs some clarification. The input image is initially partitioned into multiple level sets, each containing an image object. The question is what happens if one such object is not a line or a curved line. In this case, the final shape of its zero level set for that object will be its axis of second moment (elongation axis). To get around this, the weight of the area term, v, in our formulation is increased. This causes the level set ϕ_i associated with a non-linear object to favor taking on negative

values all over the image f_i thus ignoring the undesired object without affecting the distortion calibration procedure, see Fig. 1. Fig. 2 illustrates the separation of an image into a number of level sets. Please note that this method thus detects only linear or nearly linear objects in an image; It discards objects of other shapes, e.g., a box (even if it has straight linear boundaries).

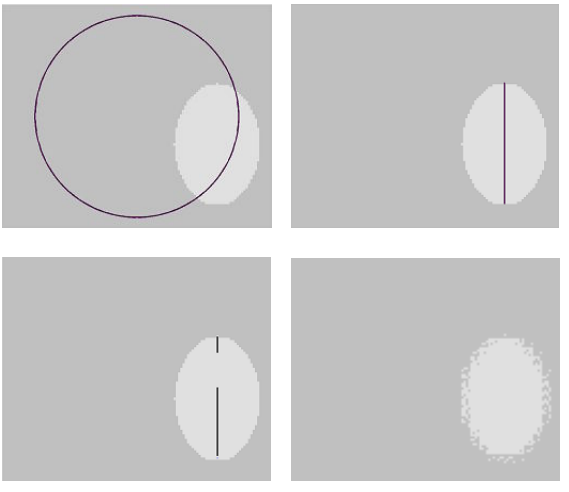


Fig. 1. Evolution of the boundary for a level set under functional (10). Due to the E_{Line} term, the final shape of the boundary is the second moment axis of the object~Increasing ν causes smaller part of the object axis be detected. Further increase in ν ignores the non-linear object~

5 Experimental Results

In this section, the performance of our technique is assessed using both synthetic and real image data. The synthetic images provide exact knowledge of line positions, orientations and distortion parameters, so precise quantitative evaluation of performance is possible. The performance on real images is shown to demonstrate the practical implementation of the technique.

5.1 Synthetic Data

A 160 ×120 image consisting of 6 lines is used as a test image as shown in Fig. 3(a). Using a big distortion coefficient $\kappa_1 = 1\times10^{-4}$, the lines' points were distorted, see Fig. 3(b). Applying the proposed method, the value of distortion coefficient was found $\kappa_1 = 9.6215\times10^{-5}$, which is close enough.

To simulate non-ideal conditions in real images, the distorted image is noised by a Gaussian noise with zero-mean and standard deviation, σ, that is varied from 5 to 40 with a step of 5. We then used our approach to estimate the distortion parameter

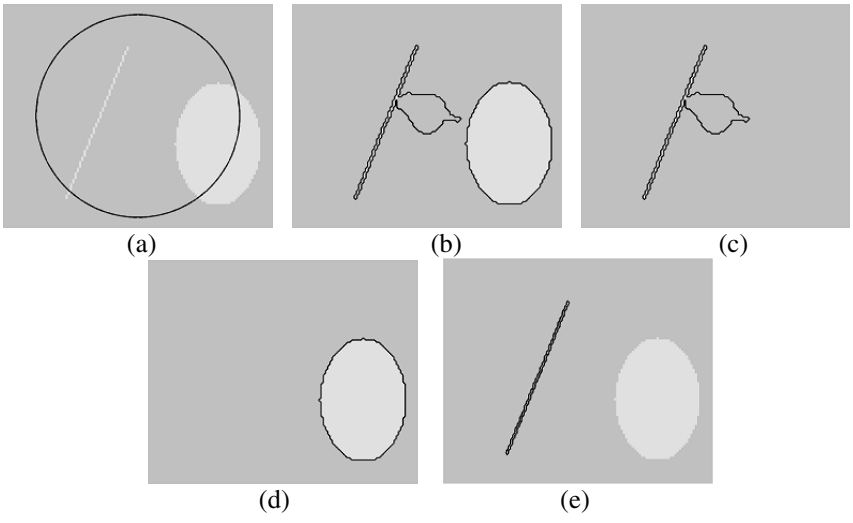


Fig. 2. The steps to extract a line in an image with multiple objects: (a) initial image and contour, (b) initial segmented objects (L=2), (c) f_1 and associated ϕ_1, (d) f_2 and associated ϕ_2 (d) final $\phi = \max(\phi_1, \phi_2)$

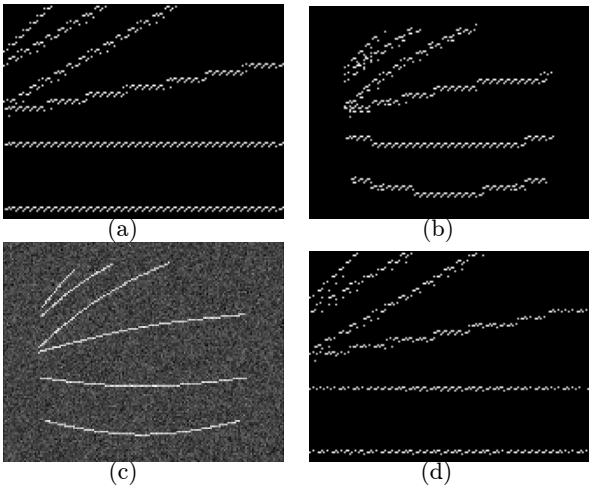


Fig. 3. Performance on synthetic images: (a) the true image, (b) distorted image $\kappa_1 = 1 \times 10^{-4}$, (c) noised image $\sigma = 35$, (d) undistorted image

from the noisy data. A sample noisy distorted image is shown in Fig. 3(c) for σ =*35.* The distortion coefficient estimated by our approach is then used to correct the distorted image (before adding the noise) and the output image is shown in Fig. 3(d). Since comparing the ground truth distortion parameters and the estimated ones does not give an intuitive feel of how much distortion remains in the image, we have con-

sidered the *RMS* error in pixels between the true and the undistorted line points as an accuracy measure [14]. We used this measure, denoted E_{RMS}, to evaluate the accuracy of the recovered radial parameter versus different noise levels. The value of E_{RMS} at each σ is plotted in Fig. 4.

For the sake of comparison, the same experiment is repeated using two well-known methods [7] and [19], which are based on the minimization of distortion measures of image lines. Prescott and McLean [19] minimized the sum of square distances of the lines' points from their best-fit lines, while Devernay and Faugeras [7] minimized the χ^2 of the line least square approximation. Note both methods need points belonging to image lines be extracted before the optimization procedures. In our implementation, we used the Levenberg-Marquardt (LM) algorithm to find the radial distortion coefficient minimizing each of the distortion measures. The values of E_{RMS} for both methods versus σ are also shown in Fig. 4.

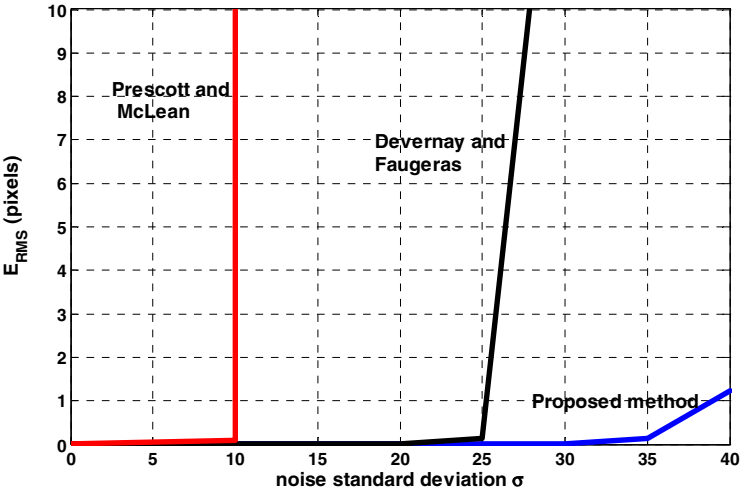


Fig. 4. The variation of E_{RMS} versus noise standard deviation (σ) for three methods

As shown in Fig. 4, the accuracy of our proposed method remains within 0.1 pixels up to a high noise level of $\cong$35. Afterwards, the error starts to increase rapidly. Both the other two methods start to be effected significantly by noise at lower levels. This proves that our approach is quite robust to additive noise, especially in comparison to the other two methods. This is attributed to the fact that both methods, similar to most existing techniques for lens distortion calibration in literature, proceed after applying a feature extraction stage, typically, based on edge detection and edge linking, which can be severely affected by high noise levels. Even if the input image has been pre-processed with a low-pass filter (in our implementation of the methods in [7] and [19], we smoothed out the noisy input images with a 7x7 Gaussian mask, which was not used at all in our method), the input to the distortion estimation process would still be contaminated with pretty much useless data.

5.2 Real Data

This approach is also applied real images acquired by a BenQ camera. To calibrate the radial lens distortion coefficient, we captured an image of a group of straight lines on a white paper, see Fig. 5(a). Such a calibration pattern is easily prepared (e.g., with just a printer) without any special construction overhead. Another sample image captured by the same camera is shown in Fig. 5(b). Both acquired images are 160×120 and have noticeable lens distortion. Our approach is then applied to the calibration image to recover the value of lens distortion parameter. Figs. 5(c-d) show the initial and final zero-level sets, respectively. Our method took less than a minute on P4 2.8GHz pc. The estimated κ_1 is employed to undistort the original distorted images taken by the camera, see Fig. 5(e-f). Clearly 3D lines are indeed mapped to straight lines in the undistorted images. One may notice some artifacts with the undistorted images due to the inverse mapping of the distortion model, which can be fairly fixed, if desired, by doing some post-processing.

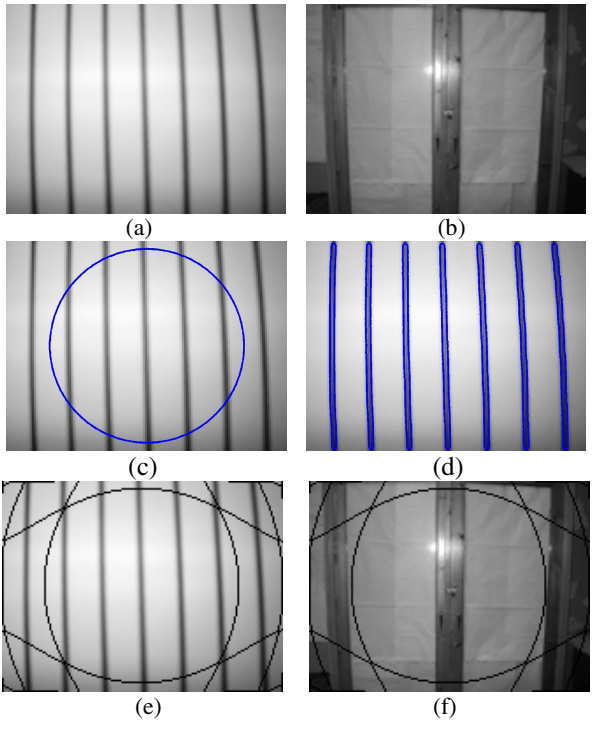


Fig. 5. Performance on real images: (a) the calibration image which is used to get κ_1, (b) input distorted image, (c) initial zero level set, (d) final zero level set (e) calibration image undistorted, (f) output undistorted image

6 Conclusions

We have presented a variational approach to calibrate camera lens distortion. In order to achieve this, we have extended the level set formulation of Chan and Vese by a term, which depends on the parameters of lens distortion parameters. By simultaneously minimizing the proposed energy functional with respect to the level set function and the line parameters, the linear shapes, the line parameters and lens distortion parameters are obtained.

All this approach needs is an image captured by the camera for a group of straight lines on a white paper. Such a calibration pattern is easily prepared (e.g., with just a printer) without any special construction overhead.

One key advantage of our method is that it integrates the extraction of image features needed for calibration and the computation of distortion parameters; thus avoiding, unlike most other nonmetric calibration methods, the propagation of errors in feature extraction onto the computation stage. This results in a more robust computation even at high noise levels. Our experimental results verified the accuracy and noise-robustness of our method.

Our future research is directed towards more comparisons between our method and other existing methods that recover lens distortion parameters. In addition, it is directed towards incorporating more lens distortion parameters in order to achieve more accurate calibration.

References

1. T. Chan and L. Vese. Active contours without edges, IEEE Trans. Image Processing, 10(2):266–277, 2001.
2. S. Osher and R. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag, 2003.
3. S. Osher and N. Paragios, Geometric Level Set Methods in Imaging, Vision and Graphics. Springer Verlag, 2003.
4. J. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanisms, Computer Vision, and Material Sciences. Cambridge University Press, 1999.
5. J.Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *PAMI*, 14(10), Oct 1992.
6. R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off the shelf tv cameras and lenses. IEEE Journal of Robotics and Automation, vol. RA-3(4), pp. 1165-1172, Aug. 1987.
7. F. Devernay and O. Faugeras, Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments, Machine Vision and Applications, Vol. 1, 14-24, 2001.
8. S. Becker and V. Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In Proc. SPIE Visual Data Exploration and Analysis II, volume 2410, pages 447–461, February 1995.
9. S. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 364–370, June 1996.

10. C. Taylor, P. Debevec, and J. Malik. Reconstructing polyhedral models of architectural scenes from photographs. In Fourth European Conference on Computer Vision (ECCV'96), volume 2, pages 659–668, Cambridge, England, April 1996. Springer-Verlag.
11. L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. Computer Graphics (SIGGRAPH'95), pages 39–46, August 1995.
12. R. Szeliski. Image Mosaicing for Tele-Reality Applications. In IEEE Workshop on Applications of Computer Vision (WACV'94), pages 44–53, Sarasota, Florida, December 1994. IEEE Computer Society.
13. R. Szeliski and S. Kang. Direct methods for visual scene reconstruction. In IEEE Workshop on Representations of Visual Scenes, pages 26–33, Cambridge, Massachusetts, June 1995.
14. S. Kang, Radial distortion snakes, IAPR Workshop on Machine Vision Applications (MVA2000), Tokyo, Japan, Nov. 2000, pp. 603-606.
15. T. Chan, B. Sandberg and L. Vese, Active Contours without Edges for Vector-Valued Images, Journal Visual Communication and Image Rep. 11, 130-141 (2000).
16. T. Chan and L. Vese. A level set algorithm for minimizing the Mumford–Shah functional in image processing. In IEEE Workshop on Variational and Level Set Methods, pages 161–168, Vancouver, CA, 2001.
17. L. Vese and T. Chan. A Multiphase levelset Framework for Image Segmentation using the Mumford and Shah Model. In international Journal of Computer Vision 50(3), 271–293, 2002.
18. D. Brown. Close-range camera calibration. Phtogrammetric Engineering, 37(8):855–866, Aug 1971.
19. B. Prescott and G. McLean. Line-based correction of radial lens distortion. Graphical Models and Image Processing, 59(1):39–47, 1997
20. R. Swaminathan and S. Nayar. Non-metric calibration of wide-angle lenses and polycameras. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), 22(10), Oct. 2000.

Author Index