

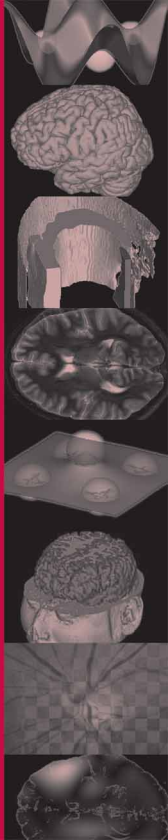
Handbook of
Biomedical Image
Analysis

Volume I:

**Segmentation
Models**

Part A

Jasjit S. Suri
David L. Wilson
Swamy Laxminarayan



*Handbook of Biomedical
Image Analysis*

TOPICS IN BIOMEDICAL ENGINEERING INTERNATIONAL BOOK SERIES

Series Editor: Evangelia Micheli-Tzanakou

*Rutgers University
Piscataway, New Jersey*

Signals and Systems in Biomedical Engineering:

Signal Processing and Physiological Systems Modeling
Suresh R. Devasahayam

Models of the Visual System

Edited by George K. Hung and Kenneth J. Ciuffreda

PDE and Level Sets: Algorithmic Approaches to Static and Motion Imagery

Edited by Jasjit S. Suri and Swamy Laxminarayan

Frontiers in Biomedical Engineering:

Proceedings of the World Congress for Chinese Biomedical Engineers
Edited by Ned H.C. Hwang and Savio L-Y. Woo

Handbook of Biomedical Image Analysis:

Volume I: Segmentation Models Part A
Edited by Jasjit S. Suri, David L. Wilson, and Swamy Laxminarayan

Handbook of Biomedical Image Analysis:

Volume II: Segmentation Models Part B
Edited by Jasjit S. Suri, David L. Wilson, and Swamy Laxminarayan

Handbook of Biomedical Image Analysis:

Volume III: Registration Models
Edited by Jasjit S. Suri, David L. Wilson, and Swamy Laxminarayan

Handbook of Biomedical Image Analysis

Volume I: Segmentation Models Part A

Edited by

Jasjit S. Suri

*Department of Biomedical Engineering
Case Western Reserve University
Cleveland, Ohio*

David L. Wilson

*Department of Biomedical Engineering
Case Western Reserve University
Cleveland, Ohio*

and

Swamy Laxminarayan

*Institute of Rural Health
Idaho State University
Pocatello, Idaho*

Kluwer Academic / Plenum Publishers

New York, Boston, Dordrecht, London, Moscow

ISBN 0-306-48550-8
eISBN 0-306-48551-6
set ISBN: 0-387-23126-9

©2005 Kluwer Academic / Plenum Publishers, New York
233 Spring Street, New York, New York 10013

<http://www.wkap.nl/>

10 9 8 7 6 5 4 3 2 1

A C.I.P. record for this book is available from the Library of Congress

All rights reserved

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording, or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Permissions for books published in Europe: *permissions@wkap.nl*

Permissions for books published in the United States of America: *permissions@wkap.com*

Printed in the United States of America

Jasjit Suri would like to dedicate this handbook to
his youngest uncle Paramjeet Chadha and his immediate family:
his late sister Sharan, his late brother Amarjeet, and his
late parents Kulwant Kaur and Udam S. Chadha
(Fellow of Royal Institute of London).

David Wilson would like to dedicate this handbook to his
family and students.

Swamy Laxminarayan would like to dedicate
this book in memory of his beloved parents who were a
constant source of inspiration in his life and to his in-laws
Corie and Derk Zwakman for their genuine sense
of family attachments and friendship.

Contributors

Alessandro Sarti, Ph.D.
University of Bologna
Bologna, Italy

Aly A. Farag, Ph.D.
University of Louisville
Louisville, KY, USA

Andrew Laine, Ph.D.
Columbia University
New York, NY, USA

Avdhesh Sharma, Ph.D.
Indian Institute of Delhi
New Delhi, India

Ayman ElBaz, M.S.
University of Louisville
Louisville, KY, USA

David Chopp, Ph.D.
Northwestern University
Chicago, IL, USA

David E. Breen, Ph.D.
Drexel University
Philadelphia, PA, USA

David L. Wilson, Ph.D.
Case Western Reserve University
Cleveland, OH, USA

Elsa Angelini, Ph.D.
Columbia University
New York, NY, USA

Fiorella Sgallari, Ph.D.
University of Bologna
Bologna, Italy

Gilson A. Giraldi, Ph.D.
National Laboratory for Scientific
Computing
Petrópolis, Rio de Janeiro, Brazil

Hossam Hassan, M.S.
University of Louisville
Louisville, KY, USA

Jasjit S. Suri, Ph.D.
Case Western Reserve University
Cleveland, OH, USA

Karol Mikula, Ph.D.
Slovak University of Technology
Bratislava, Slovakia

Ken Museth, Ph.D.
Linköping University
Norrköping, Sweden

Koon-Pong Wong, Ph.D.
Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Leandro S. Marturelli, B.S.

National Laboratory for Scientific
Computing
Petrópolis, Rio de Janeiro, Brazil

Leonid Zhukov, Ph.D.

California Institute of Technology
Pasadena, CA, USA

Mahmed N. Ahmed, Ph.D.

Lexmark International Inc.,
Lexington, KY, USA

Majid Mirmehdi, Ph.D.

University of Bristol
Bristol, UK

Misael Rosales, Ph.D.

Universidad de los Andes
Mérida/Venezuela

Paulo S. Rodrigues, Ph.D.

National Laboratory for Scientific
Computing
Petrópolis, Rio de Janeiro, Brazil

Petia Raveda, Ph.D.

Universitat Autònoma de Barcelona
Barcelona, Spain

Rakesh Sharma, Ph.D.

Columbia University
New York, NY, USA

Rodrigo L. S. Silva, M.Sc.

National Laboratory for Scientific
Computing
Petrópolis, Rio de Janeiro, Brazil

Ross Whitaker, Ph.D.

University of Utah
Salt Lake City, UT, USA

Sameer Singh, Ph.D.

University of Exeter
Exeter, UK

Swamy Laxminarayan, Dsc.

State University of Idaho
Pocatello, ID, USA

Xiaoping Shen, Ph.D.

Ohio University
Athens, OH, USA

Xianghua Xie, M.S.

University of Bristol
Bristol, UK

Yinpeng Jin, Ph.D.

Columbia University
New York, NY, USA

Acknowledgments

This book is the result of collective endeavor of several noted engineering and computer scientists, mathematicians, medical doctors, physicists, and radiologists. The editors are indebted to all of their efforts and outstanding scientific contributions. The editors are particularly grateful to Drs. Petia Reveda, Alex Falco, Andrew Laine, David Breen, David Chopp, C. C. Lu, Gary Christensen, Dirk Vandermeulen, Aly Farag, Alejandro Frangi, Gilson Antonio Giralaldi, Gabor Szekely, Pierre Hellier, Gabor Herman, Ardeshir Coshtasby, Jan Kybic, Jeff Weiss, Jean-Claude Klein, Majid Mirmehdi, Maria Kallergi, Yangming Zhu, Sunanda Mitra, Sameer Singh, Alessandro Sarti, Xioping Shen, Calvin R. Maurer, Jr., Yoshinobu Sato, Koon-Pong Wong, Avdhesh Sharma, Rakesh Sharma, and Chun Yuan and their team members for working with us so closely in meeting all of the deadlines of the book. They would like to express their appreciation to Kluwer Publishers for helping create this invitational handbook. They are particularly thankful to Aaron Johnson, the acquisition editor and Shoshana Sternlicht for their excellent coordination for the book at every stage.

Dr. Suri would like to thank Philips Medical Systems, Inc. for the MR datasets and encouragements during his experiments and research. Special thanks are due to Dr. Larry Kasuboski and Dr. Elaine Keeler at Philips Medical Systems, Inc. for their support and motivations. Thanks are also due to his past Ph.D. committee research professors, particularly Professors Linda Shapiro, Robert M. Haralick, Dean Lytle and Arun Somani, for their encouragements.

We extend our appreciations to Drs. Ajit Singh, Siemens Medical Systems, George Thoma, chief, Imaging Science Division at the National Institutes of Health, Dr. Sameer Singh, University of Exeter, UK for his motivations. Special thanks go to the book series editor, Professor Evangelia Micheli Tzanakou for advising us on all aspects of the book.

We thank the IEEE Press, Academic Press, Springer-Verlag Publishers, and several medical and engineering journals publishers for permitting us to use some of the images previously published in these journals.

Finally, Jasjit Suri would like to thank my wife Malvika Suri for all the love and support she has showered over the years and to our baby Harman

whose presence is always a constant source of pride and joy. I also express my gratitude to my father, a mathematician, who inspired me throughout my life and career, and to my late mother, who most unfortunately passed away a few days before my Ph.D. graduation, and who so much wanted to see me write this book. Special thanks to Pom Chadha and his family, who taught me life is not just books. He is one of my best friend. I would like to also thank my in-laws who have a special place for me in their hearts and have shown lots of love and care for me.

David Wilson would like to acknowledge the support of the department of biomedical engineering. Case Western Reserve University in this endeavor. Special thanks are due to the many colleagues and students who make research in biomedical engineering an exciting, wondrous endeavor.

Swamy Laxminarayan would like to express my loving acknowledgments to my wife Marijke and to my kids, Malini and Vinod, for always giving the strength of mind amidst all life's frustrations. The book kindies fondest memories of my late parents who made many personal sacrifices that helped shape our careers and the support of my family members who were always there for me when I needed them most. I have shared many ideas and thoughts on the book with numerous of my friends and colleagues in the discipline. I acknowledge their friendship, feedbacks, and discussions with particular thanks to Professor David Kristol of the New Jersey Institute of Technology. Peter Brett of Ashton University, Ewart Carson of the City University, London, Laura Roa of the University of Sevilla in Spain, and Jean Louis Coatrieux of the University of Rennes in France for their constant support over the past two decades.

Preface

Chapter 1 presents IVUS. Intravascular ultrasound images represent a unique tool to guide interventional coronary procedures; this technique allows to supervise the cross-sectional locations of the vessel morphology and to provide quantitative and qualitative information about the causes and severity of coronary diseases. At the moment, the automatic extraction of this kind of information is performed without taking into account the basic signal principles that guide the process of image generation. In this handbook, we overview the main physical principles and factors that affect the IVUS generation; we propose a simple physics-based approach for IVUS image simulation that is defined as a discrete representation of the tissue by individual scatterers elements with given spatial distribution and backscattering differential cross sections. In order to generate the physical model that allows to construct synthetic IVUS images, we analyze the process of pulse emission, transmission, and reception of the ultrasound signal as well as its interaction with the different tissues scatterers of the simulated artery. In order to obtain the 3D synthetic image sequences, we involve the dynamic behavior of the heart/arteries and the catheter movement in the image generation model. Having an image formation model allows to study the physics parameters that participate during the image generation and to achieve a better understanding and robust interpretation of IVUS image structures. Moreover, this model allows to comprehend, simulate, and solve several limitations of IVUS sequences, to extract important image parameters to be taken into account when developing robust image processing algorithms as well as to construct wide synthetic image sequence databases in order to validate different image processing techniques.

Chapter 2 presents research in PET. The last few decades of the twentieth century have witnessed significant advances in multidimensional medical imaging, which enabled us to view noninvasively the anatomic structure of internal organs with unprecedented precision and to recognize any gross pathology of organs and diseases without the need to “open” the body. This marked a new era of medical diagnostics with many invasive and potentially morbid procedures being substituted by noninvasive cross-sectional

imaging. Continuing advances in instrumentation and computer technologies also accelerated the development of various multidimensional imaging modalities that possess a great potential for providing, in addition to structural information, dynamic and functional information on biochemical and pathophysiological processes or organs of the human body. There is no doubt that substantial progress has been achieved in delivering health care more efficiently and in improving disease management, and that diagnostic imaging techniques have played a decisive role in routine clinical practice in almost all disciplines of contemporary medicine. With further development of functional imaging techniques, in conjunction with continuing progress in molecular biology and functional genomics, it is anticipated that we will be able to visualize and determine the actual molecular errors in a specific disease very soon, and be able to incorporate this biological information into clinical management of that particular group of patients. This is definitely not achievable with the use of structural imaging techniques. In this chapter, we will take a quick tour of a functional imaging technique called positron emission tomography (PET), which is a primer biologic imaging tool able to provide *in vivo* quantitative functional information in most organ systems of the body. An overview of this imaging technique including the basic principles and instrumentation, methods of image reconstruction from projections, some specific correction factors necessary to achieve quantitative images are presented. Basic assumptions and special requirements for quantitation are briefly discussed. Quantitative analysis techniques based on the framework of tracer kinetic modeling for absolute quantification of physiological parameters of interest are also introduced in this chapter.

Chapter 3 presents MRI. With time and experience, flow characteristics in human body have become ideal modality for the evaluation of vascular disease. With this aim, in this chapter, the topic of magnetic resonance angiography (MRA) is discussed with emphasis of physical principles of MRA techniques in current practice for acquisition and display of vascular anatomy as well as flow measurement. The main techniques are described, 1 TOF (time of flight) MRA; 2. phase contrast (PC) MRA. Each technique can be performed using 2D Fourier transform (2DFT) or 3D FT(3DFT). Each of these techniques lends itself to a different type of clinical application. TOF MRA is based on 2D or 3D gradient echo (GRE) technique. For 2D TOF MRA, use of presaturation pulse and slice selection above or below each slice are discussed to eliminate the signal from vessels flowing in the opposite direction. Usually, FC (flow compensation) is used perpendicular to the vessel lumen. Considerations on selection of scan parameters TR and TE are discussed. The performance is highlighted in faster scanning and capability of increased FSE because each slice is an echo slice. 3D TOF MRA pulse sequence for 3D TOF MRA is described as capable of obtaining a slab of up to 60 slices. Basic physics principles are highlighted for inflow effects (TOF), TR effect, imaging volume effect, flow phase effects (volume imaging), gradient motion rephasing by velocity rephasing, high order rephasing, inflow rephasing, and turbulent flow effect. Various acquisition methods are discussed in current practice such as inflow (TOF), flow adjusted gradient (FLAG), rapid

sequential excitation (RSE), and black blood MRA. Flow imaging is highlighted to get better magnitude contrast, phase contrast, inflow 3D (volume imaging), and 2D(TOF), TR. Other advancements are discussed in the field of MRA such as flow suppression, inflow and maximum intensity projection (MIP) processing, quantification of flow in tissue by real time acquisition and evaluation (RACE), phase mapping, and bolus tracking. Physical principles of inflow analysis in vessels are current interest in MRA methods. In summary, new MRA approaches are on the horizon which promise earlier diagnosis of tissue ischemia by tissue-free water and perfusion besides vascular anatomy and pathology. Time and experience will demonstrate the possibility of combining parenchymal and vascular anatomy with end-organ perfusion into a single imaging examination. Recent state-of-the-art on the physical basis of MRA methods is introduced to interested readers.

Chapter 4 presents work in level sets. In this chapter, we will begin with a concise description of the basic level set method, its origins, and why it has become so popular in a wide range of applications. We will then discuss many of the modifications and enhancements to the level set method which have made it a robust tool for propagating fronts. These modifications will include the use of the fast marching method and the more general ordered upwind methods, narrow-banding, velocity extensions, and particle level set methods. The chapter will conclude with techniques used for coupling the level set method to different physical problems, including coupling to elliptic solvers such as the ghost-fluid method, immersed interface method, and the extended finite element method. Implementation details will be provided throughout the text so that the methods can be quickly implemented, as well as specific examples of results obtained with these methods.

Chapter 5 presents work in shaping in shading. In many applications, for instance, visual inspection in robot vision, autonomous land vehicle navigation to name a few, a procedure of recovering three-dimensional surfaces of unknown objects is of considerable practical interest. In this chapter, we consider one of the reconstruction methods: shape from shading (SFS). The chapter is organized as follows: the first section serves as a brief review of the SFS models, their history, and recent developments. Section 2 provides certain mathematical background related to SFS. It discusses some selected numerical methods for solving discretized SFS problems. The emphasis is given to the well developed method: finite difference method (FDM). Section 3 is devoted to the illustration of numerical techniques for solving SFS problems. It concerns related algorithms and their implementations. The section ends with a discussion about the advantages and disadvantages of the algorithms introduced in this section. The last section attempts to introduce the recently developed wavelet-based methods by using an example. A part of the section, however, is distributed to a brief introduction of the basic facts of wavelet theory. In the hope of readers being able to extrapolate the elements presented here to initiate the understanding of the subject on his/her own, the chapter concludes with some remarks on other advanced methods. Finally, we include an intensive set of references to make up

whatever important spirits for which the authors have indeed hardly to touch in this short chapter.

Chapter 6 presents work in the area of wavelet transforms. Wavelet transforms and other multiscale analysis functions have been used for compact signal and image representations in denoising, compression, and feature detection processing problems for about 20 years. Numerous research works have proven that space-frequency and space-scale expansions with this family of analysis functions provided a very efficient framework for signal or image data. The wavelet transform itself offers great design flexibility. Basis selection, spatial-frequency tiling, and various wavelet threshold strategies can be optimized for best adaptation to a processing application, data characteristics, and feature of interest. Fast implementation of wavelet transforms using a filter-bank framework enables real time processing capability. Instead of trying to replace standard image processing techniques, wavelet transforms offer an efficient representation of the signal, finely tuned to its intrinsic properties. By combining such representations with simple processing techniques in the transform domain, multiscale analysis can accomplish remarkable performance and efficiency for many image processing problems. Multiscale analysis has been found particularly successful for image denoising and enhancement problems given that a suitable separation of signal and noise can be achieved in the transform domain (i.e. after projection of an observation signal) based on their distinct localization and distribution in the spatial-frequency domain. With better correlation of significant features, wavelets were also proven to be very useful for detection and matching applications. One of the most important features of wavelet transforms is their multiresolution representation. Physiological analogies have suggested that wavelet transforms are similar to low level visual perception. From texture recognition and segmentation to image registration, such multiresolution analysis gives the possibility of investigating a particular problem at various spatial-frequency (scales). In many cases, a “coarse to fine” procedure can be implemented to improve the computational efficiency and robustness to data variations and noise.

Without trying to cover all the issues and research aspects of wavelet in medical imaging, we focus our discussion in this chapter on three topics: image denoising/enhancement, image segmentation, and image registration using wavelet transforms. We will introduce the wavelet multiscale analysis framework and summarize related research work in this area and describe recent state-of-the-art techniques.

Chapter 7 presents work in deformable models. Traditional deformable models cannot deal with topological changes and suffer from the strong sensitivity to the initial contour position. Besides, when using deformable surfaces, memory limitations can lower the performance of segmentation applications for large 3D images, or even make it undoable. In this chapter we describe some techniques to address these limitations. The problem of topological changes is addressed by the T-surfaces model by embedding a deformable model in the framework of a simplicial decomposition of the domain. The sensitivity to initialization has been addressed through automatic techniques for obtaining an initial contour

closer to the desired boundary. Efficient memory utilization can be obtained through out-of-core techniques. Also, diffusion schemes can be explored in order to improve the convergence of the model toward the desired boundaries. We demonstrate the discussed techniques for segmentation of both gray level and color medical images. Finally, we present perspectives and challenges in the area of deformable models.

Chapter 8 presents work in the area of application. This chapter describes level set techniques for extracting surface models from a broad variety of biological volume datasets. These techniques have been incorporated into a more general framework that includes other volume processing algorithms. The volume datasets are produced from standard 3D imaging devices such as MR, CT, ultrasound and are all noisy samplings of complex biological structures with boundaries that have low and often varying contrasts. The level set segmentation method, which is well documented in the literature, creates a new volume from the input data by solving an initial value partial differential equation (PDE) with user-defined feature-extracting terms. Given the local/global nature of these terms, proper initialization of the level set algorithm is extremely important. Thus, level set deformations alone are not sufficient, they must be combined with powerful preprocessing and data analysis techniques in order to produce successful segmentations. In this chapter the authors describe the preprocessing and data analysis techniques that have been developed for a number of segmentation applications, as well as the general structure of our framework. Several standard volume processing algorithms have been incorporated into the framework in order to segment datasets generated from MRI, CT and TEM scans. A technique based on moving least-squares has been developed for segmenting multiple nonuniform scans of a single object. New scalar measures have been defined for extracting structures from *diffusion tensor MRI scans*. Finally, a direct approach to the segmentation of incomplete tomographic data using *density parameter estimation* is presented. These techniques, combined with level set surface deformations, allow us to segment many different types of biological volume datasets.

Chapter 9 presents work in the area of segmentation using the tools of level sets and statistics. The book also demonstrates a chapter in the area of mathematical foundations of the statistical-based approaches that have been known to provide robust performance on medical imaging. Other approaches-based level sets will also be covered. We will introduce 2D and 3D phantoms to validate the theory. Applications will include CT, MRI, and MRA volume segmentation of the human brain, and CT segmentation of the lungs.

Chapter 10 presents work in the area of level sets with classifier embedded for snakes. The standard geometric or geodesic active contour is a powerful segmentation method, whose performance however is commonly affected by the presence of weak edges and noise. Since image modalities of all types in medical imaging are prone to such outcomes, it is important for geometric snakes to develop some level of immunity toward them. In this chapter, a region-aided geometric snake, enhanced for more tolerance toward weak edges and noise,

is introduced. It is based on the principle of the conjunction of the traditional gradient flow forces with new region constraints. We refer to this as the region-aided geometric snake or RAGS. The RAGS formulation is easily extended to deal with color images. Quantitative comparisons with other well-known geometric snakes in synthetic noisy images are presented. We also show the evaluation of RAGS with application to the localization of the optic disk in color retinal images. Many other images are also used to demonstrate the proposed method.

Finally, Chapter 11 presents work in the area of level set applications for missing boundaries. The authors present application of the semi-implicit complementary volume numerical scheme to solving level set formulation of Riemannian mean curvature flow in image segmentation, edge detection, missing boundary completion, and subjective contour extraction. Our computational method is robust, efficient, and stable without any restriction on a time step. The computational results related to medical image segmentation with partly missing boundaries and subjective contour extraction are presented.

Contents

1. A Basic Model for IVUS Image Simulation	1
Misael Rosales and Petia Radeva	
2. Quantitative Functional Imaging with Positron Emission Tomography: Principles and Instrumentation	57
Koon-Pong Wong	
3. Advances in Magnetic Resonance Angiography and Physical Principles	117
Rakesh Sharma, and Avdhesh Sharma	
4. Recent Advances in the Level Set Method	201
David Chopp	
5. Shape From Shading Models	257
Xiaoping Shen and Lin Yang	
6. Wavelets in Medical Image Processing: Denoising, Segmentation, and Registration	305
Yinpeng Jin, Elsa Angelini, and Andrew Laine	
7. Improving the Initialization, Convergence, and Memory Utilization for Deformable Models	359
Gilson A. Giraldi, Paulo S. Rodrigues, Leandro S. Marturelli, and Rodrigo L. S. Silva	
8. Level Set Segmentation of Biological Volume Datasets	415
David Breen, Ross Whitaker, Ken Museth, and Leonid Zhukov	
9. Advanced Segmentation Techniques	479
Aly A. Farag, Mohamed N. Ahmed, Ayman El-Baz, and Hossam Hassan	
10. A Region-Aided Color Geometric Snake	535
Xianghua Xie and Majid Mirmehdi	

11. Co-Volume Level Set Method in Subjective Surface
Based Medical Image Segmentation 583
Karol Mikula, Alessandro Sarti, and Fiorella Sgallari

The Editors 627

Index 633

Chapter 1

A Basic Model for IVUS Image Simulation

Misael Rosales¹ and Petia Radeva²

Introduction

Arteriosclerotic cardiovascular diseases [1], which increase their incidence in vertiginous form, were the principal cause of mortality in the industrialized countries in the latter half of the 1950s, and cause more deaths today than deaths due to cancer and accidents. In effect, arteriosclerotic diseases in general (infarct to miocardio, angina of chest, cerebrovascular accidents, and arterial diseases of the legs and arms) have caused, and continue causing, more deaths and increased consultations in emergency rooms and doctors' offices than any other disease in this century, including AIDS. Coronary arterial diseases develop due to a narrowing (stenosis) of the walls of the coronary arteries, caused by the accumulation of fibrotic material and calcium crystals [2]. In humans, the development of this type of plaque normally takes between 15 and 20 years. There exist complicated injuries, which generate the formation of thrombus; the rupture of the plaque can be considered an important mechanism of acute coronary syndromes. An imminent problem if the arteries narrow too much due to stenosis is that the blood flow diminishes significantly. A total occlusion can take

¹Laboratorio de Física Aplicada, Facultad de Ciencias, Departamento de Física de la Universidad de los Andes Mérida, Venezuela

²Centre de Visió per Computador, Universitat Autònoma de Barcelona, Edifici O, Campus UAB, 08193 Bellaterra, Barcelona, Spain

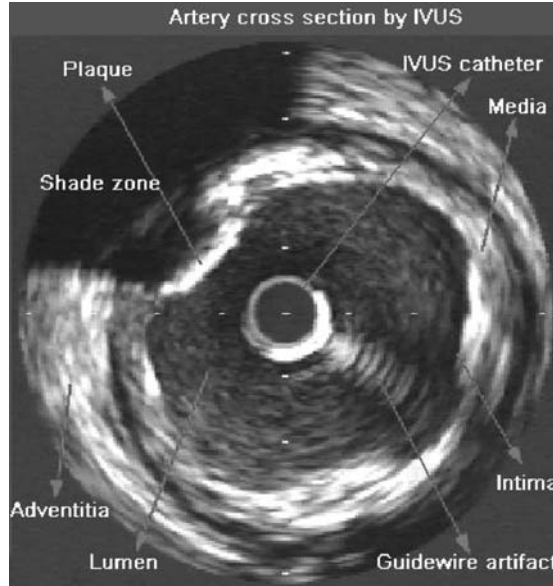


Figure 1.1: Typical 2D IVUS image indicating the location of the principal morphological arterial structures and artifacts.

place if the increase in stenosis persists and can become serious due to a thrombosis. The result can be an infarct. The introduction of intravascular ultrasound (IVUS) [3, 4] in the field of medical image as an exploratory technique has made a significant change to the understanding of the arterial diseases and individual patterns of diseases in coronary arteries. Although coronary angiography [5, 6] provides with 2D information about the coronary anatomy, serving as a guide in operations, it has limitations when not allowed to access the mechanism of the disease, its composition, and its extent. On the contrary, the IVUS technique shows the cross-section (Fig. 1.1) of the artery, allowing an evaluation of the plaque as well as of the different layers in the arterial wall. The IVUS image [2, 5, 6] provides qualitative (Fig. 1.2) information about the causes and severity of the narrowing of the arterial lumen, distinguishes the thrombus of the arteriosclerotic plaque, shows calcium deposits in the arterial wall, evaluates the changes and complications in the coronary arteries that occur after an intervention such as angioplasty, evaluates and diagnoses coronary arterial aneurysms, and diagnoses fissures of arterial coronary plaques: determination and location, dimensions, type (eccentric and concentric), and composition of the arteriosclerotic plaque.

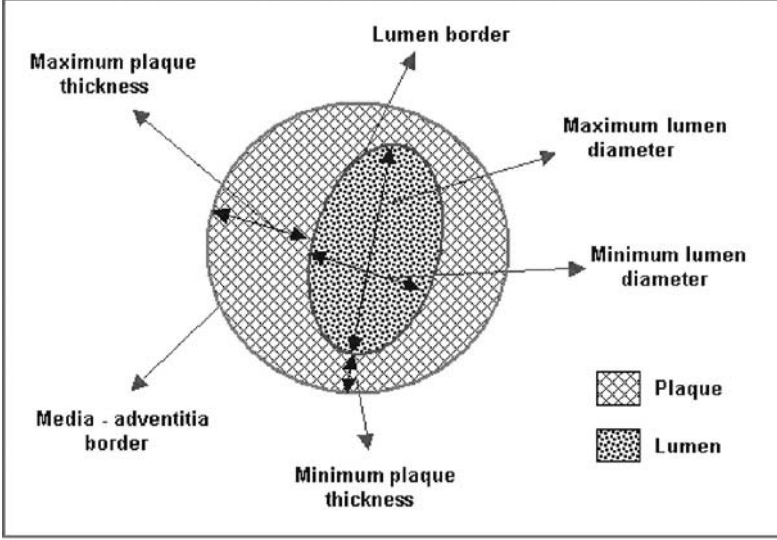


Figure 1.2: Geometric parameters of an artery that make it possible to measure by means of intravascular ultrasound.

1.1 Technical Aspects of the IVUS

The images of ultrasound [7] are based on the reception and transmission of high-frequency sound waves. The transmitted wave propagates through the material until it strikes the reflecting object. The reflected wave returns and is received by a transducer. The time interval between the transmission and the reception of the wave is directly related to the distance between the source and the reflector. The advantage of ultrasound is that it can travel through water smooth tissue. Additionally, ultrasound is not harmful at the levels of energy used for the generation of the image. A standard configuration of IVUS acquisition images consists of three components. Figure 1.3 shows a scheme of a catheter with a miniaturized piezoelectric transducer, the pull-back unit, and the console for reconstructing the images. IVUS catheter has a rank of measures that oscillates between 2.9 and 3.5 F (0.96–1.17 mm) in diameter. The quality of the image depends on the operation frequency, which is of the order of 20–50 MHz; the lateral resolution is approximately of the order of 113 μm and the axial resolution is of the order of 80 μm [8]. The IVUS images acquisition process is initiated when the catheter is manually (guided by the angiography) inserted within the artery (Fig. 1.3(a)). The catheter pull-back is made at linear constant velocity (usually 0.5 mm/sec)

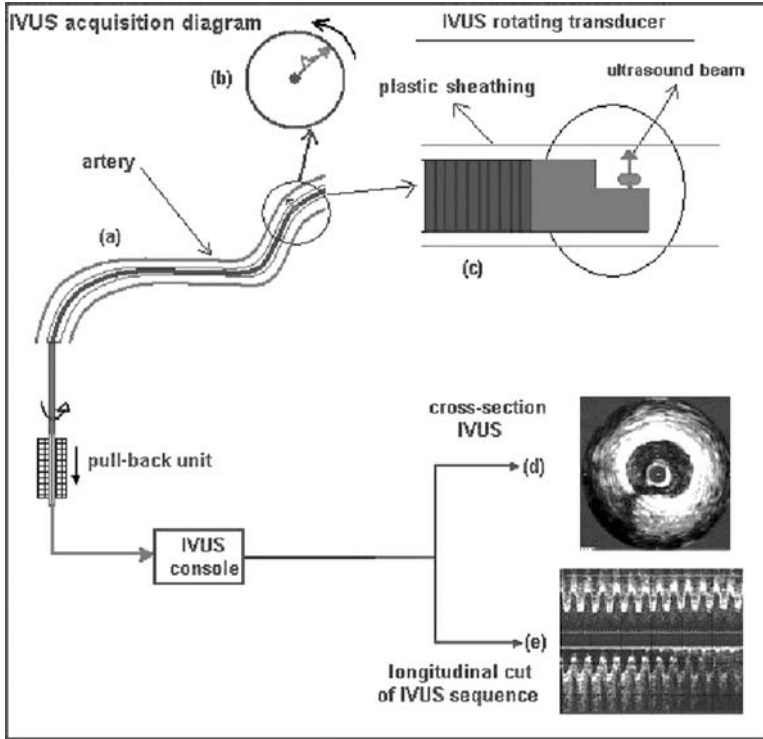


Figure 1.3: The IVUS catheter is manually positioned within the artery (a) and extracted by a pull-back unit at a constant linear velocity and rotated at a constant angular velocity. The information is transformed by the IVUS console as a unique cross-section artery gray-levels image (d) or a longitudinal image sequence (e).

and constant angular velocity of 1800 rev/min. The pivoting transducer sends a radially focused beam of ultrasound and receives its corresponding echoes. The radial lines obtained for different transducer angular positions are adequately processed, giving, as a result, a 2D cross-section artery image (Fig. 1.3(d)). The sequence can be shown as a longitudinal sequence, which gives a longitudinal artery cut (Fig. 1.3(e)). The resolution of an ultrasound image is directly related to the ultrasound signal frequency: high frequencies allow one to obtain better resolution. Nevertheless, when the frequency is increased, the attenuation of the waves of ultrasound increases while penetrating the biological tissue. The typical frequencies in the IVUS technique are in the rank of 20–50 MHz, with inferior resolutions of 50 μm .

1.2 Present Limitations of the IVUS Technique and the Need for a Generation Model of IVUS Data

The main function of the IVUS technique is to serve as a guide in the interventional procedures, allowing us to measure the cross section of the artery. The precision in the measurements of distance is subject to the following potential sources of error [9]:

- (a) *Incorrect identification of the surface and the sections to be measured.*
Although the vessel and the interface defining the wall vessel have sufficiently good acoustics in most of the cases, in several cases the identification of the surface and differentiating tissues can be difficult. Improving the radial resolution could improve the detection of contours, which would reduce the errors. These errors can in some cases be systematic and lead to an overestimation of the dimensions. This could get considerably worse if the irregularities of the vessels are very pronounced.
- (b) *Assumption that the sound speed is constant in the arterial structure.*
The second kind of problem related to the assumption of constant speed of the sound of 1540 m/sec is systematic and small (of the order of 1–2%), which brings as a consequence the propagation of the error in the location of each one of the structures under study.
- (c) *Artifacts caused by inhomogeneities in the rotation of the catheter and pronounced reverberations generated by very acute irregularities of the vessel.* The appearance of some artifacts such as the inhomogeneities in the rotation of the catheter influences the quality of the image. The absence of beams, when the catheter stops momentarily, brings as a consequence a propagation of errors in the tangential direction of the image.
- (d) *Presence of zones of acoustic shade, which prevents access to certain regions of interest (ROIs).* The presence of zones of acoustic shade is intimately related to the presence of calcification or regions of high acoustic impedance. The shades prevent some structures from being evaluated from the distribution of the gray levels.

- (e) *The presence of the catheter, the reticule, and the guide proves disadvantageous to the processing of the images* and to the evaluation of the data by some procedure of images processing.
- (f) *Impossibility of spatially locating the catheter.* The impossibility of locating the catheter with respect to a specific axis of coordinates makes it impossible to make any attempt for three-dimensional representation of the vessel only with the IVUS technique. For example, spatial location of the effective section of the lumen and location of plaque and the reconstruction in the lengthwise direction of the vessel are still an open problem of investigation [9].
- (g) *Impossibility of evaluating dynamic parameters, different from the single static characterization using the gray levels.* First achievements are related to IVUS elastography [7] the purpose of which is to propose a technique for tissue characterization.

The mentioned shortcomings are difficult to quantify and depend on the experience of the operator, that is he should have been trained in handling a large number of patient cases. Some of the limitations of the IVUS technique can be attenuated through algorithms of image processing; the limitations due to a sub-optimal location of the borders of the arterial structure can be overcome with new algorithms of segmentation. The question is how to develop robust algorithms that can solve these problems, analyzing the artifacts with their multiple appearances in IVUS images. Having a complete set of patient data to present all variance of artifacts appearance in images would mean to dispose of a huge number of patient cases. A more efficient solution is to develop a simulation model for IVUS data construction so that synthetic data is available in order to “train” image processing techniques. In this way, different appearances of artifacts can be designed to assure robust performance of image processing techniques.

Differences in IVUS data are caused not only by different morphological structures of vessels but also by different parameters that influence the formation of IVUS images. The images depend on the IVUS apparatus calibration as well as on interventional devices; small differences in parameters can lead to a different gray-level appearance that can be interpreted in a different way by physicians. A simulation model for IVUS data can help train the medical staff as well as play an important role in designing and testing new interventional

devices. At the end, being aware which parameters and in which grade influence to image formation is of unquestionable importance for all persons involved in comprehension of IVUS data and taking final decision for diagnosis and intervention of vessel lesions. In this chapter, we discuss a simple simulation model for the formation of 2D IVUS data that explains the complete process of data generation as a result of the interaction between ultrasound signals and vessel morphological structures.

1.2.1 A Basic IVUS Image Model

Correct image processing needs an understanding of image formation, gray-level meaning, artifact causes, the averaging, and the motion of the dynamics structures effects in the image. The generation of simulated IVUS images investigates four important aspects: (a) The generation, processing, and visualization of the data in the format that doctors use, (b) the exploration of some of the artifacts generated by the averaging of the beams, (c) the smoothing and treatment of the images to generate sufficient data for the validation of image processing algorithms, and (d) comparison of data generated by the image formation model with the real data. IVUS images can be obtained in a simulated form, from a simple physical model based on the transmission and reception of high-frequency sound waves, when these radially penetrate a simulated arterial structure (Fig. 1.4). We assume that for this model the waves are emitted by a transducer located at the center of the artery and that these waves propagate radially through the blood and the arterial structures (intima, media, and adventitia), being reflected progressively by them. The reflected waves or echoes that return are received by the transducer, which now behaves as a receiver. The time interval between the emission and the reception of the waves is directly related to the distance between the source and the reflector (Fig. 1.5). The echo amplitude, which is a function of time, is transformed on gray scale and later to penetration depth, so the radial coordinate is determined. If we place a rotatory transducer, make a registry of the corresponding echoes for each angular position of the transducer, and combine all the lines obtained from different positions, we will be able to obtain a simulated 2D image of the structure under study. The 3D IVUS simulated images can be generated as a sequence of n-planes generated independently, taking into account the arterial deformation caused by the blood pulsatile pressure.

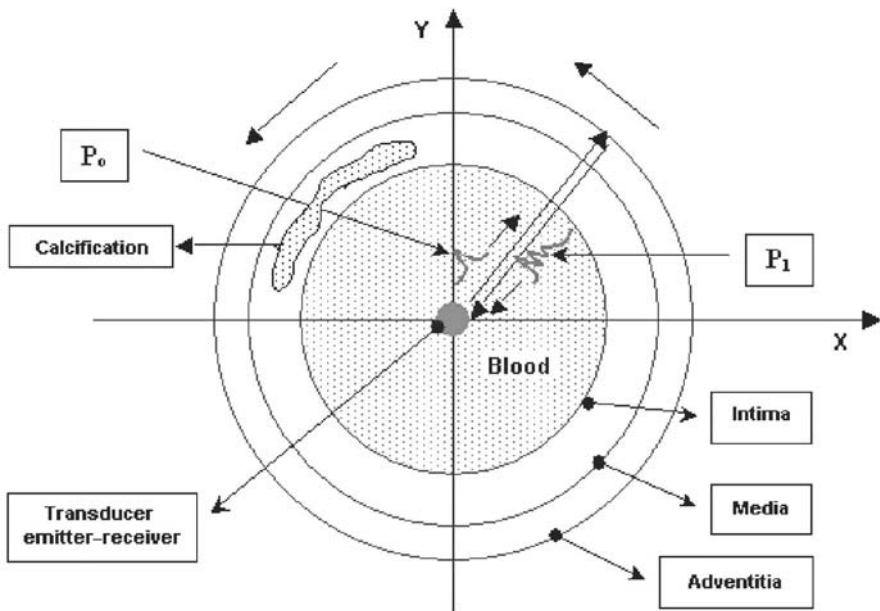


Figure 1.4: The disposition of the simulated arterial structures (blood, intima, media, and adventitia) and calcification are illustrated. The ultrasound rotatory transducer, which emits the pulse P_0 and receives pulse P_1 , has been placed at the coordinate center.

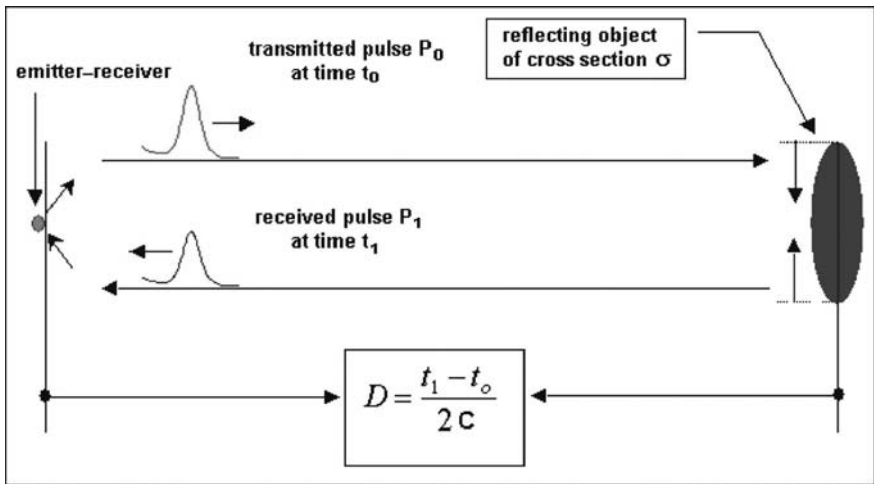


Figure 1.5: The determination of the distance D between the emitter/receiver and the reflecting object is made from the difference of time interval between the transmitted pulse P_0 and the received pulse P_1 , assuming that the pulse speed c is constant.

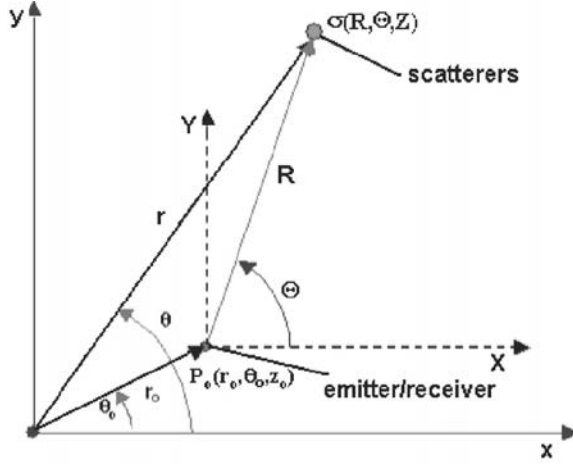


Figure 1.6: Coordinates system used with the corresponding ultrasound emitter/receiver and the scatterers localization.

1.3 Formal Definition of the Image Model

Let us consider an ultrasound pulse P_o that is emitted at time t_o with speed c from a point with coordinates (r_o, θ_o, z_o) (Fig. 1.6), and that interacts with the scatterer located at position (R, Θ, Z) with the spatial distribution of the differential backscattering cross-section, $\sigma(R, \Theta, Z)$. The reflected pulse P_i for the i th scatterer is an exact replica [10] of the transmitted sound pulse P_o that will return to the point (r_o, θ_o, z_o) at time $(t_i - t_o)$ and will be out of phase temporarily with respect to the pulse P_o by time difference $\delta = t_i - t_o$ between the emitted pulse at t_i and the received pulse at t_o . The time delay δ is given by

$$\delta = \frac{2|R|}{c} \quad (1.1)$$

$$\vec{R} = \vec{r} - \vec{r}_o, \quad \vec{r} = x\hat{i} + y\hat{j} + z\hat{k}, \quad \vec{r}_o = x_o\hat{i} + y_o\hat{j} + z_o\hat{k}$$

We choose a coordinate system (X, Y, Z) with respect to the emitter/receiver position:

$$\vec{X} = (x - x_o)\hat{i}, \quad \vec{Y} = (y - y_o)\hat{j}, \quad \vec{Z} = (z - z_o)\hat{k}$$

and the corresponding cylindrical coordinates are given by

$$|R| = \sqrt{X^2 + Y^2 + Z^2}, \quad \Theta = \arctan(Y/X)$$

where $X = |\vec{X}|$, $Y = |\vec{Y}|$, and $Z = |\vec{Z}|$.

Assuming the Born approximation [11, 12], the ultrasound reflected signal $S(t, \tau)$ for a finite set of N reflecting scatterers with coordinates (R, Θ, Z) and spatial distribution of the differential backscattering cross-section $\sigma(R, \Theta, Z)$ is given by:

$$S(R, \Theta, Z, t, \tau) = \sum_{i=1}^N \sigma_i(R, \Theta, Z) \zeta_i(t, \tau) \quad (1.2)$$

where N is the number of scatterers, $\sigma_i(R, \Theta, Z)$ is the spatial distribution of the differential backscattering cross-section (DBC) of the i th scatterer located at position (R, Θ, Z) , $\zeta_i(t, \tau)$ is the transducer impulse function, and τ is the delay time which leads to constructive and destructive contributions to the received signal. The Born approximation implies that the scattered echoes are weak compared to the incident signal and it is possible to use the principle of superposition to represent the wave scattered by a collection of particles by adding their respective contribution.

1.3.1 The Ultrasound Pulse

We consider a planar transducer that is mounted inside an infinite baffle, so that the ultrasound is only radiated in the forward direction. We assumed that the transducer is excited with uniform particle velocity across its face [13, 14]. According to the coordinates system illustrated in the far field circular transducer, pressure $P(r, \theta, t)$ (Fig. 1.7) can be written as:

$$P(r, \theta, t) = j \frac{\rho_o c k a^2 v_o}{2r} \left[\frac{2J_1(ka \sin(\theta))}{ka \sin(\theta)} \right] \exp(j(\omega t - kr))$$

where t is time, ρ_o is the medium propagation density, c is the sound speed for biological tissue (typically $c = 1540$ m/sec), v_o is the radial speed at a point on the transducer surface, a is the transducer radius, $\vec{k}$ is the propagation vector, defined as $k = |\vec{k}| = 2\pi/\lambda$, where λ is the ultrasound wavelength defined as $\lambda = c/f_o$, where f_o is ultrasound frequency, $\omega = 2\pi f_o$, and $J_1(x)$ is the first class Bessel function. Figure 1.8 shows a graphics of the pressure as a function of v , where $v = ka \sin(\theta)$. In some applications, particularly when discussing biological effects of ultrasound, it is useful to specify the acoustic intensity [16]. The intensity at a location in an ultrasound beam, I , is proportional to the square of the pressure amplitude P . The actual relationship is:

$$I(r, \theta, t) = \frac{P(r, \theta, t)^2}{2\rho c} \quad (1.3)$$

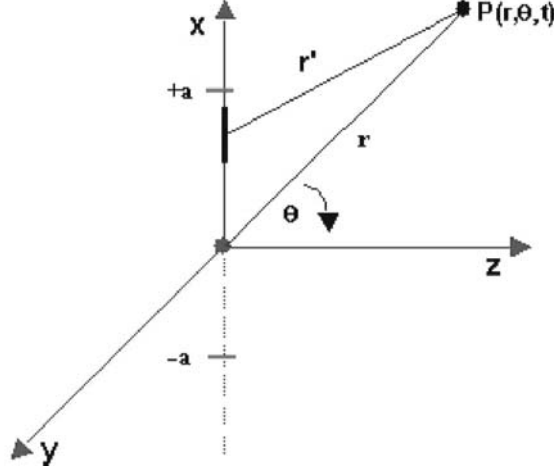


Figure 1.7: Geometrical variables used for the calculations of the pressure distributions $P(r, \theta, t)$ for a planar circular transducer of radius a .

Again, ρ is the density of the medium and c is the speed of sound. The impulse function $\zeta(t, \delta)$ is generally approximated [15] by a Gaussian (Fig. 1.9(a)), which envelopes the intensity distribution, and is given by:

$$\zeta(t, \delta) = I(r, \theta, t) \exp \left(- \frac{(t - \delta)^2}{2\sigma^2} \right) \quad (1.4)$$

where σ is the pulse standard deviation. We consider that the beam is collimated by $\theta = \theta_a$. In our model only the corresponding interval $d\theta \approx 0.1^\circ$ is used that corresponds to the transducer lateral resolution zone (Fig. 1.9(b)). Hence

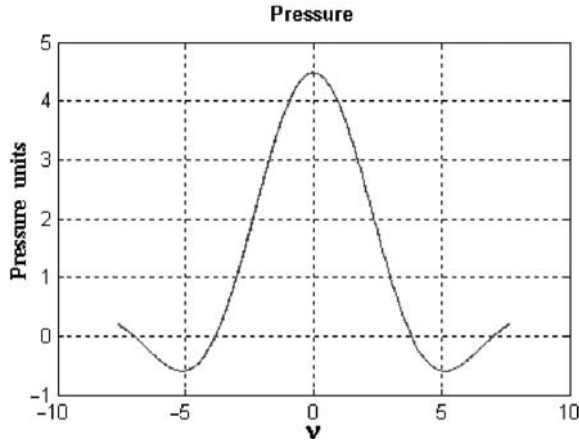


Figure 1.8: Transducer pressure distribution.

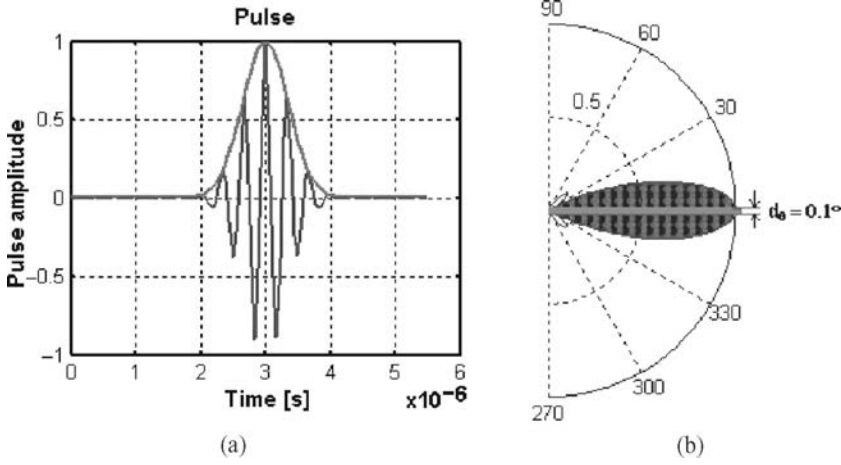


Figure 1.9: Typical ultrasound pulse and its Gaussian envelope (a). The transducer pattern radiation (b).

Eq. (1.2) in the transducer coordinate system is based on a discrete representation of the tissue of individual scatterer elements with given position and DBC with respect to the transducer coordinates given by:

$$S(R, \Theta, Z, t, \delta) = C_o \sum_{i=1}^N \frac{\sigma_i(R, \Theta, Z)}{|R_i|} \zeta(t, \delta) \quad (1.5)$$

where δ is given by Eq. (1.1), and $\zeta(t, \delta)$ is the impulse function given by Eq. (1.4). If we consider only the axial intensity contributions, C_o can be written as [14]:

$$C_o = \frac{\rho_o c k^2 v_o^2 A}{8\pi} \quad (1.6)$$

where A is the transducer area.

1.4 Principal Features of IVUS Data

1.4.1 Image Resolution

Resolution is the capacity of a technique or an instrument to separate two events or objects in time and/or space [14]. At the moment, much of the effort in the design of new transducers is centered in improving the spatial and the temporal resolution. Unfortunately, most of the medical applications demand that the

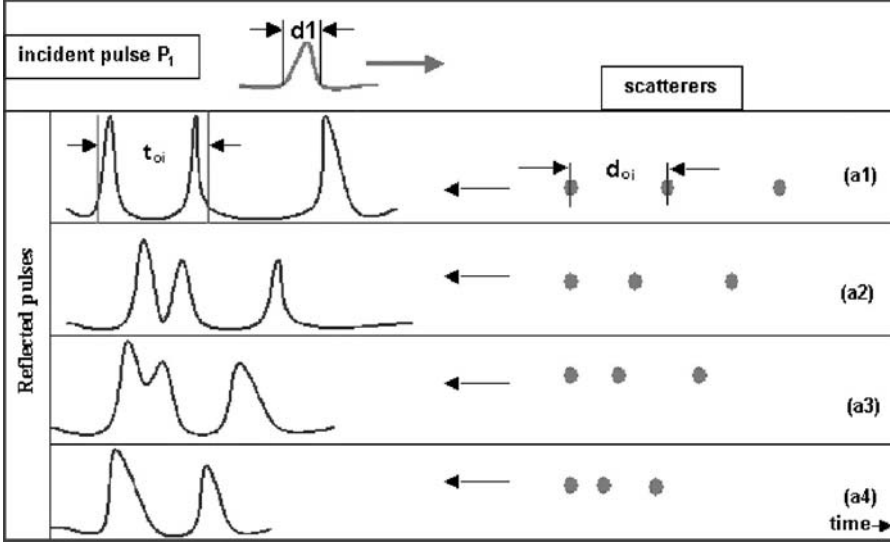


Figure 1.10: An ultrasound pulse P_1 that has width d_1 frontally affects a linear scatterer array placed at a distance d_{oi} .

transducers be smaller so that the resolution is increased, but this diminishes their capacity to explore greater tissues depth. For the IVUS techniques, the resolution plays a very important role since most of the structures to be visualized directly depend on these parameters.

1.4.1.1 Axial Resolution

Axial resolution is the capacity of an ultrasound technique to separate the spatial position of two consecutive scatterers through its corresponding echoes [13, 14, 16]. In Fig. 1.10 an ultrasound pulse P_1 that has a width d_1 frontally affects a linear scatterer array at a distance d_{oi} . Each one of the echoes forms a “train” of pulses temporally distanced according to the equation $t_{oi} = 2|R_i|/c$, R_i being the i th relative emitter/scatterer distance and c is the pulse propagation speed. The progressive distance reduction of the linear scatterers, given by $(a_1, \dots, a_4)$ (Fig. 1.10) and $(b_1, \dots, b_4)$ (Fig. 1.11), reduces the time interval between the maximums of the “trains” pulses. There exists a critical distance width d_i at which the pulses that arrive at the receiver are superposed, therefore, not being able to discriminate or separate individually the echoes produced by each scatterer. In Fig. 1.11 one can observe that the resolution can be improved by

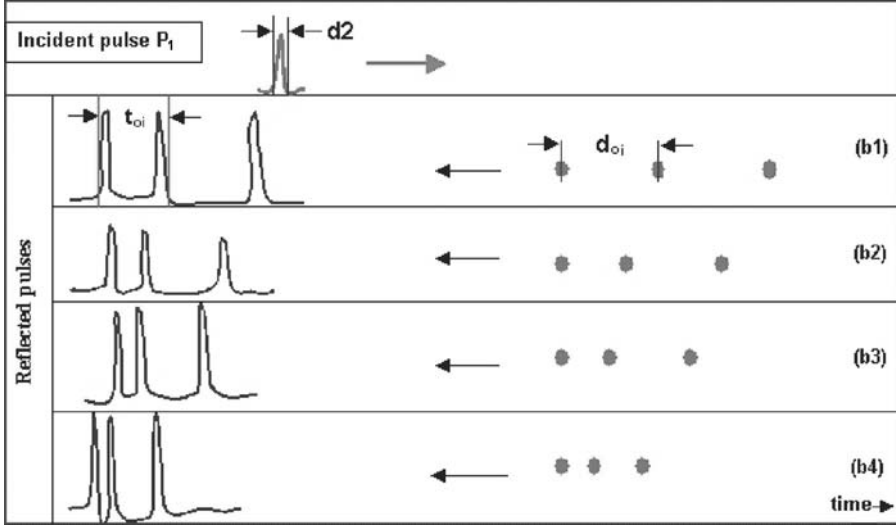


Figure 1.11: We can see that the progressive distance reduction of the linear scatterers, from $(a_1, \dots, a_4)$ (Fig. 1.10) to $(b_1, \dots, b_4)$ reduces the time difference between the maximums of the “train” pulses. The maximums can be separated reducing the pulse width from d_1 (Fig. 1.10) to d_2 , this is equivalent to an increase in the pulse frequency.

diminishing the pulse width d_t , which is equivalent to increasing the frequency of the emitted pulse. The axial resolution of this technique depends essentially on two factors: *ultrasound speed* c and *pulse duration* d_t . The functional dependency between the spatial resolution, the frequency, and the ultrasound speed propagation is given by:

$$d_r = cd_t = cT = \frac{c}{f} \quad (1.7)$$

where d_r is the axial resolution, c is the ultrasound speed for biological tissues, d_t is the pulse width, T is the period of ultrasound wave, and f is the ultrasound frequency. For IVUS, the typical values are: $c = 1540$ m/sec and $f = 30$ MHz, the axial resolution is approximately $d_r = 1540/(30 \times 10^6) = 0.05$ mm ≈ 50 μ m, and the relative error of the axial resolution is given by:

$$\frac{\Delta d_r}{d_r} = \left| \frac{\Delta c}{c} \right| + \left| \frac{\Delta f}{f} \right|$$

The axial resolution dependency versus the ultrasound frequency is shown in Fig. 1.12.

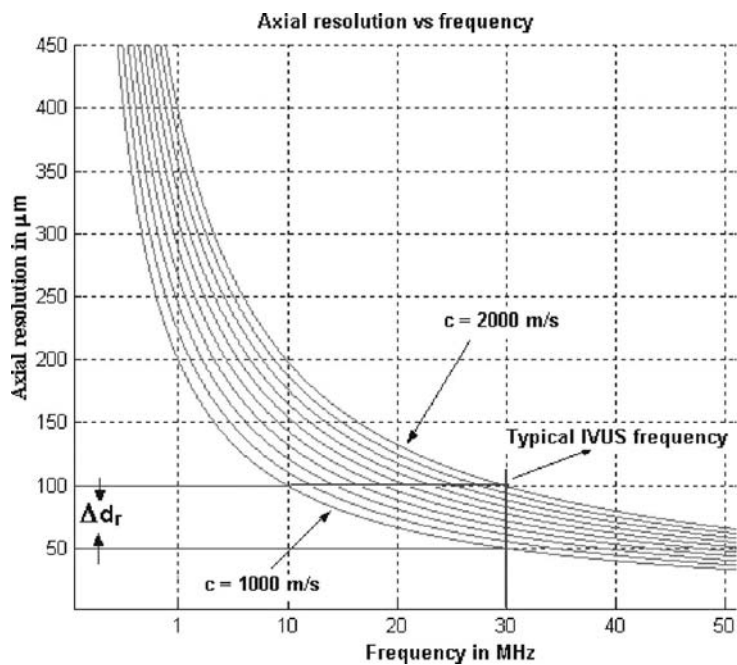


Figure 1.12: The functional dependency between the axial resolution and the ultrasound frequency for a rank of typical ultrasound speeds (see Table 1.1) in biological tissue. The typical IVUS (30 MHz) frequency as well as the tolerance in the axial resolution Δd_r are emphasized.

1.4.1.2 Angular Resolution

Angular resolution is the capacity to discern two objects or events located in the tangential direction [13, 14, 16] and depends on the *beam width*. The beam

Table 1.1: Sound speed in selected tissues [16]

Material	Sound speed (m/sec)
Fat	1460
Aqueous humor	1510
Liver	1555
Blood	1560
Kidney	1565
Muscle	1600
Lens of eye	1620
Average	1553

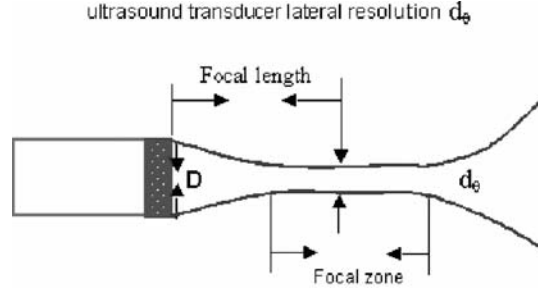


Figure 1.13: The focal length and the focal zone of an ultrasound transducer are indicated. The transducer lateral resolution d_θ is a function of its diameter D and the emission frequency f .

width depends on the transducer effective emission area (Fig. 1.13). Figure 1.14 shows the standard dimensions of a typical IVUS ultrasound transducer. The tangential or lateral resolution of an ultrasound emitter of diameter D having emission frequency f is given by:

$$d_\theta = 1.22 \frac{\lambda}{D}, \quad d_\theta = 1.22 \frac{c}{D} \frac{1}{f}$$

and the focal distance (focal length) F is given by:

$$F = \frac{1}{4} \frac{D^2}{\lambda}$$

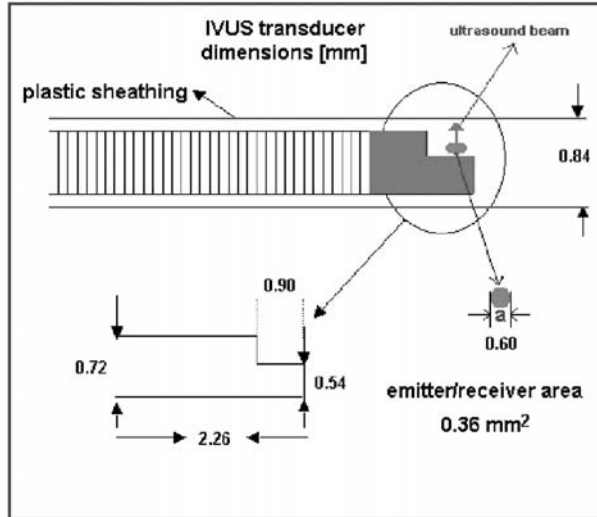


Figure 1.14: Typical IVUS transducer dimension used by Boston Sci.

where $\lambda = c/f$ and D is the transducer diameter. For a typical transducer of 30 MHz, $c = 1540$ m/sec and transducer dimensions given in Fig. 1.14, the lateral resolution is $d_\theta \approx 0.10^\circ$ and the focal length is $F = 2$ mm.

1.4.2 The Beam Intensity

The beam ultrasound intensity, as a function of the penetration depth and the ultrasound frequency, is given by [13, 14, 16]:

$$I(r) = I_0 \exp(-\alpha(N_\theta)rf) \quad (1.8)$$

where I_0 is the beam intensity at $r = 0$ and the coefficient α gives the rate of diminution of average power with respect to the distance along a transmission path [17]. It is composed of two parts, one (absorption) proportional to the frequency and the other (scattering) dependent on the particle size, or the scatterer number N_θ located along the ultrasound beam path (see Section 1.5.2). Since the attenuation is frequency dependent, a single attenuation coefficient only applies to a single frequency. The attenuation coefficient of ultrasound is measured in units of dB/cm, which is the logarithm of relative energy loss per centimeter traveled. In biological soft tissues, the ultrasound attenuation coefficient is roughly proportional to the ultrasound frequency (for the frequency range used in medical imaging). This means that the attenuation coefficient divided by the frequency (unit dB/MHz cm) is nearly constant in a given tissue. Typical soft tissue values are 0.5–1.0 dB/MHz cm. In our model we assumed that the attenuation coefficient α is only dependent on the scatterer number in the way beam. Figure 1.15 shows the beam intensity dependence on penetration depth for several typical frequencies used by IVUS.

1.4.3 Ultrasound Beam Sweeping Criterion

Let us explore a criterion that assures that all the reflected echoes reach the transducer before it moves to the following angular position. Let us define β as the ratio between transducer diameter D and arc length ϵ (Fig. 1.16):

$$\beta = \frac{D}{\epsilon}$$

where D is the transducer diameter and ϵ is the arc segment swept by the beam

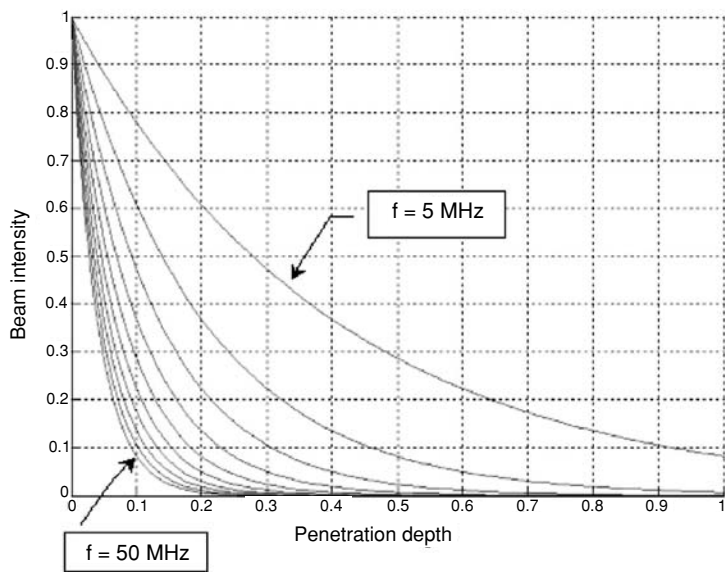


Figure 1.15: Ultrasound beam intensity versus the penetration depth for several frequencies (5–50 MHz).

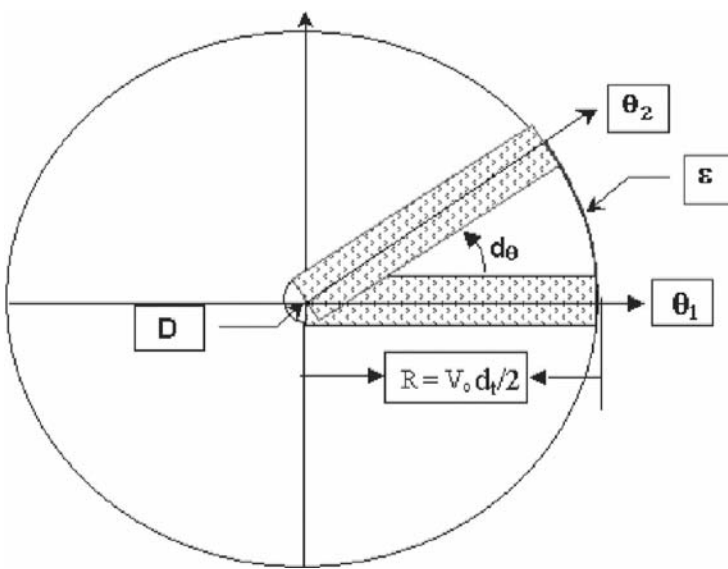


Figure 1.16: A rotatory transducer emits a radially focused beam. Angular positions θ_1 and θ_2 define a segment of arc S , which can be calculated from the speed of rotation and the speed of propagation of the ultrasound beam.

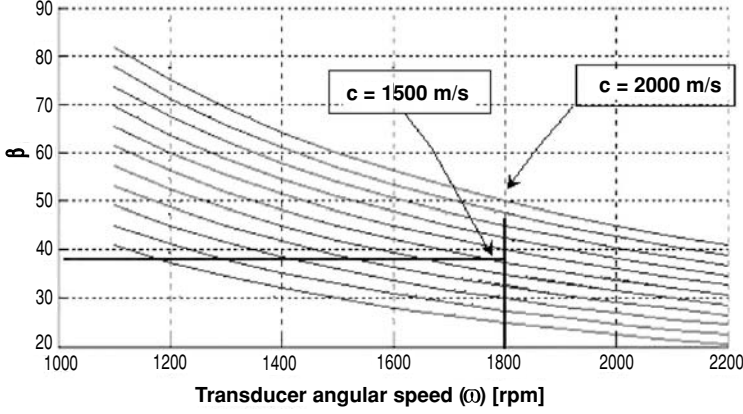


Figure 1.17: Functional dependence between parameter β and transducer angular speed ω .

between two angular consecutive positions. Note that:

$$d_\theta = \omega d_t, \quad d_t = 2\frac{R}{c}, \quad \epsilon = R d_\theta \quad (1.9)$$

Taking into account these definitions, β can be rewritten as:

$$\beta = \left(\frac{r}{R^2}\right)\left(\frac{c}{\omega}\right)$$

where r is the transducer radius, R is the maximum penetration depth, c is the ultrasound speed, and ω is the transducer angular speed. The parameter β implies that the transducer area is β times the sweeping area for the rotatory beam and the maximal depth penetration. This assures that a high percentage of echoes is received by the transducer before it changes to the following angular position. We can determine the parameter β by calculating the frequency at which the ultrasound pulse should be emitted. Figure 1.17 shows the functional dependence between parameter β and the transducer angular velocity for several typical velocities in biological tissues. We emphasize the typical IVUS transducer angular velocity. Figure 1.18 gives the relation between the sample frequency ($f_m = 1/d_t$) and the typical IVUS transducer angular velocity ω .

1.4.4 Determining the Scatterer Number of Arterial Structures

1) The *red blood cells* (RBCs) number swept by the ultrasound beam (Fig. 1.19) can be estimated by taking into account the plastic sheathing dimensions of

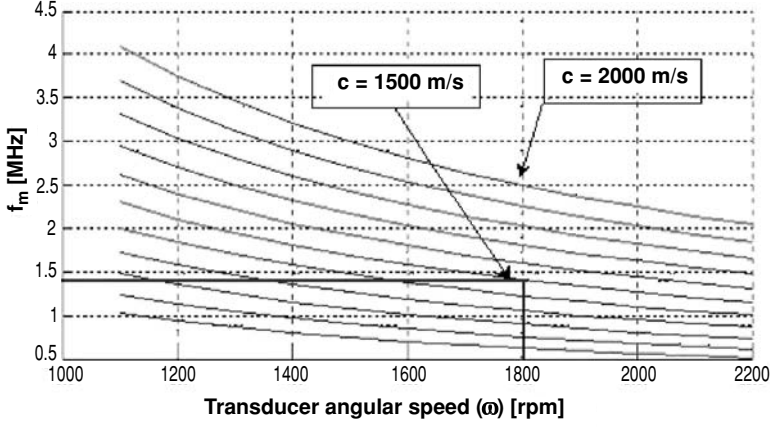


Figure 1.18: Functional dependence between the sample frequency and the transducer angular speed.

the transducer (Fig. 1.14) and the typical arterial lumen diameter. The scatterer number contained in a sweeping beam volume given by the difference between the sweeping lumen arterial volume V_a and the plastic sheathing transducer volume V_t :

$$V_b = V_a - V_t = \pi a(D^2 - D_M^2)/4 \quad (1.10)$$

where D and D_M are the arterial lumen and the sheathing transducer exterior

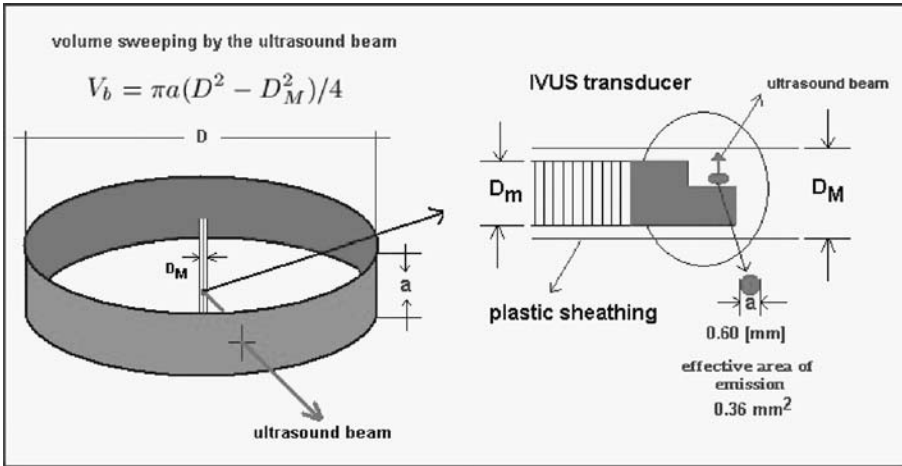


Figure 1.19: The scatterers volume for each arterial structure can be calculated by taking into account the total volume V_b swept by the ultrasound beam.

diameters respectively, and a is the effective emission diameter of the transducer. Typical arterial lumen diameter of coronary arteries is $D \approx 3$ mm [18, 19]. From Fig. 1.14 we can see that $D_M \approx 0.84$ mm and $a = 0.60$ mm. Using Eq. (1.10) we obtain the sweeping volume of the transducer beam approximately as $V_b \approx 3.91$ mm³. The RBCs can be approximated by spherical scatterers having a volume of 87 μm^3 [20], which corresponds to a radius of 2.75 μm (diameter, $d_g = 5.5$ μm). Considering a typical hematocrit concentration [21] of 35%, we can estimate the RBCs number by the beam sweeping volume. The RBCs sweeping volume is $V_{\text{RBC}} = 1.36$ mm³, and the typical human RBCs number is approximately $N \approx 4.1 \times 10^6$ cells/mm³ [21]. Thus, the RBCs number by the sweeping volume is $N_o \approx 5.61 \times 10^6$ cells. The maximal axial resolution at 40 MHz is approximately $d_r = 38$ μm , at which we can observe the order of $d_r/d_g \approx 7$ RBCs. If we take the scatterers as perfect spheres with radius d_r at maximal axial resolution, we would have scatterers of the order of 1.37×10^7 to be simulated. It is not possible to estimate this value for RBCs scatterers with a computer. In order to generate the number of scatterers possible to emulate, we generate scatterers groups namely “voxel” [11]. In Table 1.2, the most important numerical data used by this simulation model is summarized. The minimal structure dimensions that can be measured by an IVUS image at 40 MHz is $1/25$ mm/pixel ≈ 0.04 mm. We take this dimension to estimate the minimal

Table 1.2: Important features and the corresponding approximated values used in this simulation model

Feature	Approximated values
Arterial diameter	$D = 3$ [mm]
Sheathing transducer diameter	$D_M = 0.84$ [mm]
Transducer diameter	$a = 0.60$ [mm]
Sweeping volume by the beam	$V_b = 3.91$ [mm ³]
RBC volume	87 [μm^3]
Hematocrit concentration	35%
RBC volume by 35% of V_b	1.36 [mm ³]
Typical human RBC number	$N = 4.1 \times 10^6$ [cells/mm ³]
Maximal axial resolution at 40 MHz	$d_r = 38$ [μm]
IVUS image resolution	$(1/25) \approx 0.04$ [mm/pixel]
Minimal voxel volume	6.4×10^{-5} [mm ³]
Total RBC voxel	360 [voxels]
RBC voxel to be emulated	1.5×10^4 [voxels]

Table 1.3: An example of simulated values of arterial structures

k	Structure	N_k	$\bar{R}_k$ [mm]	η_k [mm]	$(DBC)\mu_k$ [m ²] $\times$ E - 6	σ_k [m ²] $\times$ E - 6
0	Transducer	475	0.59	0.05	7.2E-1	2.68E-2
1	Blood	6204	1.57	1.22	9.0E-2	9.48E-1
2	Intima	729	2.18	0.25	8.2E-1	2.86E-2
3	Media	150	2.38	0.35	3.3E-3	1.82E-1
4	Adventitia	25794	3.44	3.02	7.3E-1	2.71E-2

N_k is the scatterer number, $\bar{R}_k$ is the mean radial position, η_k is the radial deviation, μ_k is the backscattering cross section, and σ_k is the DBC deviation.

“voxel” volume. For the RBCs, $V_o = 0.04 \times 0.04 \times 0.04 \approx 6.4 \times 10^{-5} \text{ mm}^3$. The total number of RBCs per voxel is $N_t = V_o \times N \approx 360 \text{ cells/voxel}$. Now, we can calculate the total RBCs “voxel” number as $N_{\text{RBC}} = N_o/N_t \approx 1.5 \times 10^4 \text{ voxels}$ for the sweeping volume by the ultrasound beam. This “voxel” number is even computer intractable. Therefore, we must consider that the typical structure dimensions that can be measured by IVUS image are greater than 0.04 mm. A well contrasted image structure dimension by IVUS begins from 0.06 mm. Using these “voxel” dimensions, $V_o = 2.14 \times 10^{-4} \text{ mm}^3$, the total “voxel” number is $N_t \approx 880 \text{ cells/voxel}$, and the RBC “voxels” number is approximately $N_1 \approx 6200 \text{ voxels}$. An example of RBCs “voxel” number used in this simulation is given in Table 1.3.

(2) *The intima, media, and adventitia.* The numerical values necessary for the evaluation of the scatterer number for the intima, media, and adventitia were taken from results of Perelman *et al.* [22], which give the typical nuclear cells size l (μm) distribution for human cells. The “voxel” number for each layer was computed taking into account the typical dimensions of intima, media, and adventitia of a normal artery.

(3) *The voxel number for the sheathing transducer* was calculated taking into account the minimal scatterers that can be observed at maximal resolution when the frequency is fixed at 40 MHz, a typical IVUS frequency. From Figs. 1.14 and 1.19, the transducer sweeping volume is $V_t = \pi a(D_M^2 - D_m^2)/4$, where $a \approx 0.60 \text{ mm}$ is the transducer diameter, and $D_M \approx 0.84 \text{ mm}$ and $D_m \approx 0.72 \text{ mm}$ are the exterior and interior transducer sheathing diameters respectively. Using these dimensions, $V_t \approx 0.08 \text{ mm}^3$. The sheathing “voxel” number N_o can be calculated as $N_o = V_t/V_o$, where $V_t \approx 0.08 \text{ mm}^3$ is the sheathing volume by the beam and $V_o = 4\pi d_r^3/3$ is formed by the minimal spherical scatterers with radius

$d_r = c/f$ able to be measured when the frequency f and the ultrasound speed c are known. Taken typical values for $c = 1540$ m/sec and frequency of 40 MHz, $V_o \approx 2.39 \times 10^{-4}$ mm³, thus $N_o \approx 370$ “voxels.”

1.5 Simulation of IVUS Image

1.5.1 Generation of the Simulated Arterial Structure

Considering the goal of simulating different arterial structures, we can classify them into three groups: tissue structures, nontissue structures, and artifacts. The spatial distribution of the scatterer number with a given DBC, $\sigma(R, \Theta, Z)$ at point (R, Θ, Z) , has the following contributions:

$$\sigma(R, \Theta, Z) = A(R) + B(R, \Theta, Z) + C(R) \quad (1.11)$$

where $A(R)$, $B(R, \theta, Z)$, and $C(R)$ are the contributions of tissue structures, nontissue structures, and artifacts respectively.

1. *Tissue scatterers.* These are determined by the contribution of the normal artery structures, corresponding to *lumen*, *intima*, *media*, and *adventitia*. Figure 1.20 shows a k -layers spatial distribution of the scatterers for a simulated arterial image. These scatterers are simulated as radial Gaussian

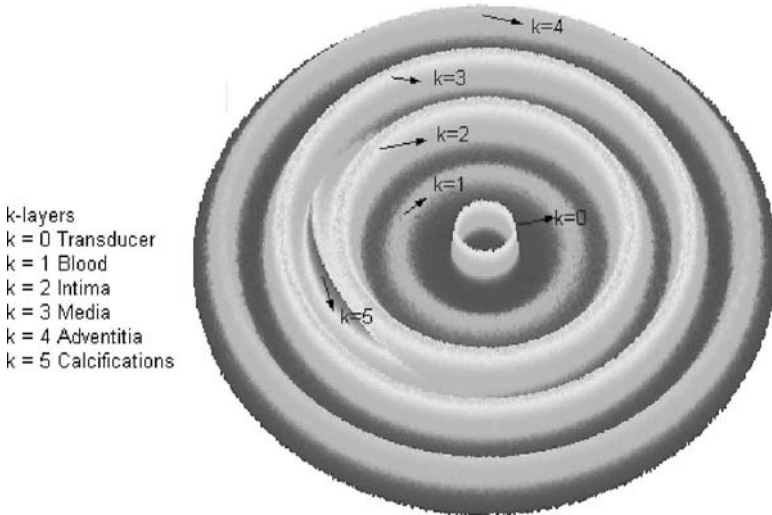


Figure 1.20: A plane of k -layers simulated artery. The scatterer numbers are represented by the height coordinate in the figure.

distributions [23] centered in the average radius R_k and having standard deviation η_k corresponding to each arterial structures. Tissue scatterers are represented by:

$$A(R) = \sum_{k=1}^{k_o} \frac{a_k}{\eta_k} \exp \left(- \frac{(R - R_k)^2}{2\eta_k^2} \right) \quad (1.12)$$

where a_k is the maximal number of scatterers at $R = R_k$, k is the k th radial simulated tissue layer, and R_k is the radial layer average position.

2. *Nontissue scatterers.* These contributions can be made by structures formed by spatial calcium accumulation, which are characterized as having greater DBC density than the rest of the arterial structures. They are simulated by a Gaussian distribution in the radial, angular, and longitudinal arterial positions of the simulated structure:

$$B(R, \Theta, Z) = \sum_{l=1}^{l_o} \sum_{m=1}^{m_o} \sum_{n=1}^{n_o} \frac{b_l c_m d_n}{\beta_l \gamma_m \nu_n} F(R, \Theta, Z)$$

$$F(R, \Theta, Z) = \exp \left(- \frac{1}{2} \left(\frac{(R - \bar{R}_l)^2}{\beta_l^2} + \frac{(\Theta - \bar{\Theta}_m)^2}{\gamma_m^2} + \frac{(Z - \bar{Z}_n)^2}{\nu_n^2} \right) \right)$$

where (l, m, n) correspond to the radial, angular, and longitudinal axes directions, (l_o, m_o, n_o) are the structures number in radial, angular, and longitudinal directions, (b_l, c_m, d_n) are the scatterer numbers that have a maximum at $R = R_l$, $\Theta = \Theta_m$, and $Z = Z_n$, $(\beta_l, \gamma_m, \nu_n)$ are the radial, angular, and longitudinal standard deviations, and $(\bar{R}_l, \bar{\Theta}_m, \bar{Z}_n)$ are the radial, angular, and longitudinal average positions.

3. *Artifacts scatterers.* In our model we consider the artifact caused by the sheathing transducer:

$$C(R) = \frac{a_o}{\alpha_o} \exp \left(- \frac{(R - R_o)^2}{2\alpha_o^2} \right)$$

where a_o is the scatterers number that has a maximum at $R = R_o$, α_o is the artifact standard deviation, and R_o is the artifact radial average position.

1.5.2 1D Echogram Generation

To obtain a 1D echogram, an ultrasound pulse is generated in accordance with Eq. (1.4) and emitted from the transducer position. The pulse moves

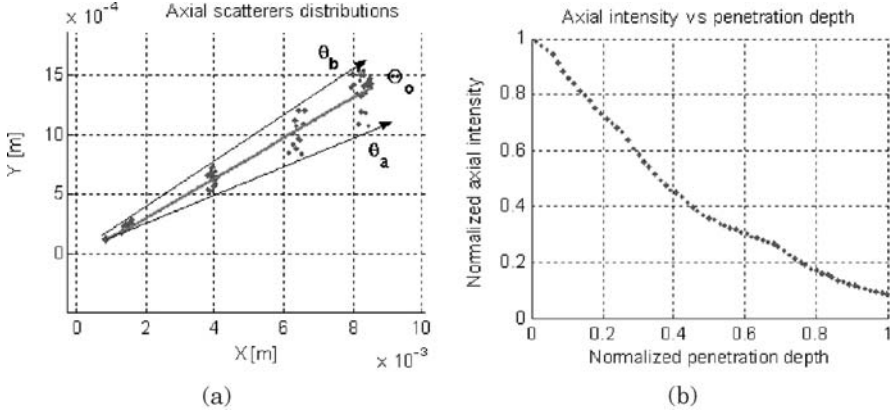


Figure 1.21: The 1D echogram is obtained by fixing the angular position $\Theta_0 = \bar{\Theta}$ of the ultrasound beam (a). The total signal $S(t)$ is only generated by the scatterers N_{Θ} located at an angular position $\Theta_a \leq \Theta_0 \leq \Theta_b$. The intensity distribution decreases with the depth penetration and the scatterers numbers N_{Θ} through the beam path (b).

axially through scatterers (Fig. 1.21(a)) and its intensity distribution decreases (Fig. 1.21(b)) with the penetration depth and the scatterers numbers in the ultrasound path given by Eq. (1.8). The echo amplitude is registered by the transducers (Fig. 1.22) as a signal function of time $S(t)$ (Eq. 1.13). The value is transformed to penetration depth replacing $t = x/c$ and normalized to gray scale. The spatial distribution of cross-section scatterers, σ , is generated by

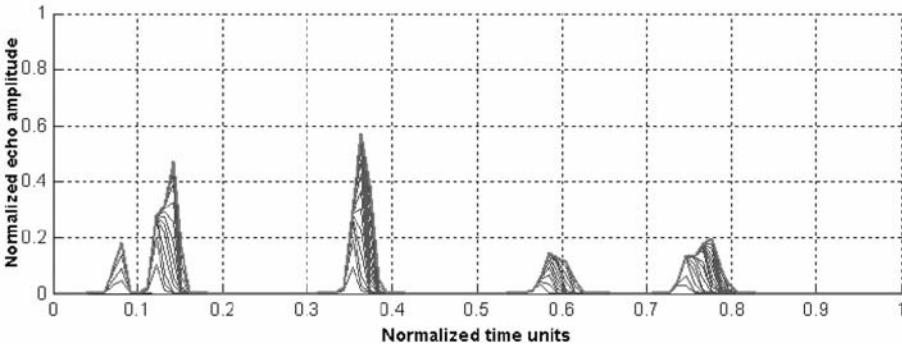


Figure 1.22: The corresponding echoes are finally transformed to normalized echo amplitude and then to gray-level scale versus time or penetration depth.

using Eq. (1.11). Figure 1.21 shows the simulations of N scatterers located in $(R_i, \Theta_a \leq \Theta_j \leq \Theta_b)$:

$$S(t, \Theta_o) = \sum_{i=1}^{N_R} \sum_{j=1}^{N_{\Theta_i}} \frac{\sigma(R_i, \Theta_o \pm \Theta_j) \zeta(t, \delta_i)}{|R_i|} \quad (1.13)$$

$$S(t, \Theta_o) = C_o \sum_{i=1}^{N_R} \sum_{j=1}^{N_{\Theta_i}} \frac{\sigma(R_i, \Theta_o \pm \Theta_j)}{|R_i|} \exp\left(\frac{-(t - \delta_i)^2}{2\sigma^2}\right) \sin(\omega t - \delta_i)$$

where $\Theta_o = (\Theta_a + \Theta_b)/2$, C_o defines the transducer constant parameters, and N_{Θ_i} is the total scatterers number at the angular position $\theta_a \leq \Theta \leq \theta_b$ for a radial position R_i . The sum only operates on the scatterers located in the angular position $\theta_a \leq \Theta \leq \theta_b$ that is the focal transducer zone (Figs. 1.9(b) and 1.13). Therefore, N_{Θ} is the total scatterers number in this region. Equation (1.13) can be written as a function of the penetration depth, replacing $t = x/c$. Equation (1.13) can be rewritten on gray-level scale as:

$$S(t, \Theta_o) = \frac{256}{\max(S(t))} C_o \sum_{i=1}^{N_R} \sum_{j=1}^{N_{\Theta_i}} \frac{\sigma(R_i, \Theta_o \pm \Theta_j)}{|R_i|} \exp\left(\frac{-(t - \delta_i)^2}{2\sigma^2}\right) \sin(\omega t - \delta_i) \quad (1.14)$$

where $\delta_i = 2R_i/c$ and $S(x)$ is the 1D echogram generated by a set of N_{Θ} scatterers located in $(R_i, \Theta_a \leq \Theta_i \leq \Theta_b)$. The overall distribution backscattering cross-section $\sigma_i(R_i, \Theta_i \pm \delta\Theta)$ is given by Eq. (1.11).

1.5.3 2D Echogram Generation

The procedure to obtain the 2D simulated IVUS is the following: A rotatory transducer with angular velocity ω (Fig. 1.23(a)) is located at the center of the simulated arterial configuration given by Eq. (1.11). The transducer emits an ultrasound pulse radially focused at frequency f_o along angular direction θ_1 (Fig. 1.23(a)). The pulse progressively penetrates each one of the layers of the simulated arterial structure according to Eq. (1.15). Each one of the layers generates a profile of amplitude or echoes in time, which can be transformed into a profile of amplitude as a function of the penetration depth (Fig. 1.23(b)). Therefore, the depth can be calculated using Eq. (1.1). As the penetration depth is coincident with the axial beam direction, the radial coordinate R is thus determined. This procedure is repeated n times for angles, $(\theta_1, \dots, \theta_n)$ and the 2D image is generated. The generated echo profiles are transformed to a polar

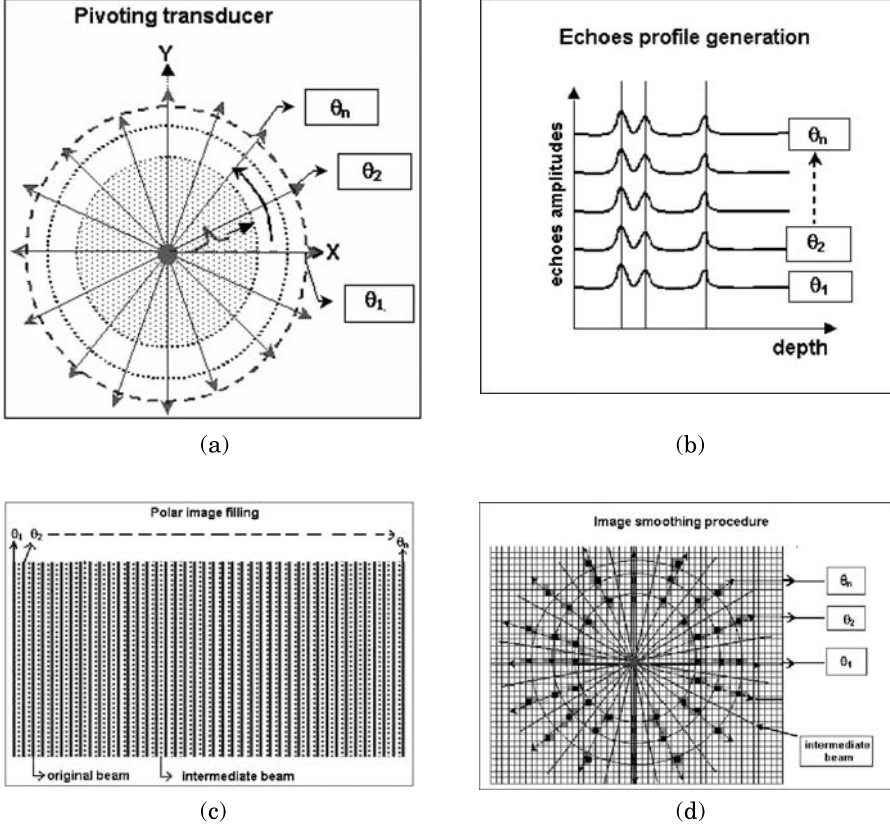


Figure 1.23: The transducer emits from the artery center (a), echo profile transformed into penetration depth (b), the echo profiles are transformed to a polar image (c), and empty pixels filled and the final IVUS image is smoothed (d).

image, and the intermediate beams are computed (Fig. 1.23(c)). The image is transformed to Cartesian form and the empty pixels are filled (Fig. 1.23(d)).

Using the ultrasound reflected signal $S(t, \Theta)$ for a finite set of N reflecting scatterers with coordinates (R, Θ, Z) and spatial distribution of the differential backscattering cross-section, $\sigma(R, \Theta, Z)$, the 2D echo signal $S(t, \Theta)$ can be written as:

$$S(t, \Theta) = C_o \sum_{i=1}^{N_R} \sum_{j=1}^{N_{\theta_i}} \frac{\sigma(R_i, \Theta \pm \theta_j) \zeta(t, \delta_i)}{|R_i|} \quad (1.15)$$

where $S(t, \Theta)$ is the temporally generated signal by a set N_R of scatterers, which are localized in angular position θ , $\theta \in [\theta_a, \theta_b]$, N_{θ_i} is the total scatterers number

in the angular position $\theta_a \leq \bar{\Theta} \leq \theta_b$ for a radial position R_i . We consider two forms of $\bar{\Theta}$:

- with no uniform distributed scatterers:

$$\bar{\Theta} = (\theta_a + \theta_b)/2$$

- with uniform distributed scatterers:

$$\bar{\Theta} = \frac{1}{N_R} \sum_{j=1}^{N_R} \Theta_j$$

1.5.4 Final Image Processing

The actual image obtained with only the original beams is very poor; we must explore several smoothing procedures to improve the image appearance. The procedures to obtain the final simulated image are as follows:

1. The echoes are obtained by the pivoting transducer (Fig. 1.23(a)).
2. Each echo profile is ordered according to the angular position (Fig. 1.23(b)).
3. The original image is transformed to a polar form (Fig. 1.23(c)).
4. Secondary beams are computed between two original neighboring beams (Fig. 1.23(c)).
5. The image is smoothed by a 2×2 median filter.
6. The image is again transformed to Cartesian form. As a result of this transformation, a significant number of pixels will be empty (Fig. 1.23(d)).
7. The empty pixels are filled in a recursive way form, using an average of the eight nearest neighbors (Fig. 1.23(d)).
8. An image reference reticle is added and a Gaussian filter is applied.

Figure 1.24 shows the scatterers distribution for a concentric arterial structure and an axial ultrasound beam position (a), and its corresponding echo profiles (b). Each axial echo is positioned by an angular position (c). In this way, the 2D echogram is constructed (d). The procedure of image smoothing is described in Section 1.5.4.

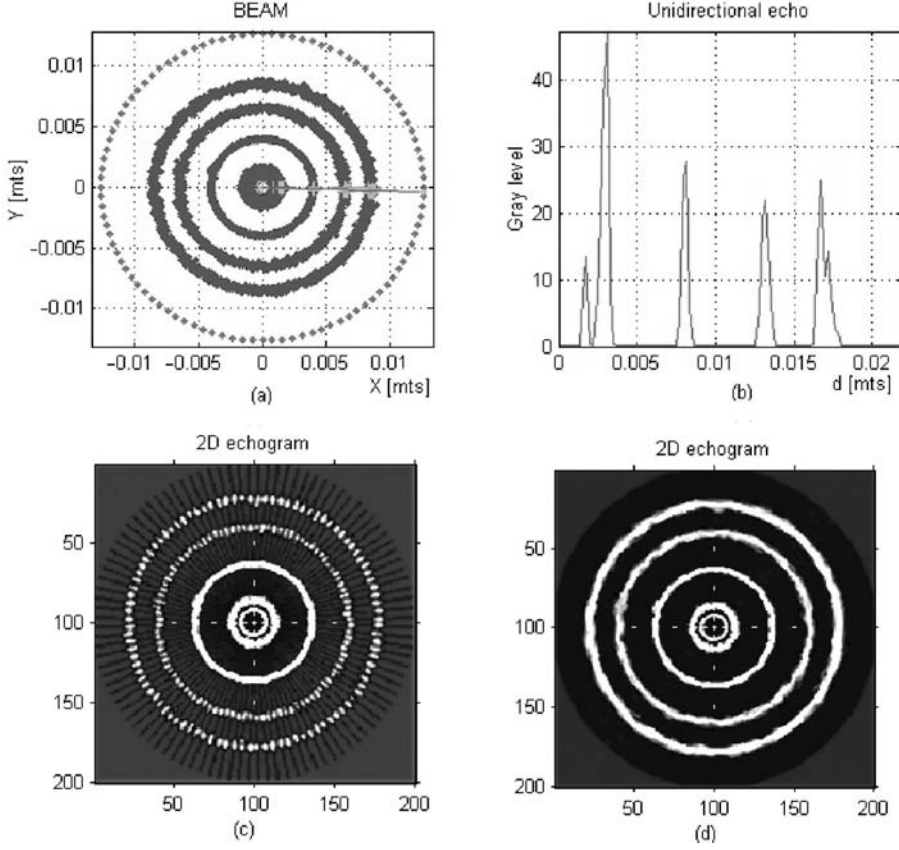


Figure 1.24: The scatterers distribution (a), the corresponding 1D echoes (b), 2D echogram is constructed (c), and the image is smoothed (d).

1.6 Validation of the Image Simulation Model

Once the generic basic model of IVUS image formation is defined, we need to compare it to real images contrasting expert opinion to test its use. For this purpose, we defined procedures to extract quantitative parameters that permit the measurement of the global and local similarities of the images obtained. The main goal of this simulation is to give a general representation of the principal characteristics of the image. The comparison of real and simulated images should be done on the global image descriptors. We concentrated on the distribution of the gray levels. Data such as transducer dimensions (Fig. 1.14), the catheter as well as the reticle locations, operation frequency, band width,

and original and secondary beam number used for the simulation are standard values obtained from Boston Sci. [24]. However, the optimal values of frequency and attenuation coefficient are obtained by the cross validation procedure [23]. The dimensions, scatterer number, and the backscattering cross-section of the simulated arterial structures were obtained from different literature [7, 10, 11, 19, 22, 24]. Typical values of the RBCs “voxel” numbers took into account the typical hematocrit percentage [11] (Section 1.4.4). Instrumental and video noise has been incorporated into the simulated image, due to electronics acquisition data, and the acquisition and processing to the video format.

The zones of greater medical interest (lumen, lumen/intima, intima/media, and media/adventitia) were simulated for several real IVUS images. The smoothing image protocol is not known so that the corresponding tests were done until the maximal similarity to the real images was found, based on the use of three progressive methods. (1) The empty pixels are filled using the average of eight neighbors, (2) a median filter is used, and (3) a Gaussian filter is applied in order to find the noise reduction. The quantitative parameters used for the image comparison were directed for global and local image regions, and are described below.

1. *Gray-level average projections* p_x and p_y , that is horizontal and vertical image projections, are defined for an $m \times n$ image I as [25]:

$$p_{x(i)} = \frac{1}{m} \sum_{j=1}^m I_{ij}, \quad p_{y(j)} = \frac{1}{n} \sum_{i=1}^n I_{ij} \quad (1.16)$$

2. We define a global linear *correlation* between real (x) and simulated (y) data as follows:

$$y = mx + b \quad (1.17)$$

where m and b are the linear correlation coefficients.

3. *Contrast to noise ratio signal* (CNRS) as figure of merit, defined as [26]:

$$CNRS = \frac{(\mu_1 - \mu_2)^2}{\sqrt{\sigma_1^2 + \sigma_2^2}} \quad (1.18)$$

where μ_1 , μ_2 , σ_1 , and σ_2 are the mean and the standard deviations inside the ROIs.

1.6.1 Scatterer Radial Distribution

The radial scatterer distribution is an important factor for a good image simulation. The scatterers under consideration in this simulation are: the transducer sheath, blood, intima, media, and adventitia. We can obtain the arterial structure configuration from an emulated form and from a real validated IVUS image. For the study of the synthetic images, we have used two procedures:

1. *Standard data.* Typical geometric arterial parameters and their interfaces such as lumen/intima, intima/media, and media/adventitia are obtained from standard literature.
2. *Validated data.* Geometrical parameters are obtained from manually segmented IVUS images.

In order to investigate the image dependencies of IVUS parameters (frequency, attenuation coefficient, original beam number, secondary beam, and smoothing procedures), we have used a standard data procedure, using a concentric scatterer distribution for this modality. To compare simulated images to real data, we use manually segmented real images, which correspond to the validated data procedure. In manually delineated structures of IVUS images, we extract the position radius R_k of lumen, intima, media adventitia, and transducer sheath. Figure 1.25 shows typical 2D spatial scatterer distributions obtained from standard procedure for the most important arterial structures and the scatterer artifact caused by the transducer sheath.

The radial scatterer distributions play a crucial role in the definition of the IVUS images because they define the ultrasound attenuation in the axial direction. Medical doctors have special interest in gray-level transition in the interface of two media. For instance, the lumen/intima transition defines the frontiers of the lumen. These transitions can only be found through a good radial scatterer distribution.

The radial scatterers distribution of the typical arterial structures and the transducer sheath are shown in Fig. 1.26.

1.6.2 DBC Distribution

The k -layers DBC_k values for a typical simulated arterial structure are shown in Figs. 1.27 and 1.28 where the count of scatterers of each tissue is shown as

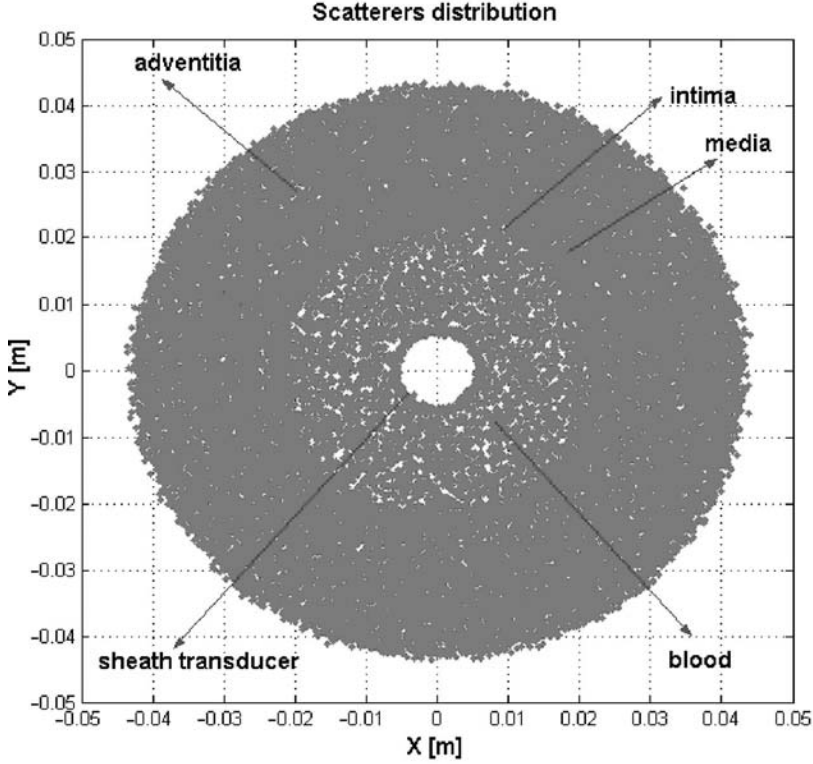


Figure 1.25: Typical concentric 2D scatterer distribution for the most important simulated arterial structures (blood, intima, media, and adventitia) and the scatterer artefact generated by the transducer sheath.

a function of the cross-section of scatterers. The numerical values are given in Table 1.3 [27].

1.6.3 IVUS Image Features

1.6.3.1 Spatial Resolution

A good spatial resolution gives the possibility of improving the visualization of the lumen/intima transition and studying the structures, which gives important information for medical doctors. Typical numerical parameters such as *scatterers number* N_k , *k-layer average radial position* $\bar{R}_k$, its standard deviation η_k , the DBC *k-layer mean* μ_k , and its standard deviation σ_k are given in Table 1.3. The typical IVUS parameters used in this simulation are given in

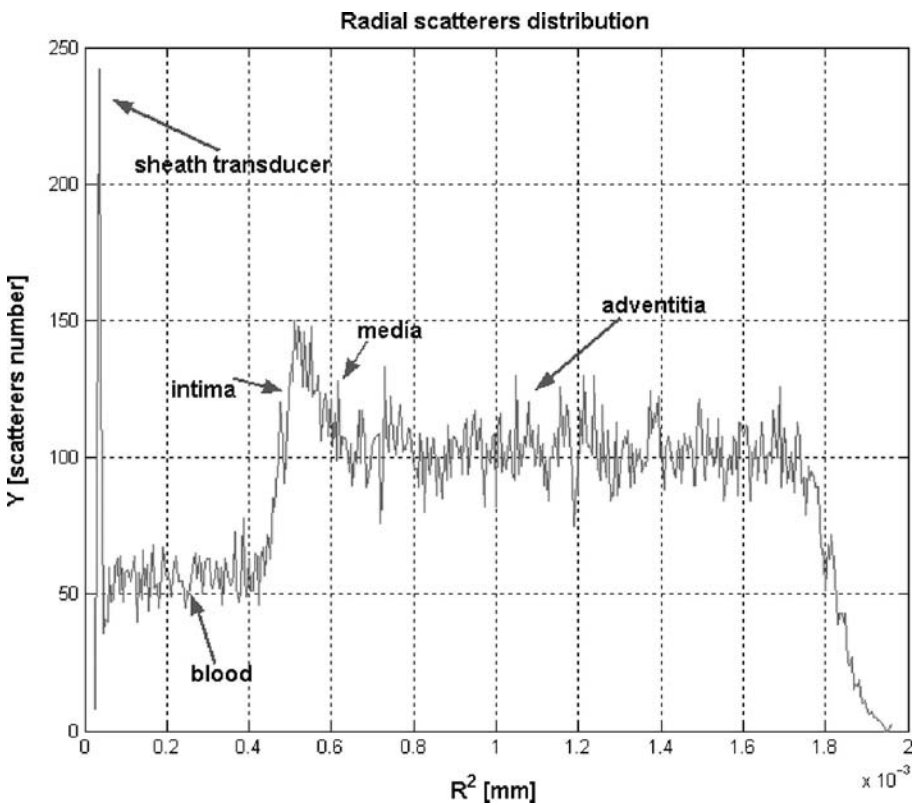


Figure 1.26: Radial scatterer distribution for the arterial structure: blood, intima, media, adventitia, and the transducer sheath.

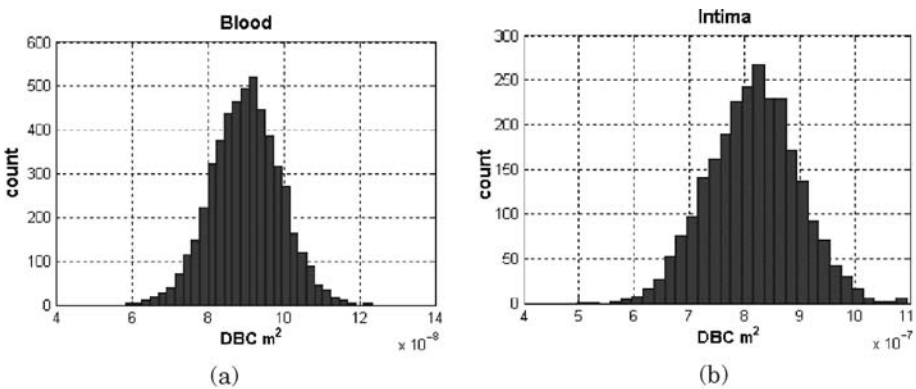


Figure 1.27: DBC distributions of simulated arterial structures: blood (a) and intima (b).

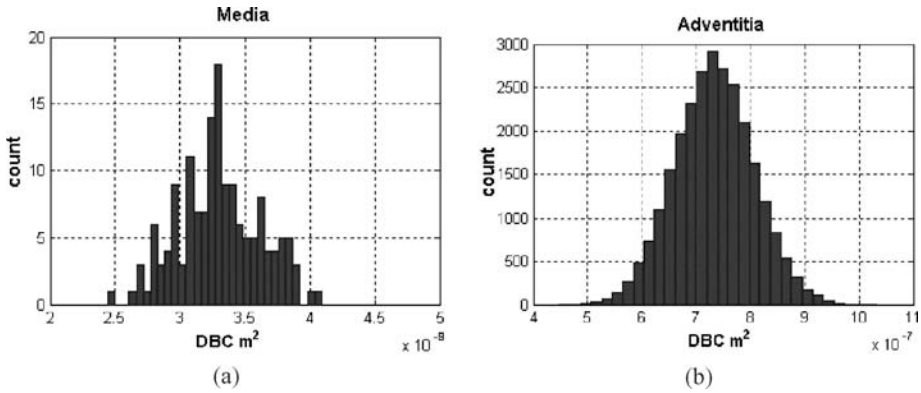


Figure 1.28: DBC distributions of simulated arterial structures: media (a) and adventitia (b).

Table 1.4. The typical cell nuclear size was obtained by Perelman *et al.* [22]. In Fig. 1.29 we can observe the dependency of axial resolution and the ultrasound frequency. To illustrate this, four IVUS simulated images are shown. *Low frequency* ranging from 10 to 20 MHz corresponds to an axial resolution from 154 to 77 μm , and *intermediate frequency* from 20 to 30 MHz gives axial resolution from 77 to 51 μm . In these cases, it is possible to visualize accumulations around 100 RBCs. *High frequency* from 30 to 50 MHz leads to 51–31 μm of axial resolution. Moreover, it is now possible to visualize accumulations of tens of RBCs. The IVUS appearance improves when the frequency increases, allowing different structures and tissue transition interfaces to be better detected.

Table 1.4: Typical IVUS simulation magnitudes

Parameter	Magnitude
Ultrasound speed	1540 m/sec
Maximal penetration depth	2E – 2 m
Transducer angular velocity	1800 rpm
Transducer emission radius	3E – 4 m
Attenuation coefficient α	0.8 dB/MHz cm
Ultrasound frequency	10–50 MHz
Beam scan number	160–400
Video noise	8 gray level
Instrumental noise	12.8 gray level
Beta parameter	$\beta = 38.5 \text{ ad}$

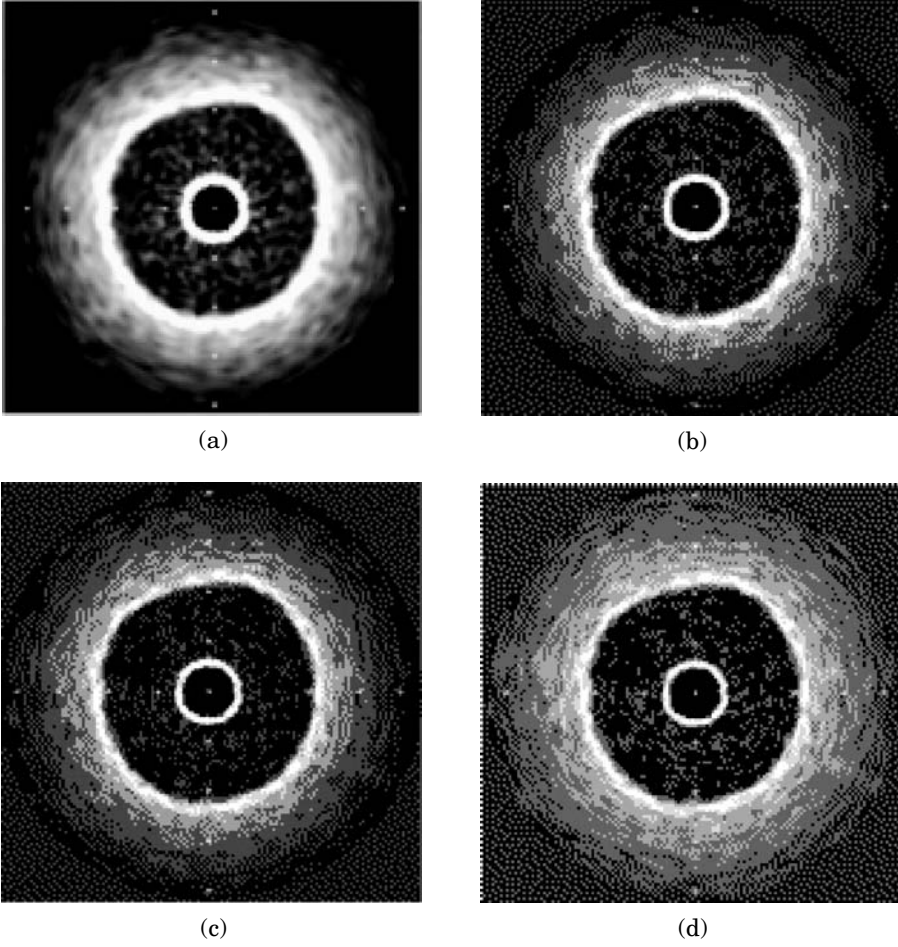


Figure 1.29: Synthetic images generated by low frequency: 10 MHz (a) and 20 MHz (b), intermediate frequency of 30 MHz (c), and high frequency of 50 MHz (d).

1.6.3.2 Optimal Ultrasound Frequency

In order to validate our model, we compare synthetic to real images. We generated synthetic images for a great rank of frequency and used the cross-validation method [23] to find the most similar image to the real one generated using Boston Sci. equipment at 40 MHz frequency. The sum square error (SSE) from the real to the simulated images for each ultrasound simulated frequency is computed. Figure 1.30(a) shows the SSE versus ultrasound frequency. The optimal frequency

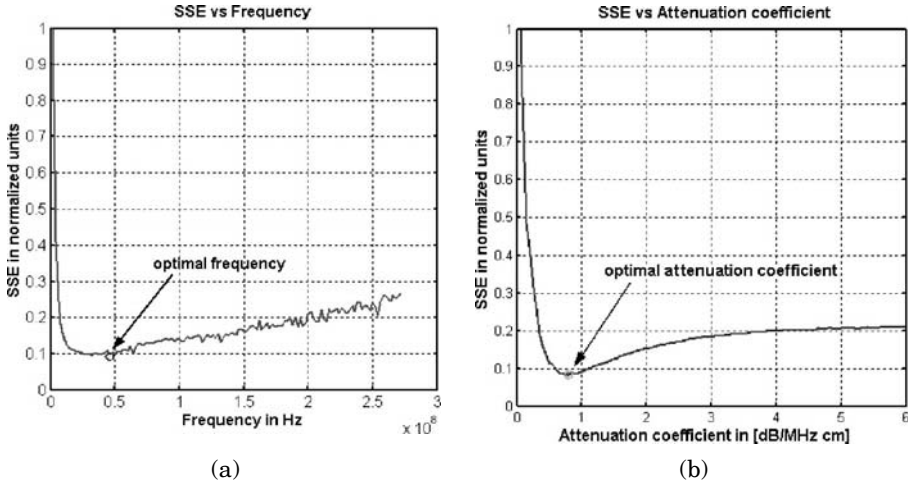


Figure 1.30: The optimal ultrasound simulation frequency $f_o \approx 46$ MHz (a) and the optimal attenuation coefficient (b) $\alpha \approx 0.8$ dB/MHz cm are obtained by the cross validation method.

is located in the interval 40–50 MHz. Note that the central frequency of Boston Sci. equipment is 40 MHz; therefore, it can be considered as evidence to show the correctness of the method.

1.6.3.3 Optimal Attenuation Coefficient

We have emulated synthetic IVUS images with different attenuation coefficients; the optimal attenuation coefficient was tested by applying the cross validation method of the synthetic images versus the real images. Figure 1.30(b) shows SSE versus attenuation coefficient α ; the optimal attenuation coefficient obtained was 0.8 dB/MHz cm. There is a range of suboptimal attenuation coefficient values for a fixed ultrasound frequency due to the great axial variability of scatterers. However, the attenuation coefficient can be taken as constant for each simulated region [28]; however, in the transition zones (lumen/intima, intima/media, and media/adventitia) the attenuation gives great variability. For this reason, we must average the attenuation coefficient value. It is very important to confirm that the optimal frequency is approximating the standard central ultrasound frequency of 40 MHz and that the attenuation coefficient is near the standard values of biological tissues, which ranges from 0.5 to 1 dB/MHz cm. This result can be used in different ways: first, to check the used simulation parameters in

the case of ultrasound frequency and second to find structures of interest when the attenuation coefficient is known.

1.6.3.4 The Beam Number Influence

Figure 1.31 shows the appearance of several simulated IVUS images when the original and intermediate beam numbers are changed. We obtained the best appearance when the original beam number was 80 and the secondary beam number was 240. In total, 320 beams were used by the simulation. We can see that the IVUS appearance in the tangential direction is significantly affected by

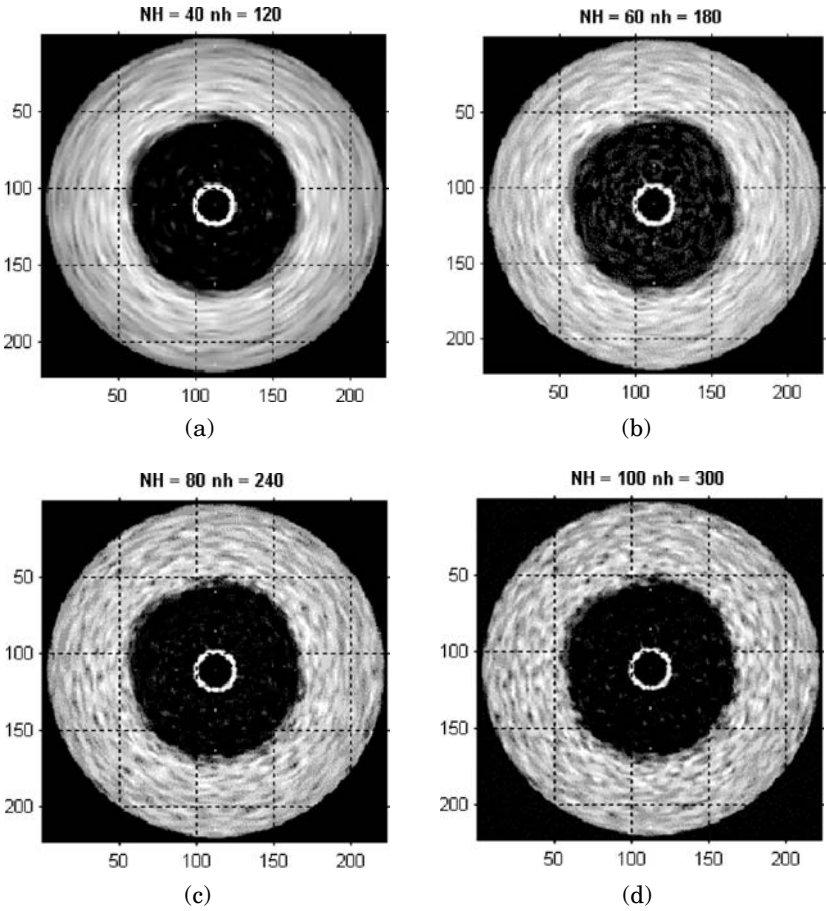


Figure 1.31: Different combinations of original (NH) and intermediate (nh) beams yield different IVUS appearances.

the beam number change. The total number of beams for the standard IVUS equipment is normally between 240 and 360 beams [24].

1.6.4 Real versus Simulated IVUS

In order to compare the real and simulated IVUS images, we have generated 20 synthetic images with morphological structures corresponding to the structures of a set of real images. We have used a real IVUS image with manually delimited lumen, intima, and adventitia to obtain the average radius location $\bar{R}_k$ for each arterial structure. We applied the optimal frequency of 46 MHz and attenuation coefficient of 0.8 dB/MHz cm. Figure 1.32(a) shows an IVUS real image of right coronary artery, obtained with a 40 MHz Boston Sci. equipment. Figure 1.32(b) shows a simulated image obtained at the optimal ultrasound simulation frequency of 46 MHz. In the real image, we can observe a guide zone artifact (12 to 1 o'clock) due to the presence of guide; this artifact will not be simulated in this study. The horizontal ECG baseline appears as an image artifact on the bottom of the real image. The global appearance of each image region (lumen, intima, media, and adventitia) and its corresponding interface transitions (lumen/intima, intima/media, and media/adventitia) are visually well contrasted, compared to the real image. A good quantitative global measure for comparison

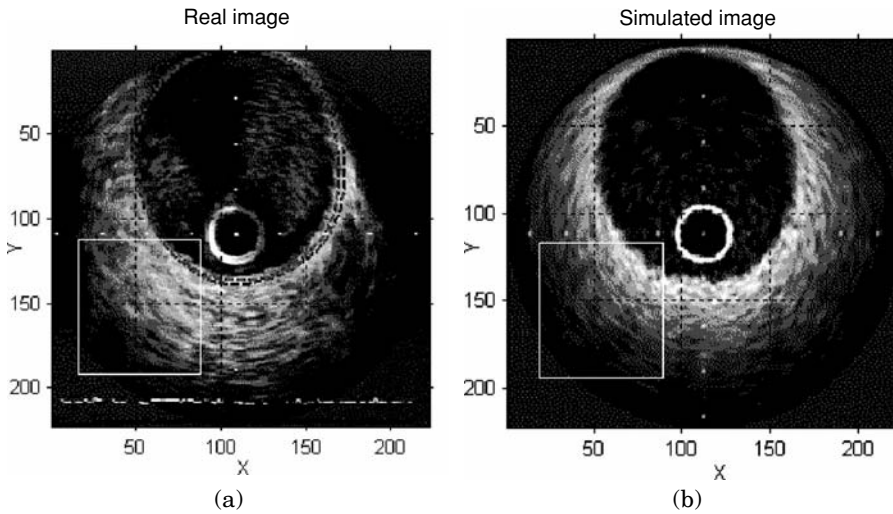


Figure 1.32: Real (a) and simulated (b) IVUS images segmentation. ROIs are given as squares. Manual segmentation of the vessel is given in (a).

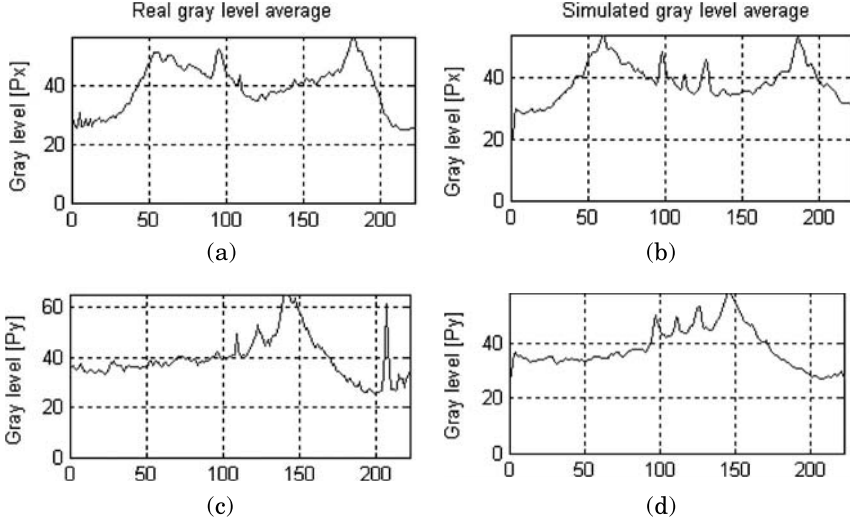


Figure 1.33: Horizontal ((a) and (b)) and vertical directions ((c) and (d)) gray-level profile average projections, from real (Fig. 1.32(a)) and simulated (Fig. 1.32(b)) IVUS images.

is the average gray-level projection that allows a simple form to find the main image correlated characteristics in an 1D shape gray-level profile. Gray-level baseline, video noise, instrumental noise, reticle influence, and the main gray-level distribution coming from the main arterial structures are roughly visible from the gray-level average projection. The average gray-level projection gives a global measure of the similarity between real and simulated images. The similarity measured can be computed, for example, by the local attenuation coefficient of the projection profile of each ROI [28]. Figure 1.33 gives the projections in the horizontal and vertical directions for the real (Fig. 1.32(a)) and simulated (Fig. 1.32(b)) IVUS images. The correlation coefficients m and b (Fig. 1.34) for the gray-level average projection in the horizontal ($m = 0.63$, $b = 13.53$) and vertical ($m = 0.75$, $b = 9.07$) directions show a positive correlation between the real and simulated data. Figure 1.35 shows two selected regions of interest of the real (Fig. 1.32(a)) and simulated (Fig. 1.32(b)) images. We can see a good gray-level distribution and a soft gray-level decay from the center to the peripheries of the IVUS image, produced by the inverse relation between the ultrasound intensity and the penetration depth. The other reason is that the normal attenuation is caused by the scattering intensity given by the tissue impedance. Figure 1.36

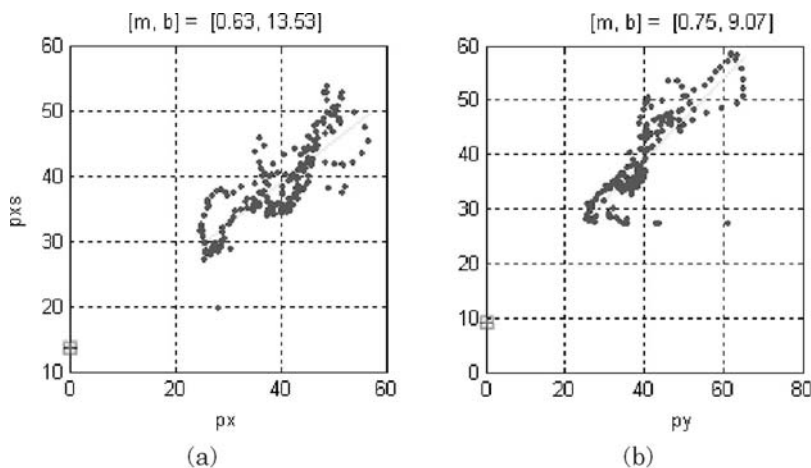


Figure 1.34: Horizontal correlation using (a) versus (b) from Fig. 1.33 and the vertical global correlation using (c) versus (d) from the same figure.

shows gray-level average projections in the vertical direction ((a) and (c)) and the horizontal direction ((b) and (d)) of the selected ROIs from Figs. 1.32(a) and (b). The linear correlation coefficients m and b (Fig. 1.37) for the gray-level average projection in the horizontal direction ($m = 0.87$, $b = 4.91$) and vertical direction ($m = 0.85$, $b = 5.79$) show a significant gray-level correspondence between the real and simulated ROIs image.

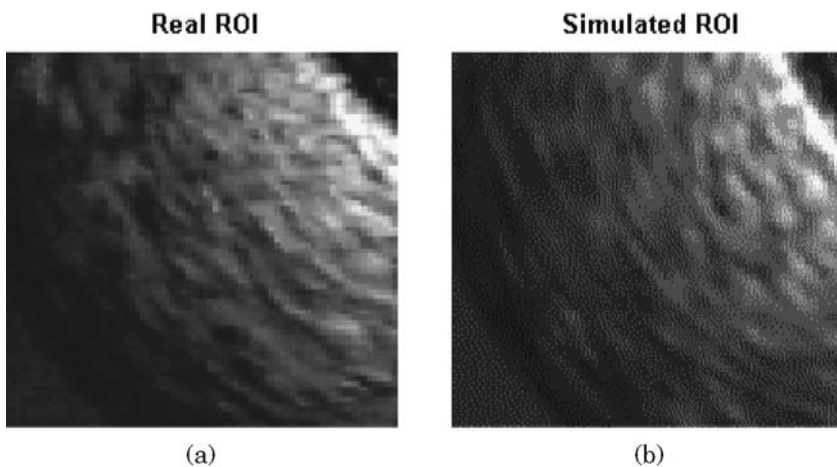


Figure 1.35: Real (a) and simulated (b) IVUS image ROIs.

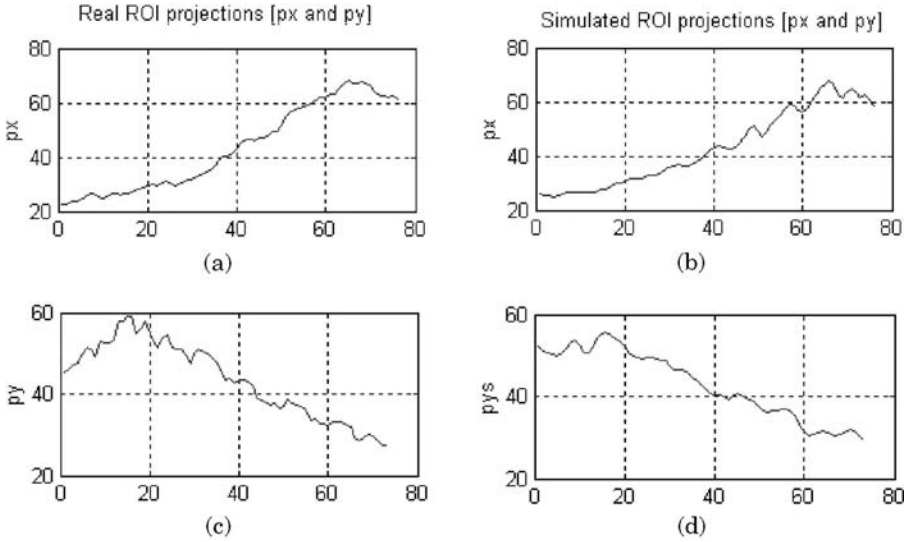


Figure 1.36: Horizontal ((a) and (b)) and vertical ((c) and (d)) projections of (Fig. 1.35(a)) and simulated (Fig. 1.35(b)) ROIs IVUS images.

1.6.5 Polar Images

A polar representation of IVUS images offers several advantages: (1) The ROIs to study are very easy to select, (2) we can compare the artifact generated by the smoothing procedures, (3) radial and angular comparisons are totally separated, therefore the transition zones in each direction are very easy to observe. Figure 1.38 shows real (a) and simulated (c) Cartesian IVUS images and the corresponding real (b) and simulated (d) polar transformations. An ROI was selected

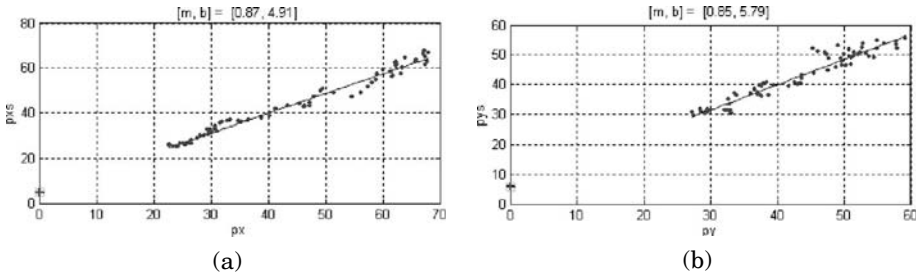


Figure 1.37: Gray-level average correlation, horizontal simulated (pxs) versus real projection (px), obtained from Fig. 1.36(a) versus (b), and vertical simulated (pys) versus real (py) data, from Fig. 1.36(c) versus (d).

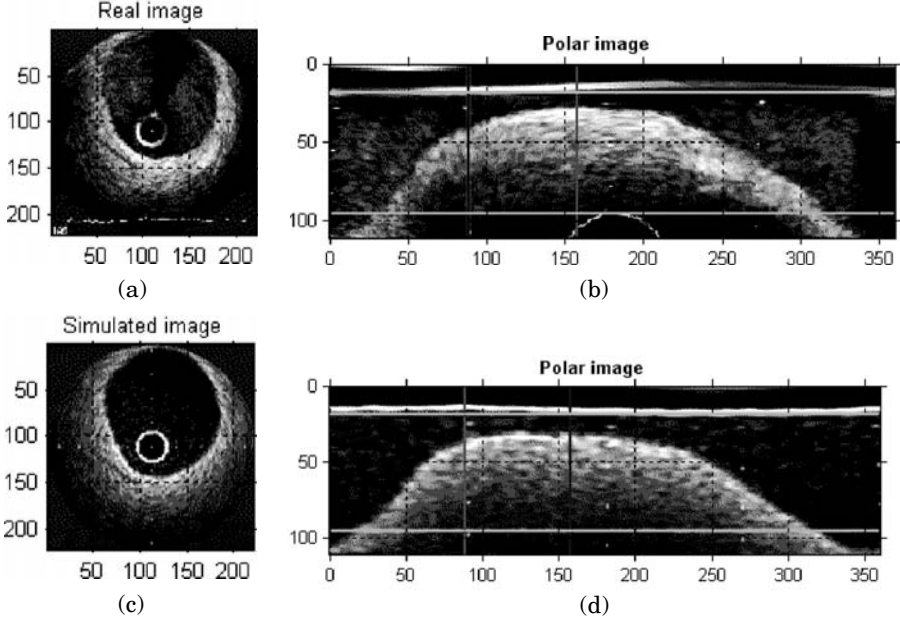


Figure 1.38: Real (a) and simulated (c) Cartesian images and their corresponding real (b) and simulated (d) polar transformation.

from the real and simulated polar images and the correlation coefficients were obtained. Figure 1.39(a) shows the gray-level average vertical projection for the real and simulated ROIs data (delineated in red in Fig. 1.38). We can see that the gray-level profiles of the transition of arterial structure in the lumen/intima, intima/media, and media/adventitia are very well simulated, the linear correlation coefficients being $m = 0.93$ and $b = 1.61$ (Fig. 1.39(b)). The *global horizontal profile* of the polar images along the projection θ (Figs. 1.40(a) and (b)) gives very important and comparative information about the real and simulated gray-level average of arterial structures. The information that can be extracted is relative to the global gray-level distribution. The histogram (Fig. 1.40(b)) of gray-level differences between the horizontal profiles of real and simulated data indicates a very good correspondence (mean $\mu = 8.5$ and deviation $\sigma = 10.2$). Figure 1.41(a) shows the global projection in the radial direction (the vertical profile). We can see a very good correspondence between the gray-level shape profiles (mean $\mu = 5.7$ and deviation $\sigma = 8.5$). The histogram (Fig. 1.41(b)) of gray-level difference confirms the good correlation between the real and simulated IVUS data.

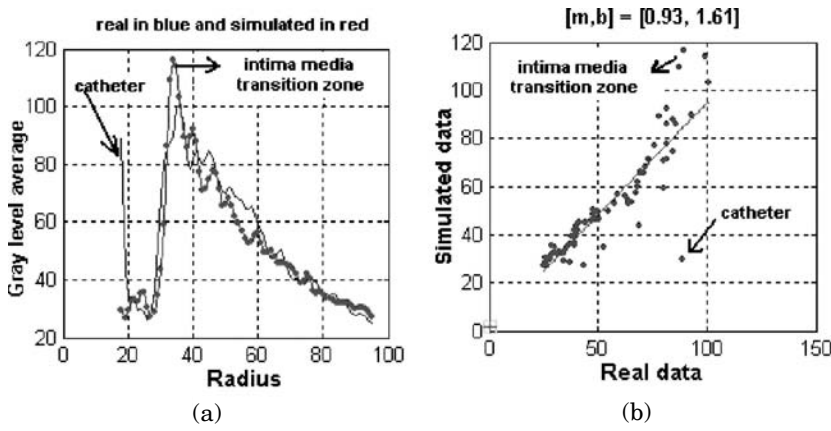


Figure 1.39: Real (blue) and simulated (red) gray-level vertical profile (a) of ROIs of Fig. 1.38(b) and data correlation (b).

The maximal difference profiles are localized in the transducer sheath gray-level distribution and the baseline of the transducer sheath inner region. These differences can be smaller, increasing the video and instrumental noise. The high-frequency oscillations in the gray-level profiles come from the concentric arterial structures. We can also observe the gradual reduction of the gray-level magnitude from intima/media interface to adventitia, caused by the ultrasound intensity attenuation.

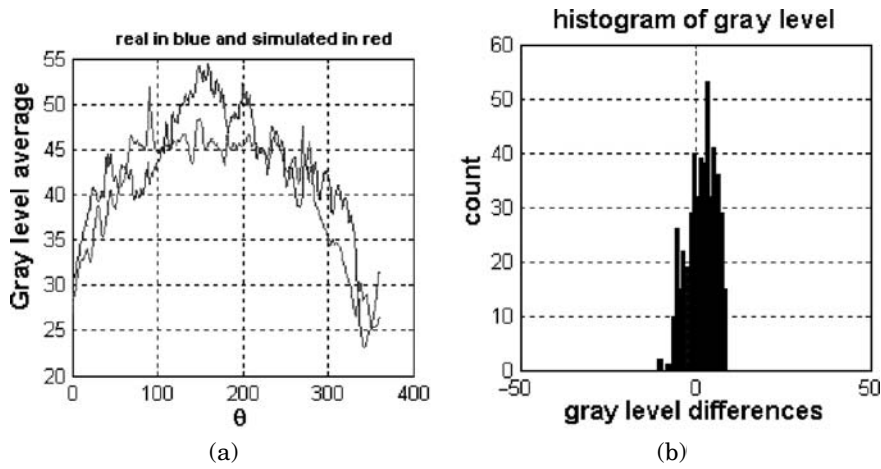


Figure 1.40: Global projections in the direction θ (a), from Figs. 1.38(b) and (d) and the corresponding histogram gray-level differences (b).

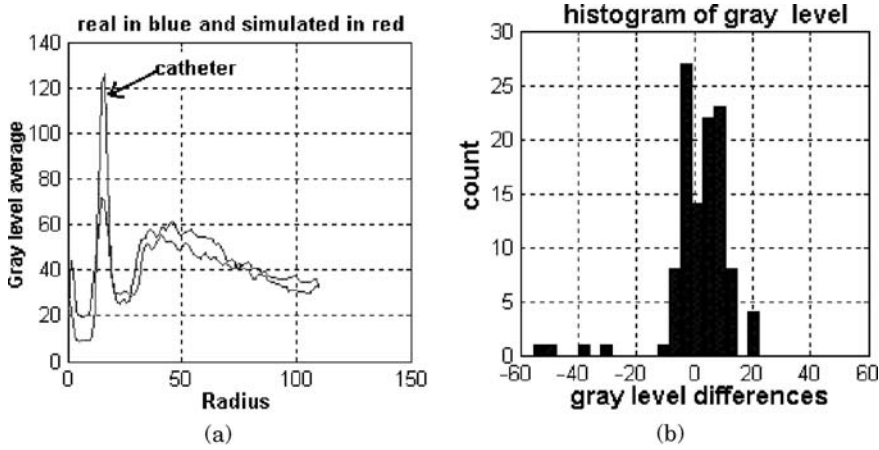


Figure 1.41: Global projection in the R direction (c), from Figs. 1.38(b) and (d), the corresponding histogram gray-level differences are shown in (b).

The next step in the validation of the model is to show the significant correspondence between real and simulated gray-level distribution data in the medical zones of interest. For this purpose, 20 validated real IVUS images and their corresponding ROIs were selected. The spatial boundaries of the morphological structures of the real data are kept in the synthetic data. Figure 1.42(a) shows ten real IVUS images and their corresponding simulated (b) synthetic images. The polar images are shown in Fig. 1.43.

Figure 1.44 shows the simulated versus real gray-level correlation for the polar ROIs images selected as shown in Fig. 1.38. The linear correlation coefficients show a good gray-levels correspondence, these being $m = 0.90$ and $b = 1.42$. The best correspondence is located by low gray levels (20–40 gray levels), lumen scatterers, lumen/intima transition, and adventitia. The transitions of intima/media and media/adventitia (45–60 gray levels) indicate gradual dispersion. The CNRS average presents significative uniformity values, $\mu = 6.89$ and $\sigma = 2.88$, for all validated frames. The CNRS as figure of merit for each arterial validated region is shown in Fig. 1.45. The CNRS region mean, standard deviation, and the SSE values referring to the 20 image frames are summarized in Table 1.5. The lumen is a good simulated region, with mean $\mu = 0.46$ and deviation $\sigma = 0.42$. The explanation is that the lumen is not a transition zone, the attenuation ultrasound intensity in this region is very poor (1–2%), which determines a simple gray-level profile.

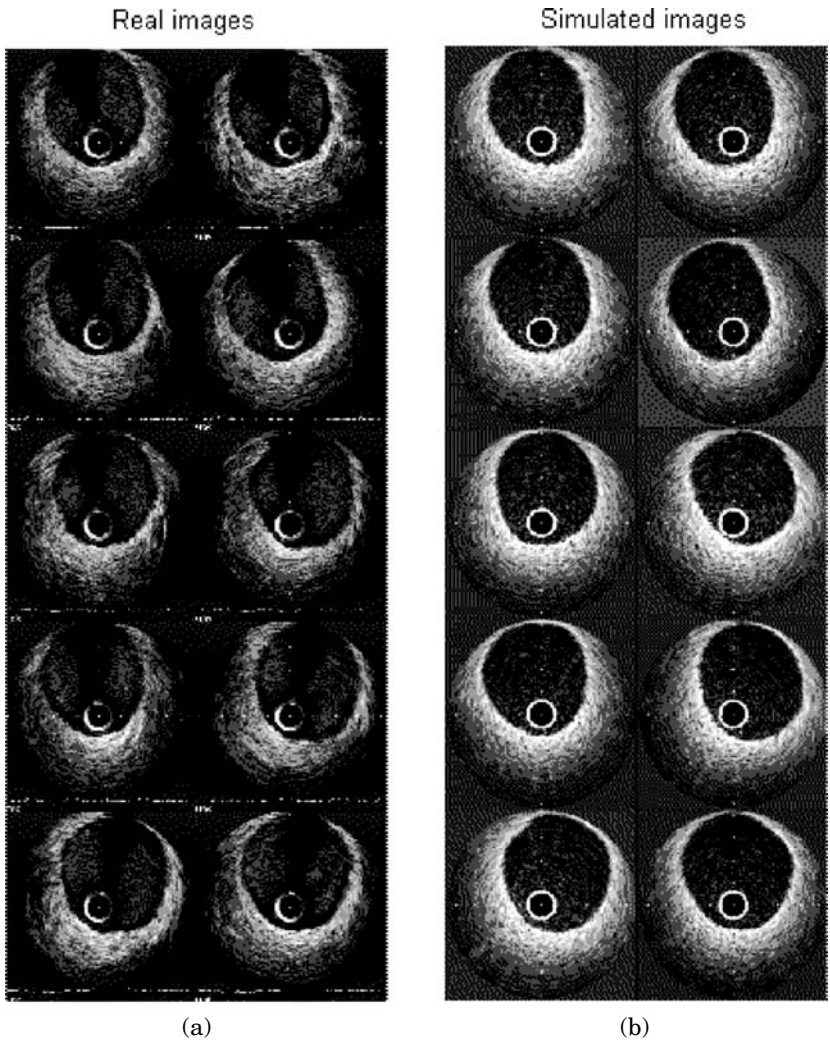


Figure 1.42: Ten original IVUS images (a) and the corresponding simulated (b) images.

The histograms of gray-level differences for each region of interest in the 20 validated frames are displayed in Figs. 1.46 and 1.47. Table 1.6 explains the distribution center μ and the standard deviation σ for the gray-level difference distribution for each simulated region. The minus sign in the mean values means that the simulated images are brighter than the real images. A symmetric Gaussian can be seen in the lumen gray-level differences distribution, with mean $\mu = -2.44$ and deviation $\sigma = 15.13$. The intima distribution has

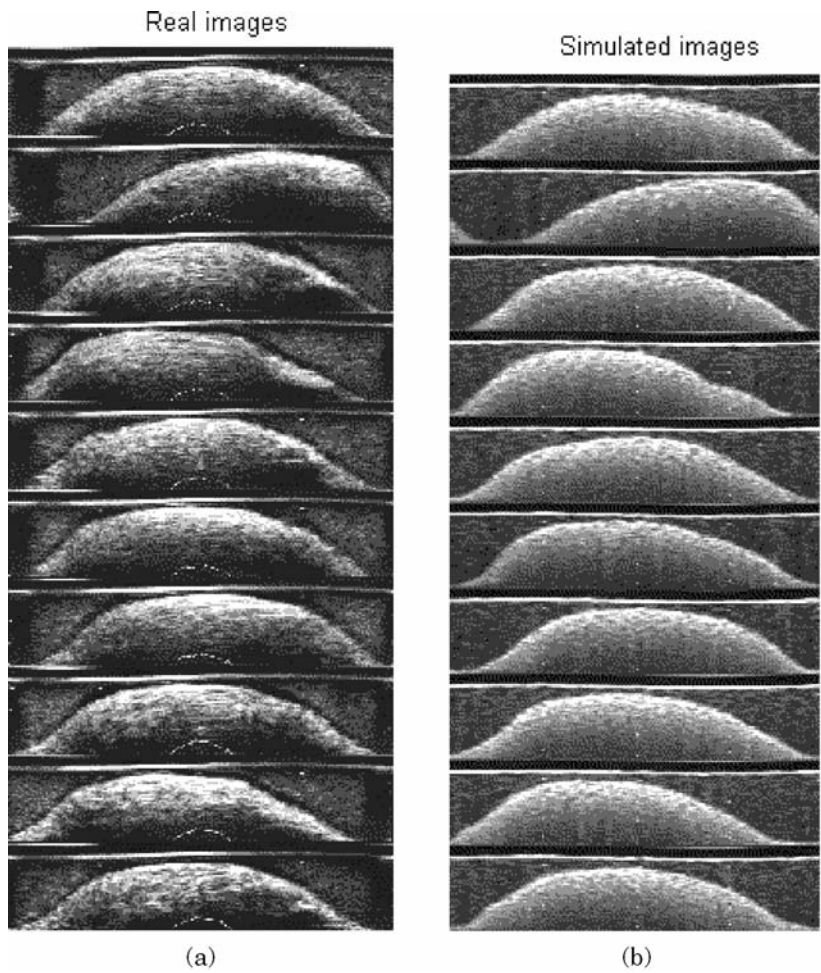


Figure 1.43: Ten polar real images (a) and the corresponding simulated (b) images.

Table 1.5: CNRS mean, standard deviation (std), and sum square error for different tissues structures

ROI	Mean	Std	SSE
Lumen	0.46	0.42	47.68
Intima	10.0	4.38	12.63
Media	9.91	5.14	15.05
Adventitia	7.21	2.76	4.28

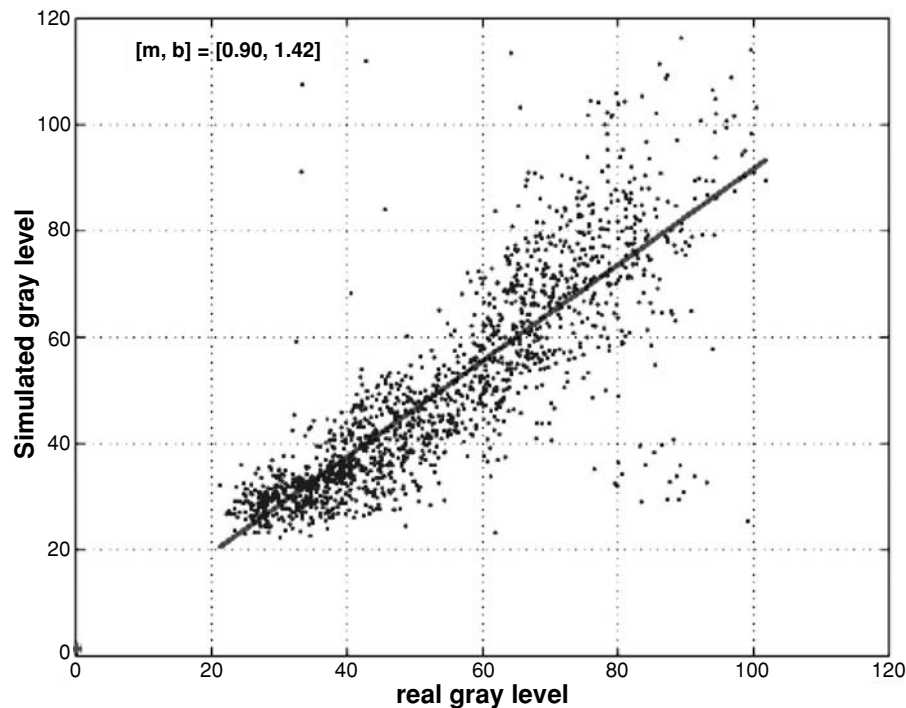


Figure 1.44: Simulated versus real gray-level values for 20 ROIs comparing pixel gray-level and the regression line.

a mean of $\mu = -18.56$ and deviation of $\sigma = 24.01$, and the media region has a mean of $\mu = -17.82$ and a deviation of $\sigma = 22.62$. The gray-level differences distribution displays a light asymmetry. As a result, the simulated image tends to be brighter than the real image. The adventitia gray-level differences values show a symmetric distribution with a center of $\mu = -13.30$ and a deviation of $\sigma = 14.27$.

Table 1.6: Mean and deviation of the ROIs gray-level differences referred from histograms in Figs. 1.46 and 1.47

ROI	μ	σ
Blood	-2.44	15.13
Intima	-18.56	24.01
Media	-17.82	22.62
Adventitia	-13.30	14.27

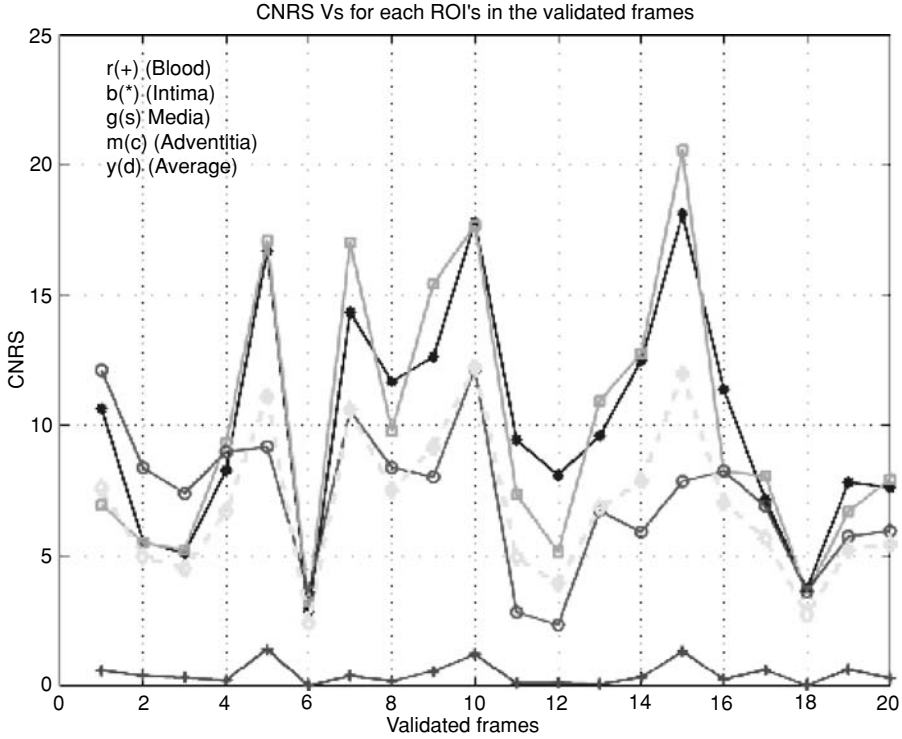


Figure 1.45: CNRS values for each ROI of 20 manually segmented image frames.

It is very important to note that the gray-level difference distribution exhibited Gaussian distributions for all regions of interest. Certainly, the synthetic image brightness is an open problem of the image formation model. The simplest approach is to variate it by modifying the original intensity I_o of the ultrasound beam, similar to the offset of the image acquisition system. Real and simulated gray-level distributions for each region of interest are shown in Figs. 1.48 and 1.49. We can note the great similarity in the gray-level distributions profile. Figure 1.50 shows the gray-level histogram of the different tissues structures that appear in IVUS images. As expected, it can be seen that the gray-level distributions of different structures overlap and as a result it is not possible to separate the main regions of interest in IVUS images, using only the gray-level distributions as image descriptors.

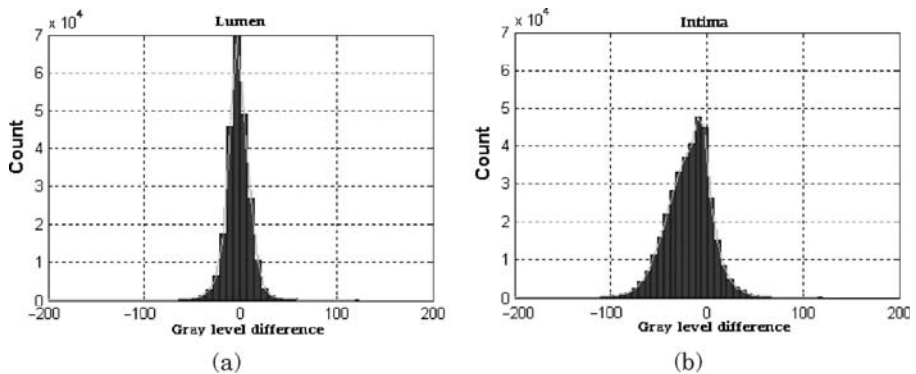


Figure 1.46: Histogram of gray-level differences for lumen (a) and intima (b).

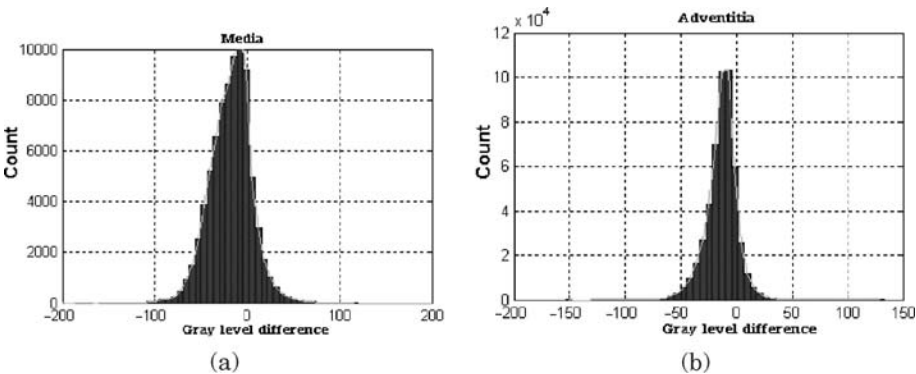


Figure 1.47: Histogram of gray-level differences for media (a) and adventitia (b).

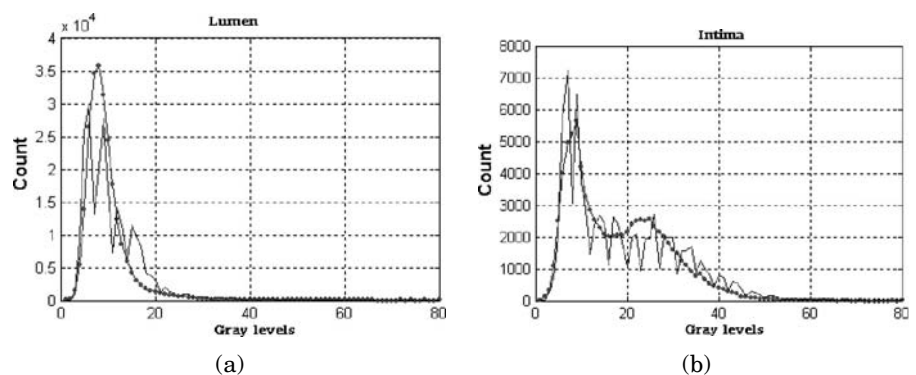


Figure 1.48: Real (blue) and simulated (red) gray-level distributions for lumen (a) and intima (b).

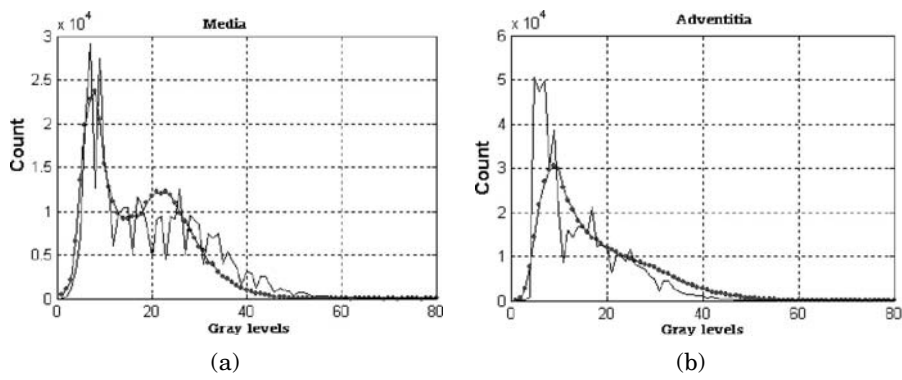


Figure 1.49: Real (blue) and simulated (red) gray-level distributions for media (a) and adventitia (b).

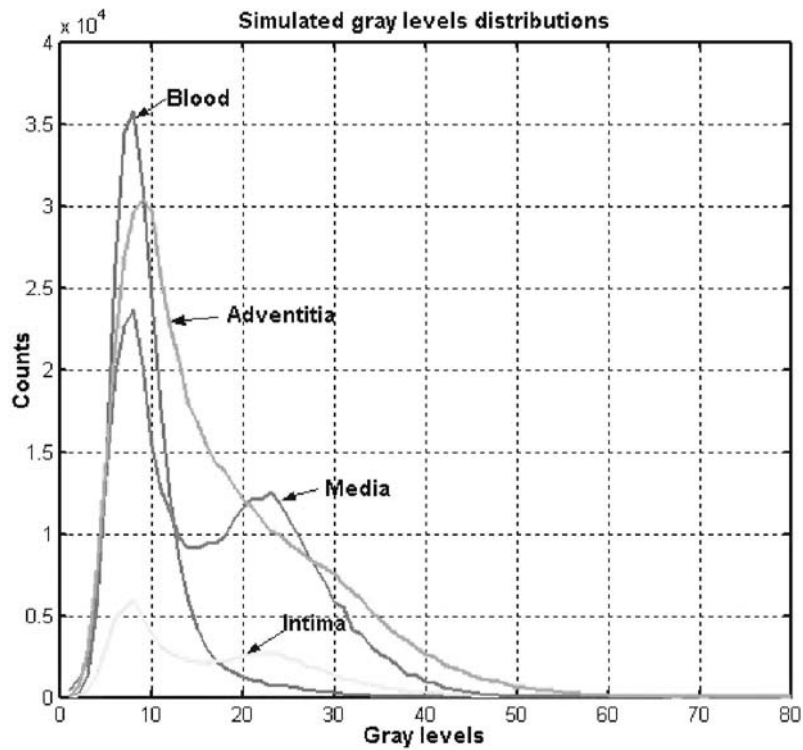


Figure 1.50: Simulated gray-level distributions for blood, intima, media, and adventitia.

1.7 Conclusions

Although IVUS is continuously gaining in use in practice due to its multiple clinical advantages, the technical process of IVUS image generation is not known to doctors and researchers developing IVUS image analysis. This fact leads to a simplified use, analysis, and interpretation of IVUS images based only on the gray-level values of image pixels.

In this chapter we discuss a basic physical model to generate synthetic 2D IVUS images. The model has different utilities: Firstly, an expert can generate simulated IVUS images in order to observe different arterial structures of clinical interest and their gray-level distribution in real images. Secondly, researchers and doctors can use our model to learn and to compare the influence of different physical parameters in the IVUS image formation, such as the ultrasound frequency, the attenuation coefficient, the beam number influence, and the artifact generations. Thirdly, this model can generate a large database of synthetic data under different device and acquisition parameters to be used for validating the robustness of image processing techniques. The IVUS image generation model provides a basic methodology that allows us to observe the most important real image emulation aspects. This initial phase does not compare pixel to pixel values generation, showing the coincidence with the real image, but looks for a global comparison method based on gray-level difference distribution. The input model applies standard parameters that have been extracted from the literature. Hence this model is generic in the sense that the model allows simulation of different processes, parameters, and makes it possible to compare to real data and to justify the generated data from the technical point of view.

The model is based on the interaction of the ultrasound waves with a discrete scatterer distribution of the main arterial structures. The obtained results of the validation of our model illustrate a good approximation to the image formation process. The 2D IVUS images show a good correspondence between the arterial structures that generate the image structures and their gray-level values. The simulations of the regions and tissue transitions of interest lumen, lumen/intima, intima/media, media/adventitia and adventitia, have been achieved to a satisfactory degree. Interested readers are invited to check the generation model in <http://www.cvc.uab.es/~misael>.

Questions

1. *Which qualitative phenomenon and parameters are possible to observe using the IVUS technique?*
2. *Which principles is IVUS data acquisition based on?*
3. *What are the principal limitations of the IVUS technique?*
4. *How is the distance to reflecting object by ultrasound technique determined?*
5. *What is attenuation coefficient?*
6. *What are axial and radial resolution?*
7. *What is the usual IVUS resolution?*
8. *How many scatterers structures are taken into account by a basic IVUS image model?*
9. *How are 1D and 2D echograms generated?*
10. *What are the steps followed in the generation of an IVUS image?*

Bibliography

- [1] Ponte Ignacio C., La epidemia del siglo XXI. Director de la Sección de Cardiología preventiva del Hospital Dr. Domingo Luciani, <http://www.aventispharma.com.ve/cardio.htm>, El Llanito, Edo. Miranda Venezuela, 2003.
- [2] Jonathan, M. and Paul, G., Intravascular Ultrasound imaging. Edited by, Jonathan M. Tobis and Paul G. Yock, Churchill Livingstone Inc., 1992, New York, USA.
- [3] Yock, P. *et al.*, Intravascular two dimensional catheter ultrasound, Initial clinical studies, abstracted, Circulations, No. 78 (Suppl II): II-21, 1988.
- [4] Graham, S. *et al.*, Assessment of arterial wall morphology using intravascular ultrasound *in vitro* and in patient, Circulations (Suppl II): II-56, 1989.
- [5] Metz Jonas, A., Paul, G., and Fitzgerald Peter, J., Intravascular ultrasound basic interpretation, In: Beyond Angiography, Intravascular Ultrasound, State of the Art, Vol. XX, Congress of the ESC Vienna-Austria, Stanford University School of Medicine, California, 1998.
- [6] Jumbo, G. and Raimund, E., Novel techniques of coronary artery imaging, In: Beyond Angiography, Intravascular Ultrasound, State of the Art, Vol. XX, Congress of the ESC Vienna-Austria, University of Essen, Germany, 1998.
- [7] Korte, Chris L., Intravascular Ultrasound Elastography, Article compilation of its doctoral thesis, Interuniversity Cardiology Institute of the Netherlands (ICIN), 1999.
- [8] Kearney, P. and Erbel, R., Imaging in the characterization laboratory, In: Beyond Angiography, Intravascular Ultrasound, State of the Art, Vol. XX, Congress of the ESC Viena-Austria, Johannes Gutenberg University, Mainz and University Clinic, Essen, Germany, 1998.

- [9] Berry, E. *et al.*, Intravascular ultrasound-guided interventions in coronary artery disease, Tech. Rep., Health Technology Assessment, NHS R D HTA Programme. A systematic literature review, with decision-analytic modelling of outcomes and cost-effectiveness, 2000.
- [10] Verhoef, W. A., Cloostermans, M. J., and Thijssen, J. M., The impulse response of a focused source with an arbitrary axisymmetric surface velocity distribution, *J. Acoust. Soc. Am.*, Vol. 75, pp. 1717–1721, 1984.
- [11] Fontaine, I., Bertrand, M., and Cloutier, G., A system-based approach to modelling the ultrasound signal backscattered by red blood cells, *Biophys. J.*, Vol. 77, pp. 2387–2399, 1999.
- [12] Fan, L., Herrington, D., and Santiago, P., Simulation of b-mode ultrasound to determine features of vessel for image analysis, *Comput. Cardiol.*, Vol. 25, pp. 165–168, 1998.
- [13] Kinsler, L., *Fundamentos de acústica*, Noriega Editores, 1995, México DF.
- [14] Cheeke, D., *Fundamentals and Applications of Ultrasonic Waves*, CRC Press, Boca Raton, Florida, 2002.
- [15] Thijssen, J. and Oosterveld, B., Performance of echographic equipment and potentials for tissue characterization, NATO ASI Series, Mathematics and Computer Science in Medical Imaging, Vol. F39, pp. 455–468, 1988.
- [16] Zagzebski, J., *Essential of Ultrasound Physics*, Mosby, A. ed., Harcourt Health Sciences Company, 1996. St. Louis, Missouri.
- [17] Arendt, Jesen J., Linear description of ultrasound imaging system, Notes for the International Summer School on Advanced Ultrasound Imaging, Technical University of Denmark, 2001.
- [18] Young, B. and Heath, J., Wheather's, *Histología Funcional*, 4ta edición, Ediciones Hardcourt, S.A., Philadelphia, Pennsylvania 19106.
- [19] Mazumdar, J., *Biofluids Mechanics*, World Scientific, Dordrecht, 1992.
- [20] K. K. Shung and G. A. Thieme, *Ultrasonic Scattering in Biological Tissues*, CRC Press, Boca Raton, Florida, 1993.

- [21] Guyton A., *Tratado de Fisiología Médica*, Décima edición, McGraw-Hill Interamericana, Madrid, Espana.
- [22] Perelman L. *et al.*, Observation of periodic fine structure in reflectance from biological tissue: A new technique for measuring nuclear size distribution, *Phys. Rev. Lett.*, Vol. 80, No. 3, pp. 627–630, 1998.
- [23] Duda, R., Hart, P., and Stork, D., *Pattern Classification*, John Wiley & Sons, Inc., New York, 2000.
- [24] Boston Scientific Corporation, Scimed division, *The ABCs of IVUS*, 1998.
- [25] Gonzales, R. and Wintz, P., *Digital Image Processing*, Addison-Wesley, Reading, Massachusetts 1987.
- [26] O'Donnell, M. and Silverstein, S., Optimum displacement for compound image generation in medical ultrasound, *IEEE Trans. Ultrason., Ferroelectr. Freq. Control*, Vol. 35, No. 4, pp. 470–476, 1988.
- [27] Rosales, M. and Radeva, P., Empirical simulation model of intravascular ultrasound, *Tech. Rep.*, No. 71, Centre de Visió per Computador, Universitat Autònoma de Barcelona, España, 2003.
- [28] Vogt, M. *et al.*, Structural analysis of the skin using high frequency broadband ultrasound in the range from 30 to 140 mHz, In: *IEEE International Ultrasonics Symposium*, Sendai, Japan, 1998.

Chapter 2

Quantitative Functional Imaging with Positron Emission Tomography: Principles and Instrumentation

*Koon-Pong Wong*¹

2.1 Introduction

The last few decades of the twentieth century have witnessed significant advances in medical imaging, which had been exclusively concerned with conventional film/screen X-ray imaging for more than 75 years after the discovery of X-ray by Wilhelm Röntgen in 1895. In particular, when Allen Cormack and Godfrey Hounsfield introduced X-ray computed tomography (CT) independently in the early 1970s [1–3] based on the mathematical foundation laid by Radon [4] for reconstructing images of an object from its projections, the field of medical imaging was revolutionized. Imaging with X-ray CT has enabled us to view noninvasively, for the first time, the anatomic structure of internal organs with unprecedented precision and to recognize any gross pathology of organs and diseases. This also marked a new era of medical diagnostics with many invasive and potentially morbid procedures being substituted by noninvasive cross-sectional imaging.

The breakthrough development of X-ray CT was made possible by continuing advances in instrumentation and computer technologies, which also accelerated the development of other multi-dimensional imaging modalities that possess a great potential for providing, in addition to morphologic (structural)

¹Department of Electronic and Information Engineering, Hong Kong Polytechnic University Hung Hom, Kowloon, Hong Kong

information, dynamic and functional information on biochemical and pathophysiological processes or organs of the human body. The importance of studying organ functions was recognized in the middle of the nineteenth century, but the actual relationship between physiological disturbances and anatomical lesions was not yet elucidated. This was partly due to the concept of disease classification, which was primarily based on anatomical lesions and causes of disease, during that period of time.

Recent advances in basic molecular and cell biology have led to a revolutionary change in our understanding of diseases. Instead of defining disease as structural changes or histopathological abnormality, it can be defined as alterations in cellular behavior that reflect functional changes. It is important to realize that in living systems, what we call *function* is a process that evolves over time as energy is produced during the life cycle or information is transferred and processed within cells, whereas *structure* is simply a snapshot of function at a particular time instant. Indeed, it is very common that in many diseases structural changes are completely absent, or physiological changes precede structural changes. A typical example is cancer, which consists of cells in which malfunctioning transformation has taken place owing to exposure to some environmental factors (e.g. viruses, bacteria, irradiation, saccharin, and a variety of chemical substances) that can cause altered membrane characteristics and cell metabolism, deformed cell morphology, etc. as a result of alternation in cell functions and damage in genes that control cell proliferation and migration. It was first hypothesized by Otto Warburg in 1930 that the rate of glucose metabolism (aerobic glycolysis) in tumors increases with higher degree of malignancy when compared to normal tissue [5], and this is regarded as one of the important indicators of tumor proliferation. If these biological characteristics could be evaluated *in vivo*, useful information may be obtained to study the nature of disease early in and throughout its evolution, as well as to identify and develop effective therapies for treatment. Functional imaging makes it possible to visualize and measure, with the use of appropriate imaging probes and agents, these complex pathophysiological and biochemical processes in a living system *in vivo* in multi-dimensional domains (three-dimensional spatial domains plus a temporal domain).

There is no doubt that substantial progress has been achieved in delivering health care more efficiently and in improving disease management, and

that diagnostic imaging techniques have played a decisive role in routine clinical practice in almost all disciplines of contemporary medicine. With further development of functional imaging techniques, in conjunction with continuing progress in molecular biology and functional genomics, it is anticipated that we will be able to visualize and determine the actual molecular errors in a specific disease within a decade or so, and be able to incorporate this biological information into clinical management of that particular group of patients. This is definitely not achievable with the use of structural imaging techniques.

In this chapter, we will take a quick tour of a functional imaging technique called *positron emission tomography* (PET), which, in conjunction with *single-photon emission computed tomography* (SPECT), is commonly known as *emission computed tomography*. PET is a premier biologic imaging tool, being able to provide *in vivo* quantitative functional information in most organ systems of the body. In the following sections, an overview of this imaging technique is provided, including the basic principles and instrumentation, methods of image reconstruction from projections, some specific correction factors necessary to achieve quantitative images, as well as basic assumptions and special requirements for quantitation. Paradigms based on the framework of tracer kinetic modeling for absolute quantification of physiological parameters of interest are also introduced. However, as they deem inappropriate for inclusion in this book, topics on hardware technologies (e.g. display and archival units, data-acquisition computer system, electronics circuitry, array processors, etc.) of a PET system, operating principles of a cyclotron, as well as design and development of radiopharmaceuticals are not discussed in this chapter.

2.2 A Brief History of PET

The development of PET has involved efforts of investigators from diverse disciplines and spanned almost the whole twentieth century. At the turn of the twentieth century, Ernest Rutherford and Frederick Soddy (who coined the term *isotope*) reported their studies on the nature and cause of radioactivity in McGill University [6]. Their work on radioactive half-life and exponential decay is the foundation for medical applications of radioisotopes, including the breakthrough development of emission computed tomography.

The existence of positively charged electrons (positrons) was postulated by Paul Dirac in 1928, based on Einstein's theory of relativity and the equations of quantum mechanics [7]. It was first observed experimentally by Carl Anderson in 1932 [8], for which he was awarded the Nobel Prize for Physics in 1936. The phenomenon of positron annihilation that gives rise to gamma rays was observed by Joliot [9] and Thibaud [10] in 1933. It was shown later that, in general, two photons are simultaneously emitted in almost exactly opposite directions whenever a positron passes through matter [11]. The use of positron emitters for medical imaging purposes was first suggested by Wrenn *et al.* [12] and Sweet [13] in the early 1950s. The first successful positron imaging device was described by Brownell and Sweet [14]. The system was used for two-dimensional imaging of positron-emitting radionuclides (copper-64 and arsenic-75) distribution to locate brain tumors in human, using a pair of NaI(Tl) detectors. In 1963, Kuhl and Edwards introduced the concept of transverse- and longitudinal-section scanning with single-photon emitting radionuclides [15] and a device (Mark IV scanner), which consisted of a square array of 32 NaI(Tl) detectors, was built later for constructing images by superimposing multiple cross sections of transverse axial scans [16]. Although the reconstruction method was very primitive and the reconstructed images were severely blurred, the development of PET was accelerated by the introduction of transverse axial X-ray CT for radiography by Cormack and Hounsfield [1–3]. There have also been a number of techniques developed for performing emission tomography during the early 1970s [17–19], but all of these approaches were limited by inadequate mathematical reconstruction algorithms, insufficient angular sampling frequency, image distortions due to photon attenuation and some other statistical limitations.

The first positron computed tomograph was developed in 1975 by Ter-Pogossian *et al.* [20]. This system was referred to as positron emission transaxial tomography (PETT II), which consisted of a hexagonal array of NaI(Tl) detectors connected in coincidence between opposite pairs. The filtered-backprojection (FBP) reconstruction method was adopted in that system, and the quality of the reconstructed images was markedly improved. The first whole-body positron computed tomograph (PETT III) was developed shortly thereafter and it was used in human studies [21–24]. This system was subsequently redesigned and manufactured by EG&G/ORTEC as the commercial PET scanner, ECAT [25].

2.3 Modes of Decay

The nucleus of an atom contains both protons and neutrons, which are collectively known as nucleons. In a stable nucleus, the number of protons and neutrons is such that the repulsive electrostatic force between the positively charged protons is balanced by the very strong attractive nuclear forces which act on all nucleons. It is possible to create unstable isotopes which have an excess number of protons using nuclear reactors or cyclotrons. These *proton-rich* (or *neutron-deficient*) isotopes can have two means of decay that will reduce the excess positive charge on the nucleus: (1) electron capture and (2) positron emission.

If the nucleus does not have sufficient energy to decay by positron emission (to be described next), it will decay by *electron capture*, whereby the nucleus captures one of the orbital electrons from the inner shells and combines this with the proton to form a neutron, while the vacancy in the inner electron shell is immediately filled by an electron from a higher energy shell, resulting in emission of characteristics X-rays whose energies are carried off by the neutrino:



where Z represents the atomic number of the atom X , A is the mass number, e^- is an electron, and ν is a neutrino, which has a very small mass and zero charge. Electron capture occurs in heavier proton-rich nuclides with higher likelihood due to the closer proximity of the inner (usually K or L) shell electrons to the nucleus and the greater magnitude of the Coulombic attractive force from the positive charges. The characteristics X-ray energy increases with the mass number of the nuclides. For example, the decay of ^{125}I produces 27 keV characteristics X-ray which is used for *in vitro* counting, whereas the decay of ^{201}Tl produces characteristics X-rays ranged from 68 to 80 keV which are used in gamma-camera imaging.

The major radioactive decay mechanism for positron emitters used in PET is *positron emission*, whereby a proton in the nucleus is transformed into a neutron and a positron. The positron (β^+) has exactly the same mass and same magnitude of charge as the electron except that the charge being carried is positive. The nuclear equation for positron emission can be written as



Table 2.1: Properties of some commonly used positron-emitting isotopes

Radionuclide	Production	β^+ fraction	$\mathcal{E}_{\max}$ (MeV) ^a	Half-life (min)	Range (mm) ^b
Carbon-11 (¹¹ C)	Cyclotron	0.99	0.96	20.4	0.28
Fluorine-18 (¹⁸ F)	Cyclotron	0.97	0.64	109.7	0.22
Nitrogen-13 (¹³ N)	Cyclotron	1.00	1.19	9.96	0.45
Oxygen-15 (¹⁵ O)	Cyclotron	1.00	1.72	2.07	1.04
Gallium-68 (⁶⁸ Ga)	Cyclotron	0.89	1.90	68.3	1.35
Rubidium-82 (⁸² Rb)	Generator	0.96	3.35	1.25	2.6

^a $\mathcal{E}_{\max}$ = maximal positron energy.

^b Approximated distance that a positron traveled before annihilation, expressed in full width at half maximum (FWHM).

For positron emission to be energetically feasible, the total energy difference between the parent and the daughter states should be at least 1.022 MeV, which is the energy equivalent of a positron and an electron, according to Einstein's energy-mass equivalence: $E = mc^2$. The energy difference between the parent and the daughter states is shared between the positron and the neutrino. In other words, the emitted positrons have a spectrum of energies, whose maximum is given by

$$\mathcal{E}_{\max} = \mathcal{E}({}_Z^AX) - \mathcal{E}({}_{Z-1}^AY) - 1.022 \text{ MeV}. \quad (2.3)$$

Typically, the likelihood of positron emission is higher for elements with lower atomic number, but for proton-rich nuclei with intermediate atomic number both decay modes are competing with each other. Table 2.1 lists some commonly used positron-emitting isotopes and their properties. Positron emitters are of special interest in medicine because the main elements (e.g. carbon, oxygen and nitrogen) that constitute living organisms have isotopes that emit positrons. The only exception is hydrogen for which fluorine-18 is an analogue.

2.4 Positron Annihilation

The positron will have some initial energy after emission from the parent nucleus. It travels a short distance from the nucleus, scatters and collides with loosely bound electrons nearby before fusing with one of them to form *positronium* (which has a very short half-life, $\approx 10^{-7}$ s) and then annihilates. Their

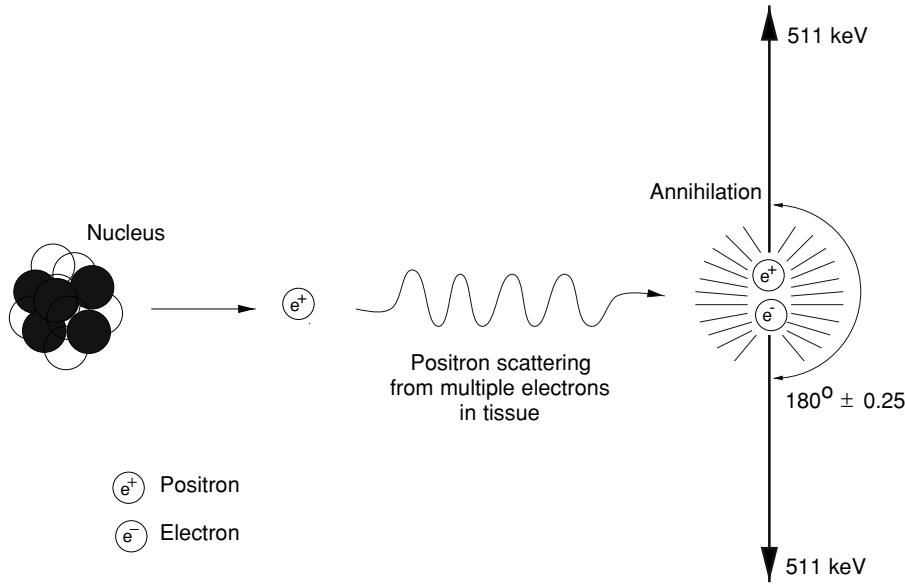


Figure 2.1: Positron emission and annihilation. A positron is emitted from a proton-rich nucleus, losing energy by scattering from atomic electrons in tissue before annihilating with an electron to produce two 511 keV photons (or gamma rays) which are moving $180^\circ (\pm 0.25^\circ \text{ FWHM})$ apart.

mass converts into energy in the form of two 511 keV photons, which are indistinguishable from gamma rays. To simultaneously conserve both momentum and energy, the photons are emitted 180° to each other. Figure 2.1 shows the positron annihilation and the emission of two 511 keV photons. The detection of these two 511 keV photons forms the basis of PET imaging.

2.5 Coincidence Detection

Since the probability that both 511 keV photons will escape from the body without scattering is very high in general, the line along which the positron annihilation occurred (i.e. the line of response, LoR) can be defined if both photons can be detected with two detectors at opposite ends of the line, as illustrated in Fig. 2.2. As the distance that a positron traveled before annihilation is generally very small, this is a good approximation to the line along which the emitted photons must be located. The scheme for detection of photon emissions is called

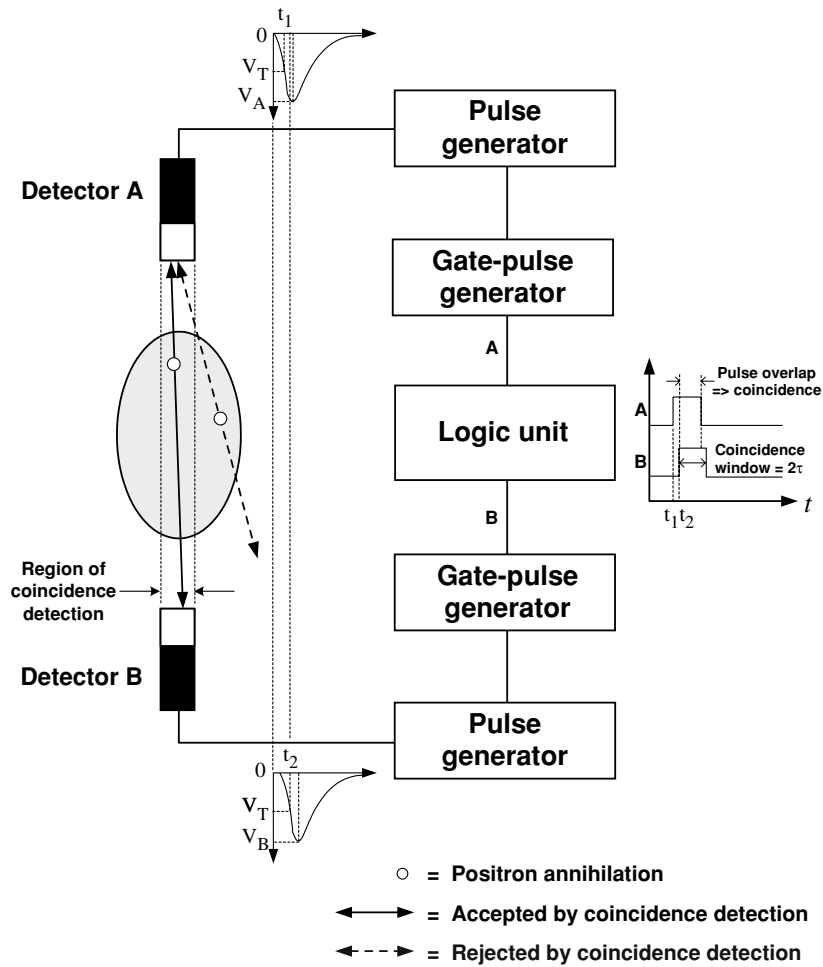


Figure 2.2: Annihilation coincidence detection. The two gamma-ray detectors are placed at the opposite ends of the object to detect the photons that originate from the positron annihilation site. The event is registered if the annihilation occurs within the region of coincidence detection of the detector pairs. If the gamma rays originate outside the region of coincidence detection of the two detectors but only one of the photons is detected, the event is not registered as the detection of a single photon violates the condition of coincidence.

coincidence detection [21], which is unique to PET imaging. It should be noted, however, that the condition of *coincidence* (or simultaneity) is not achievable in practice, and a *coincidence resolving time* (or a *coincidence timing window*) of less than 15 ns is often used to account for differences in arrival times of the

two gamma rays, time taken to produce scintillation light in the detector, and time delays in the electronic devices in the PET system.

Once the signal leaves the detector module, it is processed by several electronic circuits. The choice of components depends upon the application and, therefore, there are many ways to implement the coincidence detection circuitry. A simplified schematic representation of detecting coincidence events with two detectors is also shown in Fig. 2.2. The output signal from each detector is fed into a pulse generator. Note that the signal amplitude from the two detectors (V_A and V_B) may not be the same due to incomplete deposition of photon energies or variation in efficiency among the detectors. In addition, there exists a time difference between the detectors to react upon the photons arrival, and a finite reaction time for the electronic devices to response, resulting in difference in the time t_1 and t_2 at which the amplitude of the signal crosses a certain fixed voltage level (V_T), which triggered the pulse generator to produce a narrow pulse. The narrow pulse is then fed into the gate-pulse generator where a pulse of width 2τ (coincidence timing window) is generated for individual detectors. A coincidence detection circuit is then used to check for a logical AND between the incoming pulses. For the example shown in Fig. 2.2, there is a pulse overlap between two signals produced by the gate-pulse generators. Therefore, the event is a true coincidence which is regarded as valid and is registered. It is easy to see that if $t_2 - t_1 \geq 2\tau$, the event is not in coincidence, and thus it is not recorded by the coincidence detection circuit.

2.6 Coincidence Criteria

In general, an event (positron annihilation/photon emissions) is regarded as valid and is registered by the coincidence detection circuit if the following criteria are satisfied [26, 53]:

- two photons are detected within a predefined coincidence window,
- the LoR formed between the two photons is within a valid acceptance angle of the tomograph, and
- the energy deposited in the crystal of the detector by both photons is within the selected energy window.

Such coincident events are often referred to as *prompt* events.

2.7 Detectors

To image the distribution of positron-emitting isotope in the body, both of the 511 keV photons emitted from positron annihilation must be detected in coincidence. Unlike other instruments used in nuclear medicine, PET uses electronic rather than lead collimators to detect signal (event) results from annihilation of the positron and an electron. The probability of detecting both photons depends on the detector efficiency, which is strongly related to the stopping power of the scintillator and the thickness of the scintillator used in the detector. Early generation of PET scanners used NaI(Tl) crystals, the same material used in gamma camera. Modern PET scanners use much denser scintillators, such as bismuth germanate oxide (BGO) [27], which has been the scintillator of choice for more than two decades due to its very high density and stopping power for the 511 keV gamma rays. In order to provide higher detection efficiency and spatial resolution with lower production cost, a number of detector designs were proposed in the 1980s and the most successful one was the *block detector* technique proposed by Casey and Nutt, using BGO crystal [28]. A typical BGO block detector comprises a rectangular block consisting of between 6×8 and 8×8 individual scintillation crystals, attached to an array (usually 2×2) of photomultiplier tubes (PMTs) at which the scintillation light is amplified and converted into electrical signal for the coincidence detection circuit to register. A schematic outline of such a block detector is shown in Fig. 2.3. The BGO element in which a gamma ray interacts is determined by the relative light output

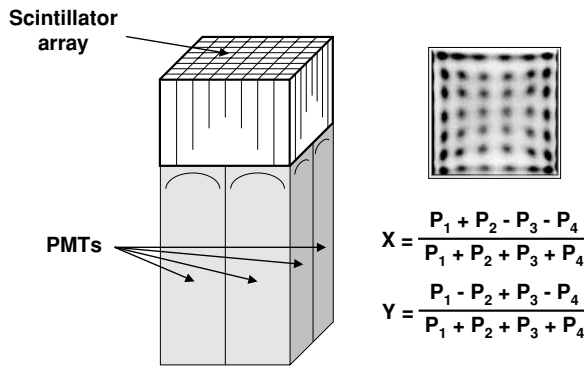


Figure 2.3: Schematic diagram of a BGO block detector commonly used in commercial PET systems.

from the four PMTs. Anger-logic is used to obtain the X and Y positions based on the four PMT outputs P_i :

$$X = \frac{P_1 + P_2 - P_3 - P_4}{P_1 + P_2 + P_3 + P_4} \quad (2.4)$$

$$Y = \frac{P_1 - P_2 + P_3 - P_4}{P_1 + P_2 + P_3 + P_4} \quad (2.5)$$

The combined BGO block/photomultiplier system has an approximately cubic spatial resolution of 4 mm full-width-at-half-maximum and coincidence timing window of approximately 12 ns.

As seen from Fig. 2.2, the probability that the annihilation event occurs exactly within the region of coincidence detection and is recorded by the detectors is very small because most gamma rays may travel out of the region of coincidence detection even if the annihilation event occurs within that region. This probability can be increased by using a ring of detectors within which any detector is in coincidence with all other detectors located at the opposite side of the ring. With the use of multiple rings of detectors, the probability of coincidence detection is further increased because coincidences can be detected by other rings of detectors if they cannot be recorded by the plane of the ring within which the annihilation events occur. The device that used to detect the 511 keV gamma rays emitted from annihilation and construct a map of radiopharmaceutical distribution inside the body is called *tomograph* (or *scanner*), which usually has multiple rings of detectors surrounding the patient.

2.8 Detected Events

Ideally the only prompt events registered by the detectors are those which arise from “real” positron annihilation. However, a number of other unwanted events that satisfied the coincidence criteria are also registered. The detection of unwanted events causes noise and degradation in spatial resolution. Therefore, their correction is essential to improve the quantification. In general, five types of event can be detected by PET scanner, and four of them are illustrated in Fig. 2.4.

A *true coincidence* refers to an event that two photons are emitted back-to-back from a single positron–electron annihilation, and are detected simultaneously by opposing detectors within the region of coincidence detection and within the coincidence timing window of the system.

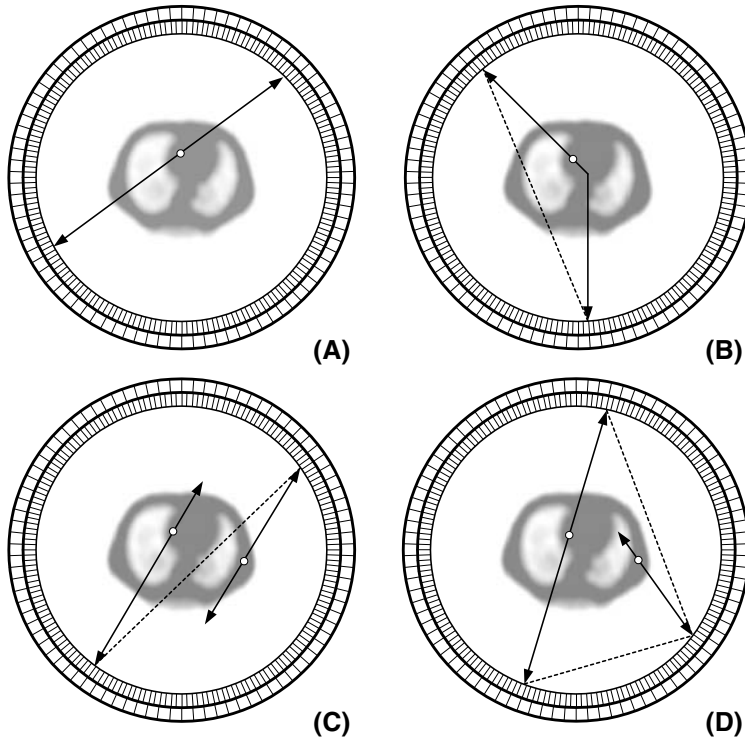


Figure 2.4: Types of coincidence event recorded by a full-ring PET system. The white circle indicates the site of positron annihilation, and the solid line represents the gamma ray, (A) true coincidence, (B) scattered coincidence, (C) random (or accidental) coincidence, and (D) multiple coincidence. The mispositioned line of response is indicated by the dashed line.

Scattered coincidence occurs when one or both of the emitted photons undergo a Compton scatter interaction in tissue. Compton scattering causes a loss in energy of the photon and change in direction of the photon. Since the direction is changed, the origin where the photons were emitted cannot be located correctly and, as a result, the event is mispositioned, leading to decreased contrast and deteriorated quantification.

A *random (or accidental) coincidence* occurs when two unrelated photons, which have not originated from the same site of positron annihilation, strike opposing detectors within the coincidence timing window. Since the random events are produced by photons emitted from unrelated decays, they are spatially uncorrelated with the activity distribution. The random coincidences are

a source of noise, the rate of which is approximately proportional to the square of the activity in the field of view (FOV). The performance of PET scanner for high count rate studies is degraded and therefore, correction for randoms is necessary.

Multiple events are similar to random events, except that three photons originated from two positron annihilations are detected within the coincidence timing window. Because of the ambiguity in positioning the event, these coincidences are normally discarded.

A *single event* for which only one photon is emitted is also possible due to some physical factors. The single events are usually rejected by the coincidence detection circuit since detection of only one event within the timing window violate the condition of coincidence. Yet in practice, about 1–10% of single events are converted into paired coincidence events.

2.9 Data Acquisition

Most of the modern PET tomographs are capable of acquiring data in two different modes: two-dimensional (planar) acquisition with septa in-place and three-dimensional (volumetric) acquisition with septa retracted, exposing the detectors to oblique and transaxial annihilation photon pairs. Both modes of configuration for data acquisition are shown in Fig. 2.5. In two-dimensional imaging, each ring of detectors is separated by septa made of lead or tungsten. The main aim is to keep the scatter and random coincidence event rates low so as to minimize the cross-talk between rings. However, in doing so, the sensitivity of the scanner is drastically reduced. Three-dimensional acquisition can be used to improve the sensitivity by removing the interplane septa, thus allowing coincidences that happened within all rings of detector to be detected. Although the sensitivity of the scanner is increased, higher fraction of scattered and random coincidences and substantial dead time are more apparent.

In a tomograph, each detector pair records the sum of radioactivity along a given line (i.e. the *line integral* or *projection*) through the body. The data recorded by many millions of detector pairs in a given ring surrounding the body is stored in a two-dimensional (projection) matrix called *sinogram*, as shown in Fig. 2.6(B) and Fig. 2.6(A) shows how data is acquired in two-dimensional mode. Each point in the sinogram represents the sum of all events detected with

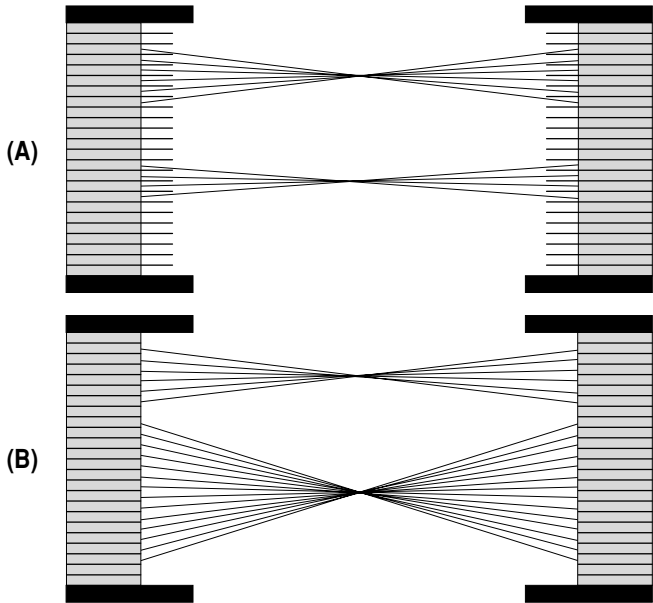


Figure 2.5: (A) Axial cross-section of a PET scanner with septa in-place for two-dimensional data acquisition. (B) Axial cross-section of a PET scanner with septa retracted for three-dimensional data acquisition.

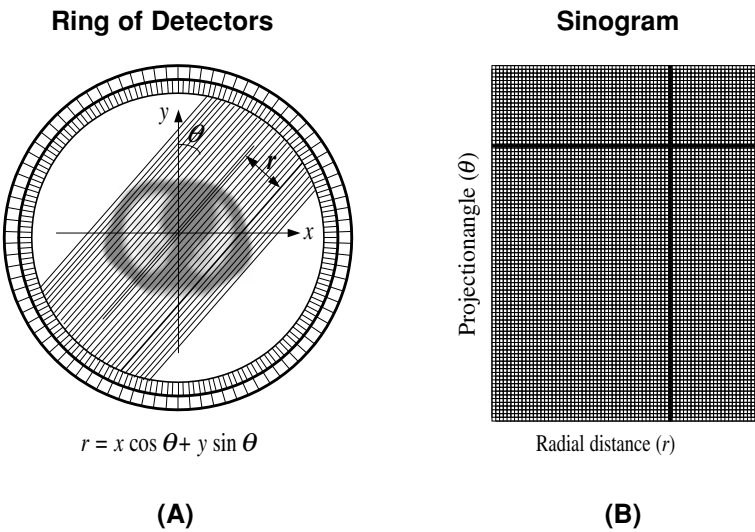


Figure 2.6: Schematic diagram showing how projection data is acquired (A) and stored in the sinogram (B) for two-dimensional PET imaging.

a particular pair of detectors, and each row represents the projected activity of parallel detector pairs at a given angle relative to the detector ring. In other words, if $\mathbf{p}$ represents the sinogram and $p(r, \theta)$ represents the value recorded at the (r, θ) position of $\mathbf{p}$, then $p(r, \theta)$ represents the total number of photon emissions occurring along a particular line joining two detectors at a distance r from the center of the tomograph, viewed at an angle θ with respect to the y -axis (or the x -axis, depending on how the coordinate system is chosen) of the tomograph. However, the sinogram provides only little information about the radiopharmaceutical distribution in the body. The projection data in the sinogram has to be reconstructed to yield an interpretable tomographic image.

2.10 Image Reconstruction

The goal of image reconstruction is to recover the radiotracer distribution from the sinogram. The reconstruction of images for the data acquired with the two-dimensional mode is simple, while the reconstruction of a three-dimensional volumetric PET data is more complicated, but the basic principles of reconstruction are the same as those for the two-dimensional PET imaging. We focus the discussion on the two-dimensional PET image reconstruction for simplicity. A more thorough discussion of three-dimensional data acquisition and image reconstruction can be found elsewhere [29].

The theory of image reconstruction from projections was developed by Radon in 1917 [4]. In his work, Radon proved that a two-dimensional (or three-dimensional) object can be reconstructed exactly from its full set of one-dimensional projections (two-dimensional projections for three-dimensional object). In general, image reconstruction algorithms can be roughly classified into (1) Fourier-based and (2) iterative-based.

2.10.1 Fourier-Based Reconstruction

The Radon transform defines a mathematical mapping that relates a two-dimensional object, $f(x, y)$, to its one-dimensional projections, $p(r, \theta)$, measured at different angles around the object [4, 30]:

$$p(r, \theta) = \int_0^\infty f(x, y) dl_{r, \theta} \quad (2.6)$$

where

$$r = x \cos \theta + y \sin \theta \quad (2.7)$$

and $l_{r,\theta}$ represents a straight line that has a perpendicular distance r from the origin and is at an angle θ with respect to the x -axis. It can be shown that an object can be uniquely reconstructed if its projections at various angles are known [4, 30]. Here, $p(r, \theta)$ is also referred to as *line integral*. It can also be shown that the Fourier transform of a one-dimensional projection at a given angle describes a line in the two-dimensional Fourier transform of $f(x, y)$ at the same angle. This is known as the central slice theorem, which relates the Fourier transform of the object and the Fourier transform of the object's Radon transform or projection. The original object can be reconstructed by taking the inverse Fourier transform of the two-dimensional signal which contains superimposed one-dimensional Fourier transform of the projections at different angles, and this is the so-called Fourier reconstruction method. A great deal of interpolation is required to fill the Fourier space evenly in order to avoid artifacts in the reconstructed images. Yet in practice, an equivalent but computationally less demanding approach to the Fourier reconstruction method is used which determines $f(x, y)$ in terms of $p(r, \theta)$ as:

$$f(x, y) = \int_0^\pi \int_{-\infty}^\infty p(r, \theta) \psi(r - s) ds d\theta \quad (2.8)$$

where $\psi(r)$ is a filter function that is convolved with the projection function in the spatial domain. Ramachandran and Lakshminarayanan [31] showed that exact reconstruction of $f(x, y)$ can be achieved if the filter function $\psi(r)$ in equation (2.8) is chosen as

$$\mathfrak{F}\{\psi\} = \begin{cases} |\omega| & \text{if } \omega \leq \omega_o \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

where $\mathfrak{F}\{\psi\}$ represents the Fourier transform of $\psi(r)$ and ω_o is the highest frequency component in $f(x, y)$. The filter function $\psi(r)$ in the spatial domain can be expressed as:

$$\psi(r) = 2\omega_o^2 \left(\frac{\sin 2\pi \omega_o r}{2\pi \omega_o r} \right) - \omega_o^2 \left(\frac{\sin \pi \omega_o r}{\pi \omega_o r} \right)^2 \quad (2.10)$$

This method of reconstruction is referred to as the *filtered-backprojection*, or

the *convolution-backprojection* in the spatial domain. The implementation of FBP involves four major steps:

1. Take the one-dimensional Fourier transform for each projection.
2. Multiply the resultant transformation by the frequency filter.
3. Compute the inverse Fourier transform of the filtered projection.
4. Back-project the data for each projection angle.

However, the side effect of the ramp filtering using equation (2.9) is that high-frequency components in the image that tend to be dominated by statistical noise are amplified [32]. The detectability of lesion or tumor is therefore severely hampered by this noise amplification during reconstruction by FBP, particularly when the scan duration is short or the number of counts recorded is low. To obtain better image quality, it is desirable to attenuate the high-frequency components by using some window functions, such as the Shepp–Logan or the Hann windows, which modify the shape of the ramp filter at higher frequencies [33]. Unfortunately, the attenuation of higher frequencies in filtering process will degrade the spatial resolution of the reconstructed images, and we will briefly discuss it in Section 2.13.

2.10.2 Iterative Reconstruction

Alternatively, emission tomographic images can be reconstructed with iterative statistical-based reconstruction methods. Instead of using an analytical solution to produce an image of radioactivity distribution from its projection data, iterative reconstruction makes a series of image estimates, compares forward-projections of these image estimates with the measured projection data and refines the image estimates by optimizing an objective function iteratively until a satisfactory result is obtained. Improved reconstruction compared with FBP can be achieved using these approaches, because they allow accurate modeling of statistical fluctuation (noise) in emission and transmission data and other physical processes [34, 35]. In addition, appropriate constraints (e.g. nonnegativity) and *a priori* information about the object (e.g. anatomic boundaries) can be incorporated into the reconstruction process so that better image quality can be achieved [36, 37].

An iterative reconstruction algorithm consists of three components: (1) a data model which describes the data and acquisition artifacts, (2) an objective function that quantifies the agreement between the image estimate and the measured data, and (3) an optimization algorithm that determines the next image estimate based on the current estimate. The measured data can be modeled by

$$\mathbf{p} = \mathbf{C}\boldsymbol{\lambda} \quad (2.11)$$

where $\mathbf{p} = \{p_j, j = 1, 2, \dots, M\}$ is a vector containing values of the measured projection data (i.e. sinogram); $\boldsymbol{\lambda} = \{\lambda_i, i = 1, 2, \dots, N\}$ is a vector containing all the voxel values of the image to be reconstructed; and $\mathbf{C} = \{C_{ij}\}$ is a transformation (or system) matrix which defines a mapping (forward-projection) from $\mathbf{f}$ to $\mathbf{p}$. The elements of the matrix C_{ij} is the probability that a positron annihilation event that occurred at voxel i is detected at projection ray j . Other physical processes such as nonuniform attenuation and scattered and random effects can be incorporated into the data model in the form of additive noise that corrupted the acquired projection data. Detailed discussion of more complex data models is considered beyond the scope of this chapter. The objective function can include any *a priori* constraints such as nonnegativity and smoothness. Depending on the assumed number of counts, the objective function can include the Poisson likelihood or the Gaussian likelihood for maximization. The iterative algorithm seeks successive estimates of the image that best match the measured data and it should converge to a solution that maximizes the objective function, with the use of certain termination criteria.

Iterative reconstruction methods based on the maximum-likelihood (ML) have been studied extensively, and the expectation maximization (EM) algorithm [38, 39] is the most popular. The ML-EM algorithm seeks to maximize the Poisson likelihood. In practical implementation, the logarithm of the likelihood function is maximized instead for computational reasons:

$$L(\mathbf{p}|\boldsymbol{\lambda}) = \sum_{j=1}^M \left[\ln \left(\sum_{i=1}^N C_{ij} \lambda_i \right) - \sum_{i=1}^N C_{ij} \lambda_i \right] \quad (2.12)$$

The EM algorithm updates the image values by

$$\lambda_i^{k+1} = \frac{\lambda_i^k}{\sum_{j'=1}^M C_{ij'}} \sum_{j=1}^M C_{ij} \frac{p_j}{\sum_{i'=1}^N C_{i'j} \lambda_{i'}^k} \quad (2.13)$$

where λ^k and λ^{k+1} are the image estimates obtained from iterations k and $k + 1$, respectively. The ML-EM algorithm has some special properties:

- The objective function increases monotonically at each iteration, i.e.

$$L(\mathbf{p}|\lambda^{k+1}) \geq L(\mathbf{p}|\lambda^k).$$

- The estimate λ^k converges to an image $\tilde{\lambda}$ that maximizes the log-likelihood function for $k \rightarrow \infty$ and
- All successive estimates λ^k are nonnegative if the initial estimate is non-negative.

The major drawback of iterative reconstruction methods, however, has been their excessive computational burden, which has been the main reason that these methods are less practical to implement than FBP. Considerable effort has been directed toward the development of accelerated reconstruction schemes that converge much rapidly. The ordered subsets EM (OS-EM) algorithm proposed by Hudson and Larkin [40] which subdivides the projection data into “ordered subsets” has shown accelerated convergence of at least an order of magnitude as compared to the standard EM algorithm. Practical application of the OS-EM algorithm has demonstrated marked improvement in tumor detection in whole-body PET [41].

A problem with iterative reconstruction algorithms is that they all produce images with larger variance when the number of iterations is increased. Some forms of regularization are required to control the visual quality of the reconstructed image. Regularization can be accomplished by many different ways, including post-reconstruction smoothing, stopping the algorithm after an effective number of reconstruction parameters (number of iterations and subsets for OS-EM), and incorporation of constraints and *a priori* information as described earlier. However, caution should be taken when using regularization because too much regularization can have an adverse effect on the bias of the physiologic parameter estimates obtained from kinetic modeling, which will be described later in this chapter. Nevertheless, with the development of fast algorithm and the improvement in computational hardware, application of iterative reconstruction techniques on a routine basis has become practical.

2.11 Data Corrections

Since one of the unique features of PET is its ability to provide quantitative images that are directly related to the physiology of the process under study, accurate data acquisition and corrections are required before or during the reconstruction process in order to achieve absolute or relative quantification.

2.11.1 Detector Normalization

A modern PET scanner consists of multiple rings of many thousands of detector. It is not possible that all detectors have the same operation characteristics due to differences in exact dimensions of the detectors, the optical coupling to the PMTs, and the physical and geometrical arrangement of the detectors. In other words, it means that different detector pairs in coincidence will register different counts when viewing the same emitting source. Therefore, the entire set of projection data must be normalized for differences in detector response. The normalization factors can be generated for each coincidence pair by acquiring a scan in the same way as blank scan, with a rotating rod source of activity orbits at the edge of the FOV of the gantry. Adequate counts must be acquired to prevent noise propagation from the normalization scan into the reconstructed image.

2.11.2 Dead-Time Correction

During the period when a detector is processing the scintillation light from a detected event, it is effectively “dead” because it is unable to process another event. Since radioactive decay is a random process, there is a finite probability that an event occurs at a given time interval. If an event occurs during the interval when the detector is “dead,” it will be unprocessed, resulting in a loss of data. Such loss of data is referred to as *dead-time loss*. As count rate increases, the probability of losing data due to dead-time increases. Dead-time losses are not only related to the count rates but also depend upon the analog and digital electronic devices of the system. To correct for dead-time, one can plot the measured count rate of a decaying source over time. If the source is a single radionuclide, one can calculate the count rate from the half-life of the

radionuclide and plot this against the measured count rate. Such a plot is linear at low radioactivity (hence low count rate), but nonlinearity is apparent when the count rate increases because the measured number of counts will be less than the expected number. The ratio of the measured to the expected number of counts will give an estimate of dead-time.

2.11.3 Scatter Correction

Compton scattering is one of the major factors that limits the quantitative accuracy of PET and SPECT. Some degree of scatter rejection can be accomplished, using scintillation detectors of higher density so that the number of photoelectric interactions can be maximized. However, Compton scattering of photons is unavoidable within human tissue, causing the location of the positron annihilation to be mispositioned. This leads to a relatively uniform background and reduction in image contrast and signal-to-noise ratio. For two-dimensional data acquisition, the contribution of scatter to the reconstructed image is moderate and in many cases it is ignored. In three-dimensional imaging, 35–50% of detected events are scattered and correction is essential. There are four major categories of scatter correction methods:

- empirical approaches that fit an analytical function to the scatter tails outside the object in projection space [42], and a direct measurement technique that takes the advantage of differences between the scatter distribution with septa in-place and the scatter distribution with septa retracted [42];
- multiple energy window techniques which make use of energy spectrum to determine a critical energy above which only scattered photons are recorded [43];
- convolution or deconvolution methods which model scatter distribution with an integral transformation of the projections recorded in the photo-peak window [44], and
- simulation-based methods which model the scatter distribution based on Monte Carlo simulation [45].

Details of all these methods are beyond the scope of this text.

2.11.4 Randoms Correction

As mentioned before, the basis of PET imaging is the coincidence detection scheme, which registers a coincidence event (as well as LoR) if two photons are detected within the coincidence timing window. This finite timing window (typically 12 ns for BGO), however, cannot prevent the coincidence detectors from registering random events that occur when two unrelated photons do not originate from the same site of positron annihilation. The rate of registering random coincidences by a detector pair relates to the rate of single events on each detector and the width of the timing window. The random rate for a particular LoR, R_{ij} , for a given pair of detectors i and j is

$$R_{ij} = 2\tau \times S_i \times S_j \quad (2.14)$$

where S_i and S_j are the rate of single events of detector i and detector j , and 2τ is coincidence timing window. As the radioactivity increases, the event rate in each detector also increases. The random event rate will increase as the square of the activity and therefore correction for random coincidences is essential.

The most commonly used method for estimating the random coincidences is the delayed coincidence detection method which employs two coincidence detection circuits with an offset inserted within their coincidence timing windows. The first coincidence detection circuit (called *prompt circuit*) is used to measure the prompt coincidences, which equal the sum of the true coincidences and the random coincidences. The second circuit is set up with an offset which is much longer than the time width of the coincidence window. Because of the offset in timing window, the second circuit records the so-called *delayed* coincidences which are random events, whereas all true coincidences are effectively discarded. To correct for random coincidences, the counts obtained from the delayed circuit are subtracted from those obtained from the prompt circuit. The resultant prompt events are then the “true” coincidences. However, because the random events obtained from the first circuit are not exactly the same as those obtained from the delayed circuit, subtraction of random events increases the statistical noise.

2.11.5 Attenuation Correction

One of the most important data correction techniques for PET (and also SPECT) studies is the correction for attenuation. Although the basic principles of image

reconstruction in emission computed tomography (PET and SPECT) are the same as transmission tomography (X-ray CT), there is a distinct difference in these two modalities on the data to be reconstructed. In X-ray CT, image reconstruction gives attenuation coefficient distribution of a known source while scattering is usually ignored. In PET (and SPECT), image reconstruction provides the number of photon emissions from unknown sources at unknown positions, and the photons have gone through attenuation by unknown matter (tissue) before they are externally detected. Therefore, attenuation correction factors must be estimated accurately to recover the original signals.

Attenuation occurs when high-energy photons emitted by the radiopharmaceutical in the patient are scattered and/or absorbed by the matter (tissue) between the detector and the emission site of the photon(s). The fraction of photon absorbed depends on a number of factors, including density and thickness of the intervening tissue, and photon energy. Typically, the attenuation coefficients (at 511 keV) for bone, soft tissue, and lungs are 0.151 cm^{-1} , 0.095 cm^{-1} , and 0.031 cm^{-1} , respectively.

Mathematically, the fraction of photons that will pass through a matter with linear attenuation coefficient μ is:

$$\Gamma = \exp(-\mu x) \quad (2.15)$$

where x is the thickness of the matter. If the matter is made up of different materials, then the total fraction of photons that passes through the matter would be the sum of the attenuation coefficients for each material multiplied by the thickness of the material that the photons pass through:

$$\Gamma = \exp\left(-\sum_i \mu_i x_i\right) \quad (2.16)$$

where μ_i is the attenuation coefficient of the i th material and x_i is the thickness of the i th material that the photons pass through. Accordingly, if a detector measures N_0 counts per unit time from a source without attenuation (for example, in air, where the attenuation coefficient is close to zero), the attenuated counts, N , after placing a matter with varying linear attenuation coefficient in between, is:

$$N = N_0 \exp\left(-\int_0^d \mu(x) dx\right) \quad (2.17)$$

where $\mu(x)$ is a distance-dependent attenuation coefficient function which

accounts for the varying attenuation within the matter, and d is the distance between the source and the detector (in cm). Therefore, in PET and SPECT, attenuation artifacts can cause a significant reduction in measured counts, particularly for deep structures. For example, attenuation artifacts can resemble hypoperfusion in the septal and inferior-posterior parts of the myocardium in cardiac PET or SPECT study. Failure to correct for attenuation can cause severe error in interpretation and quantitation. As the attenuation coefficient varies with different tissue types, the extent of photon attenuation/absorption will also vary even though the distance between the emission site of the photons and the detector remains unchanged. Therefore, spatial distribution of attenuation coefficients, i.e. an attenuation map, is required for each individual patient in order to correct for photon attenuation accurately.

Consider the attenuation in an object whose total thickness is D , measured along the LoR, and the attenuation coefficient is μ , as shown in Fig. 2.7. If the annihilation event occurs at position x , measured along the LoR, then the probabilities for the two gamma rays to reach the opposing detectors are $e^{-\mu(D-x)}$ and $e^{-\mu x}$, respectively. The probability of registering the coincidence event is the product of the probabilities of detection of the gamma rays by the opposing detectors, i.e. $e^{-\mu(D-x)} \cdot e^{-\mu x} \equiv e^{-\mu D}$, which is independent of the source position, x . This remains true when the attenuation coefficient is not uniform within the cross-section of the body. Thus, the attenuation is always the same even if the source position is outside the object.

The measured projection data will differ from the unattenuated projection data in the same fashion. Suppose $\mu(x, y)$ denotes the attenuation coefficient

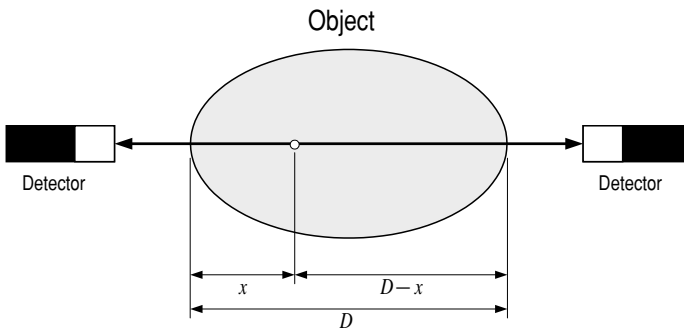


Figure 2.7: Attenuation of the gamma rays in an object for a given line of response.

map of the object, the general equation for the attenuated projection data can be described by the attenuated Radon transform

$$p_m(r, \theta) = \int_0^\infty f(x, y) \exp \left(- \int_0^{l(x, y)} \mu(x', y') ds \right) dl_{r, \theta} \quad (2.18)$$

where $p_m(r, \theta)$ is the measured projection data, $l(x, y)$ is the distance from the detector to a point (x, y) in the object, while $l_{r, \theta}$ and r have the same definitions as in equations (2.6) and (2.7). It should be noted that unlike the unattenuated Radon transform as in equation (2.6), there is no analytical inversion formula available for equation (2.18).

The attenuation correction in PET is simpler and easier as compared to SPECT due to the difference in the photon detection schemes. In SPECT, the attenuation depends not only on the source position, but also on the total path length that the photon travels through the object. It is not straightforward to correct for attenuation or find an inversion of equation (2.18) for image reconstruction. On the contrary, the attenuation in PET is independent of the source position because both gamma rays must escape from the body for external detection and the LoR can be determined. Therefore, the exponential term in equation (2.18) can be separated from the outer integral. The unattenuated projection data and the measured projection data can then be related as follows:

$$p_m(r, \theta) = p(r, \theta) p_\mu(r, \theta) \quad (2.19)$$

where $p(r, \theta)$ is the unattenuated projection data, and

$$p_\mu(r, \theta) = \exp \left(- \int_0^\infty \mu(x, y) dl_{r, \theta} \right) \quad (2.20)$$

is the projection data of the attenuation map. Therefore, if the attenuation coefficient map $\mu(x, y)$ or its projection data $p_\mu(r, \theta)$ is known, then the unattenuated projection data $p(r, \theta)$ of the object can be calculated as:

$$p(r, \theta) = \frac{p_m(r, \theta)}{p_\mu(r, \theta)} \quad (2.21)$$

and $f(x, y)$ can then be reconstructed without attenuation artifacts.

Since the attenuation is always the same regardless of the source position inside the FOV, it is possible to use an external (transmission) positron-emitting source that comprises a fixed ring or rotating rod sources, to measure the attenuation correction factors through two extra scans: *blank scan* and *transmission scan*. A blank scan is acquired with nothing inside the FOV, and a transmission

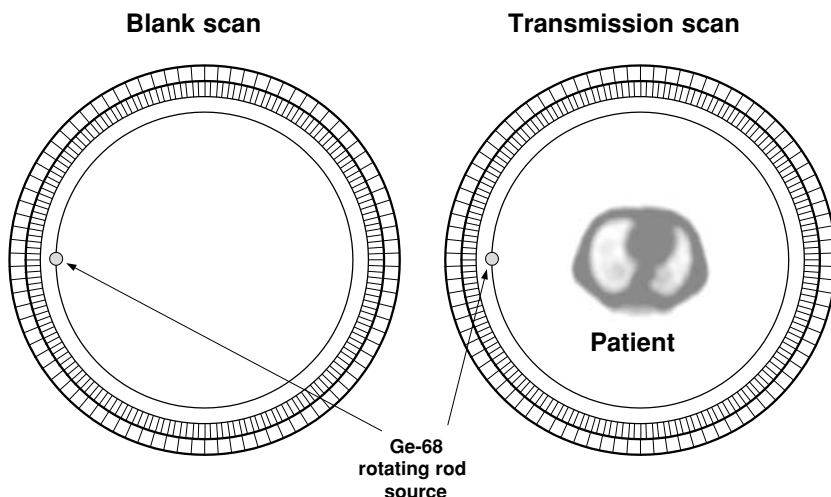


Figure 2.8: Attenuation correction in PET using a rotating rod source of ^{68}Ge . Blank and transmission scans are generally acquired before tracer administration.

scan is acquired to measure the coincidence rate when the patient being imaged is in the FOV but has not been given an injection of positron emitter. Figure 2.8 shows a schematic for measured attenuation correction using a rotating rod source of positron emitter ^{68}Ge . Attenuation correction factors are then determined by calculating the pixelwise ratio of the measured projection data obtained from the blank scan and the transmission scan. The major drawback of this approach is that statistical noise in the transmission data would propagate into the emission images [46, 47]. Therefore, transmission scans of sufficiently long duration have to be acquired to limit the effect of noise propagation. Depending on the radioactivity present in the external radiation source and on the dimension and composition of the body, transmission scans of 15–30 min are performed to minimize the propagation of noise into the emission data through attenuation correction, at the expense of patient throughput. Further, lengthened scan duration increases the likelihood of patient movement, which can cause significant artifacts in the attenuation factors for particular LoRs.

Application of analytical, so-called calculated attenuation correction eliminates the need for a transmission scan, thus making this method attractive in many clinical PET centers. This method assumes uniform skull thickness and constant attenuation in the brain and skull. However, such assumptions

do not hold for sections that pass through sinuses and regions where the adjacent bone is much thicker. Alternatively, the transmission scan may be performed after tracer administration, referred to as postinjection transmission (PIT) scanning [48], which utilizes strong rotating rod (or point) sources for the transmission source. A small fraction of “transmission” coincidences contains in the sinogram data can be distinguished from emission coincidences that originate from the administered radiopharmaceuticals by knowing the positions of the orbiting sources. Another approach is to integrate measured and calculated attenuation that makes use of the advantages of each approach. A transmission scan is still required and the attenuation coefficient images derived from the transmission and blank scans are reconstructed and then segmented into a small number of tissue types, which are assigned with *a priori* known attenuation coefficients [49–51]. These processes greatly reduce noise propagation from the transmission data into the reconstructed emission images.

2.12 Calibration

Once the acquired data has been corrected for various sources of bias introduced by different physical artifacts as mentioned in the previous section, images can be reconstructed without artifacts, provided that there are sufficient axial and angular sampling of projection data. To reconstruct images in absolute units of radioactivity concentration (kiloBecquerel per milliliter, kBq/mL, or nanoCurie per milliliter, nCi/mL), calibration of the scanner against a source of known activity is required. This can be accomplished by scanning a source of uniform radioactivity concentration (e.g. a uniform cylinder) and then counting an aliquot taken from the source in a calibrated well-counter to obtain the absolute activity concentration, which is then compared to the voxel values in the reconstructed images for the source (after corrections for physical artifacts have been applied) to determine a calibration factor.

2.13 Resolution Limitations of PET

Although there has been significant improvement in PET instrumentation over the last two decades, there is a finite limit to the spatial resolution of PET scanner.

In this section, we briefly describe some of the major factors that directly relate to the spatial resolution of PET. Spatial resolution is defined as the ability of the scanner to depict small objects and is limited by a number of factors:

- distance the positron travels before it annihilates;
- annihilation photon noncollinearity due to residual momentum of the positron;
- intrinsic resolution and size of the detectors;
- stopping power (and material) of the detector;
- sampling requirements, and
- image reconstruction parameters (e.g., reconstruction filter, matrix size, reconstruction algorithm, etc).

The finite distance travelled by the positron before annihilation also has adverse effects on the spatial resolution of PET scanner [52]. This distance is referred to as the *positron range* which varies from fraction of a millimeter to several millimeters, depending on the density of the tissue in which the emission occurs and the maximal positron energy of the radionuclide (Eq. (2.3) and Table 2.1). It is apparent that a positron with higher energy can travel farther from the nucleus before annihilation occurs. This effect leads to a blurring of the data which is characterized by an exponential function with a FWHM of the order of 0.2–3 mm for most positron-emitting isotopes.

Another factor which can degrade the spatial resolution is caused by the residual kinetic energy and momentum possessed by the positron and the electron (because both of them are moving) when they annihilate. The apparent angle between the two emitted photons deviates slightly from 180° for about 0.5° FWHM. The degradation in resolution due to this photon noncollinearity effect depends on the diameter of the detector ring of the PET scanner. This effect, and the positron range, imposes a lower limit of the spatial resolution which is approximately 3 mm for human PET imaging and 1 mm for a small-diameter animal PET system.

The intrinsic resolution of the detectors is the crucial factor which determines the spatial resolution of modern PET scanners. For arrays of a single-element detector of width D , the resulting coincidence point spread function is

triangular with spatial resolution approximately $D/2$ [53]. It is therefore apparent that a small detector must be used, in order to achieve high spatial resolution [54]. Most of the modern clinical PET scanners utilize multiple rings of BGO block detectors to simultaneously achieve high spatial resolution and sensitivity (Section 2.7). BGO crystals are commonly used in commercial PET systems because they have high stopping power (high efficiency) for the 511 keV gamma rays and high spatial resolution (≈ 5 mm which is near the theoretical limit of resolution), and are 50% more efficient than NaI(Tl) crystals. However, the major disadvantages of BGO crystals are that their photofluorescent decay time is very long ($0.3 \mu\text{s}$) which causes countrate limitations and that they have lower light output. During the last decade, many scintillators have been explored and some of them are currently in use in new generation of PET scanners. The best known ones are barium fluoride [55] and gadolinium oxyorthosilicate [56]. Block detectors are also being developed with *lutetium oxyorthosilicate* (LSO) [57], a new detector material which has much shorter photofluorescent decay time and provides higher spatial resolution images. The images obtained with PET device built from LSO detectors are much sharper and they can be acquired at a much faster rate than current PET scanners. Therefore, faster scans and higher patient throughput can be achieved. Many of PET centers in the world have installed, or planned to install, the latest generation of LSO-based PET scanner such as the ECAT HRRT system (CTI/Siemens, Knoxville, TN).

Spatial resolution is also affected by the coincidence events detected by the PET scanner, as described in Section 2.8. Image reconstruction algorithms also have an impact on the spatial resolution that can be achieved with modern PET scanner. The statistical nature of radioactive decay described by Poisson distribution produces noise in the PET measurements. This noise can be amplified by the reconstruction process and visualized in the reconstructed images due to its high-frequency nature. In order to suppress noise in the reconstructed images with FBP, the projection data (or sinogram) has to be filtered with a ramp filter (in frequency domain) before the reconstruction process [31, 58]. However, the side effect of the ramp filtering is that high-frequency components in the image that tend to be dominated by statistical noise are amplified [32]. To obtain better image quality, it is desirable to attenuate the high-frequency components by using some window functions, such as the Shepp–Logan or the Hann windows, which modify the shape of the ramp filter at higher frequencies [33]. Although the use of window functions can help control the image noise and thereby increase

the signal-to-noise ratio, the spatial resolution must degrade because reducing the higher frequencies is equivalent to smoothing the image data. A recent study found that it is difficult to select a filter for FBP based on some objective criteria [59]. The trade-off between filter selection (or image noise) and spatial resolution is therefore dependent on the preferences of the clinical physicians and the types of PET study.

As the overall resolution of PET imaging system is a convolution of all of the above components, it is therefore important to note that improvement in resolution by considering only one of the above components in designing a high-resolution PET imaging system will produce limited improvement in the resulting spatial resolution [60].

2.14 Quantitative Physiological Parameter Estimation

PET is a unique and state-of-the-art imaging tool in that it is able to target specific biochemical or physiologic processes using short-lived radiopharmaceuticals (or radiotracers) of major elemental constituents of the body, such as carbon, fluorine, nitrogen, and oxygen; and that the time course of radiotracer in the body can be recorded and reconstructed as multidimensional images which represent the radiotracer distribution in the body. With an appropriate reconstruction algorithm and with proper corrections for physical artifacts, quantitatively accurate radiopharmaceutical distribution can be obtained and calibrated in absolute units of radioactivity concentration (in kBq/mL or nCi/mL). However, the radiotracer distribution in the body is not static over the course of the study, but varies with time, depending on the different processes that govern its delivery, uptake, and subsequent biologic fate in the body. By acquiring a dynamic sequence of tomographic images, quantitative information of the time-varying radiopharmaceutical distribution can be obtained which can be used to provide *in vivo* measurements of specific physiologic functions.

2.14.1 Tracer Kinetic Modeling

Radiotracers provide a means for investigation of biochemical or physiologic processes without altering the normal functions of the biologic system. Each radiotracer must be targeted to provide a physiologic parameter of interest, such

as blood flow, glucose metabolism, oxygen utilization, protein synthesis, and receptor or binding site density, etc. in the body. The concentration of the radiotracer introduced into the biologic system is assumed to be negligible so that it does not perturb the natural process of the system. Otherwise, the measurement does not represent the process we want to measure but the effect induced by the introduction of the radiotracer. External measurable data is the time course of total tissue activity concentration obtained from the PET images, and the time course of blood (or plasma) activity concentration (i.e. the input function of the compartment model), obtained from peripheral blood sampling. These curves are described as *time-activity curves* (TACs), where the term “activity” refers to *concentration* of the radiotracer rather than the tissue (or blood) activity. Yet, the measured time course of tracer uptake and delivery does not directly provide quantitative information about the biologic and physiologic processes but the kinetic information of the radiotracer. Mathematical modeling of the measured tracer kinetics is thus required to transform the kinetic information into physiologically meaningful information, i.e. the physiologic parameters of interest. This can be accomplished through the use of an analysis technique commonly referred to as *compartmental* or *tracer kinetic modeling*.

Mathematical modeling of biologic processes and systems is well established and a wide variety of models have been developed [61]. Although nonlinear models should be used to study biological systems which are commonly nonlinear, linear compartmental models have properties which make them attractive for radiotracer experiments with PET and SPECT [62]. A given system can be described by a compartment model, which consists of a finite number of interconnected compartments (or pools), each of which is assumed to behave as a distinct component of the biologic system with well-mixed and homogeneous concentration [63]. An example is shown in Fig. 2.9 for [^{18}F]fluorodeoxyglucose, which is the primary radiopharmaceutical used in PET to assess glucose metabolism. A compartment can be a physical space, such as plasma or tissue, or a chemical entity, where tracer may exist in different forms

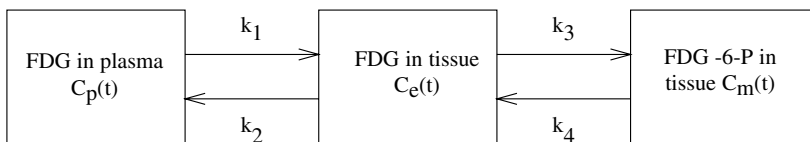


Figure 2.9: The three-compartment model for transport and metabolism of [^{18}F]fluorodeoxyglucose (FDG).

(FDG and its phosphorylated form FDG-6- PO_4). The compartments of a tracer kinetic model are linked by a set of parameters called *rate constants*, k_i , which represent the rates at which the radiotracer in one compartment is transported to the connected compartments. More precisely, these rate constant parameters represent specific physiologic or biochemical processes (e.g. flow or transport across physical spaces, or rates of transformation from one chemical form to the other in a chemical entity) within the biologic system. For the FDG model as shown in Fig. 2.9, the three compartments represent (from left) vascular space for FDG, tissue space for free FDG, and tissue space for FDG-6-phosphate (FDG-6-P). The rate constants describe the movement of FDG between compartments: k_1 and k_2 for the forward and backward transport of FDG across the blood-brain barrier, k_3 for the phosphorylation of FDG to FDG-6-P, and k_4 for the dephosphorylation of FDG-6-P back to FDG.

The aim of modeling is to interpret the fate of the administered radiotracer quantitatively in terms of the standard parameters in the compartmental model. In conjunction with knowledge of the transport and metabolism of the radiotracer, it is possible to relate the rate constants to physiologic parameters of interest. Figure 2.10 summarizes the key steps in physiologic parameter estimation

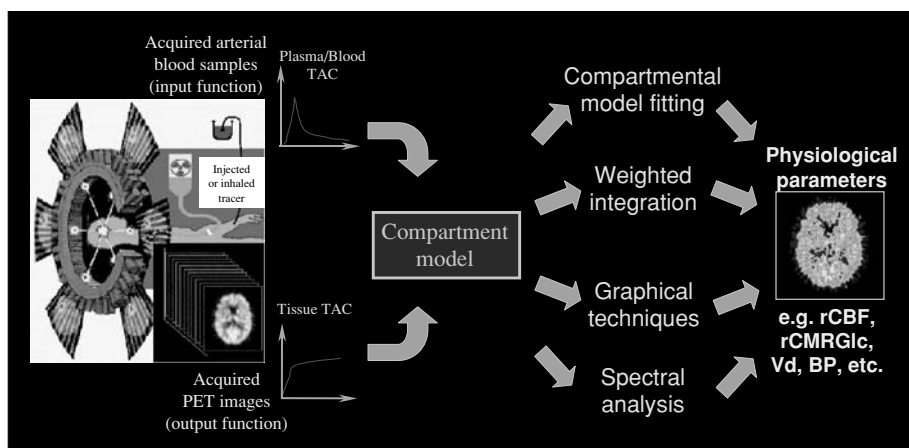


Figure 2.10: (Color slide) Quantitative physiological parameter estimation with PET includes radiotracer administration, data acquisition with a PET scanner, measurement of tracer plasma concentration, a suitable mathematical model and a parameter estimation method to estimate the physiological parameter of interest.

in a quantitative PET study. After radiopharmaceutical administration, PET data is acquired at a predefined sampling schedule and individual voxel values in the reconstructed images represent the localized radiotracer time-activity concentration in the body upon correction for some degrading factors (e.g. attenuation and scatter) and cross-calibration. A vector formed by extracting a voxel curve from the sequence of images corresponds to a tissue TAC, which represents the response of the local tissue as a function of time after the tracer administration. Alternatively, the tissue TAC can be obtained by manual delineation of region of interest (ROI) on the reconstructed PET images. Plasma tracer concentration is typically measured by means of arterial blood sampling. A mathematical model is applied to the tissue and the plasma tracer concentration to estimate the physiological parameters of interest. Kinetic modeling approaches based on the framework of tracer kinetic modeling could be applied to estimate the physiologic parameters. The same analysis procedures can also be applied to dynamic SPECT without loss of generality, although the challenges tend to be much greater in SPECT.

2.14.2 Compartmental Model Fitting

As the rate of tracer exchanges is assumed to be proportional to the amount of tracer in the compartment, a system of first-order differential equations can be derived. Compartmental model fitting performs the mathematical estimation process to seek the values for the rate constant parameters that provide a best fit for the observed tissue and blood (or plasma) TAC to the predefined compartmental model. The general solution to the system of first-order differential equations has the form

$$C_T(t) = (1 - V_B) \left[\sum_{j=1}^N \alpha_j e^{\beta_j t} \otimes C_p(t) \right] + V_B C_a(t) \quad (2.22)$$

where $C_T(t)$ is the measured time course of total tissue activity concentration (or tissue TAC) in the FOV, $C_p(t)$ is the input function of the compartment model, $C_a(t)$ is the tracer concentration in whole blood, N is the number of tissue compartments assumed for the radiotracer, α_j and β_j are the fitted parameters from which the rate constant parameters, k_i , of the compartment model can be derived, V_B is the fraction of the measured volume occupied by the blood pool ($0 \leq V_B \leq 1$), and $\otimes$ is the convolution operator. It is assumed that $C_p(t)$

and $C_a(t)$ have been corrected for delay and dispersion. The term $V_B C_a(t)$ in Eq. (2.22) represents intravascular activity present in the FOV of the scanner and is particularly important at the times immediately following tracer administration. Note that although $C_T(t)$ has an analytical representation, it is a nonlinear function of parameters α_j and β_j , and therefore nonlinear parameter estimation approach such as the nonlinear least-squares (NLLS) method using Gauss–Newton type algorithms [64] is required to estimate the parameters α_j and β_j , unless the model equation is “linearized” by certain transformations. Alternatively, the rate constant parameters in the system of differential equations can be estimated directly by numerical method.

Ideally, a comprehensive model describing the full kinetics of a tracer is desirable, but this may require a complicated compartmental model configuration with many parameters and compartments. Given that the counting statistics of the acquired data and the spatial resolution of the imaging device are limited, the compartments and parameters in the complicated compartmental model cannot be resolved and identified from the measured data. Thus, the actual compartmental model configuration must be simplified. Typically, measured data from PET and SPECT can support compartmental models with six parameters or less. In general, including more number of compartments (or parameters) or increasing the complexity of the model improves the fit to the measured data. However, the improvement in the fitting may not be statistically significant. In many cases, the reliability of individual parameter estimates degrades with increasing number of compartments (or model complexity). Therefore, the model with the smallest number of compartments which can fully describe the measured data should be used.

2.14.3 Input Function

For compartmental models used in PET, one of the compartments represents blood pool or extravascular space. This compartment can be seen as the input stage to the model because the tracer is delivered into the tissue through the blood, as indicated by Eq. (2.22), and therefore the time-activity concentration for blood is called *input function*. In other words, the time-activity concentration of the tracer in a particular tissue is dependent both on the amount of the tracer delivered to the tissue via the blood supply (the input function) and on the

exchanges of the tracer within the tissue compartment (the impulse response function of the tissue).

Ideally, the input function should be measured in the capillaries of the tissue of interest but this is not possible. Instead, samples are obtained by frequent blood sampling at the peripheral artery such as radial artery or femoral artery, under local anesthesia. The arterial concentration of the tracer has to be measured because the concentration of the tracer in venous blood may differ markedly from that in arterial blood. This is particularly true for tracer with high extraction fractions. Unless the tracer can freely diffuse between red blood cells and plasma, the concentration time course in plasma is required for the input function. Another consideration is the *in vivo* formation of radiolabeled metabolites that can contribute to the radioactivity counts in total plasma or whole-blood, necessitating a time-dependent estimation of the fraction of radiolabeled metabolites present in plasma. Estimation of radiolabeled metabolites in plasma is often possible by means of chromatographic analysis (e.g. using HPLC, TLC, or octanol extraction) for the measured samples. Modeling of metabolites formation is also possible [65,66], but the statistical quality of the measurements may not support reliable estimation of the additional parameters in the more complicated compartment model.

Arterial blood sampling is currently regarded as the *gold standard* method of measuring the input function for PET study in spite of many possible sources of error such as insufficient sampling rates and counting errors in the blood samples. Particularly for radiotracers with rapid kinetics and short measurement times, delay and dispersion effects between the blood sampling site and the actual arterial blood that drives the tissue of interest need to be taken into account during the compartmental model fitting. The procedure of arterial catheterization under local anaesthesia, however, is very invasive in nature and would involve potential risks of arterial sclerosis and ischemia to the distal extremity. An alternative, yet less invasive, approach is to obtain blood samples from arterialized-vein (or “hot” vein). This method is referred to as the *arterialized-venous* (a-v) sampling method. The hand is heated in a water bath which then promotes arterio-venous shunting to avoid the discomfort and risks associated with arterial cannulation [67,68]. Since it only requires the placement of venous cannulas, it is less invasive and is better tolerated than arterial cannulation by the subjects. However, it requires prolonged hand warming to ensure adequate shunting, and it is very dependent upon the site chosen for the placement of the

venous cannula and the rate of blood flow. The best results are usually obtained from a cannula that is placed in a large vein on the dorsum of the hand, typically in a muscular male. As mentioned earlier in this section, tracer concentration in venous blood may differ markedly from that in arterial blood, especially for tracer with high extraction fractions, and the use of venous blood as the input function could introduce significant errors in absolute quantification of physiological parameters.

When a suitable vascular structure, such as a large artery or the left ventricle, is contained within the FOV, the input functions can be derived noninvasively from the image data and the need for blood sampling is completely eliminated. However, careful correction for the extravascular contamination in the blood region is required to obtain an accurate approximation to the input function. In certain circumstances, if regions can be identified with the same input but with different kinetic behavior (e.g. gray and white matter in the brain), then the compartment model can be reformulated to solve for the kinetic parameters of interest and the required input function [69–71]. However, these approaches may have very high computational complexity as the number of parameters to be estimated (kinetic parameters and the input function) increased drastically and careful selection of parameter estimation algorithm may be required.

In the case of receptor studies, the need for measuring arterial input function may be avoided if a tissue region can be identified which is devoid of specific binding but shows similar nonspecific binding to the tissues of interest. Typically, the cerebellum is used as a reference tissue region in the brain for a number of neuroreceptor systems. Once the reference tissue region is identified, the compartmental model can be reformulated to use the reference tissue TAC as the input function [72–74]. The use of reference tissue also eliminates the need to estimate the fraction of labeled metabolites present in the plasma.

2.14.4 Parametric Imaging

PET (and SPECT) provide multidimensional images of the time-varying radiopharmaceutical distribution. The aforementioned compartmental analysis has concentrated only on fitting tissue TACs derived from ROIs to a predefined compartment model. Although it is possible to generate TACs for each voxel of the image data and fit them to the model, this is not practical as many thousands of model fit using NLLS, which is computationally expensive, would have to be performed. Further, the high noise levels and heterogeneity in individual

voxel curves make the compartmental model fitting approach unappealing. A number of “fast” estimation techniques have been developed to reduce the computational load and improve the reliability of estimating images whose voxels represent parameters of interest, commonly known as *parametric images*. A brief description of these estimation techniques follows.

2.14.5 Linearization Approaches

Linearization approaches reformulate the model equations so that (1) a linear relationship exists between the transformed data and the primary physiological parameter of interest, or (2) the reformulated model equations contain only linear parameters. In these circumstances, estimation of parameters can be accomplished by a simple linear regression or by linear least-squares (LLSs) techniques.

A number of *graphical techniques* that aim at transforming the measured data into a plot which is linear after a certain “transformed time” have been proposed for specific tracer studies, including the Patlak [75, 76], Logan [77, 72], and Yokoi [78, 79] plots. Applications of the techniques depend on the tracer studies and parameter of interest. The Patlak plot [75] was initially developed for estimating the influx rate constant of radiotracer accumulation in an irreversible compartment, and was extended to allow for slow clearance from the irreversible compartment [76]. When employed in FDG studies, the influx rate constant is directly proportional to the regional metabolic rate of glucose. The Logan plot [77, 72] was primarily developed for estimation of parameters related to receptor density such as binding potential and volume of distribution for neuroreceptor studies and the radiotracers can have reversible uptake. The Yokoi plot [78, 79] has been proposed as a rapid algorithm for cerebral blood flow measurements with dynamic SPECT. Although all these methods permit the estimation of physiologic parameter in rapid succession and have been used extensively because of their computational simplicity, the bias introduced into the physiologic parameters is significant in the presence of statistical noise in the image data.

The use of linearized model equations was first proposed by Blomqvist [80] for the Kety–Schmidt one-compartment model used for measuring cerebral blood flow [81] and was extended by Evans [82] for the three-compartment model (as shown in Fig. 2.9) to measure cerebral metabolic rate of glucose. The key idea is that by reformulating and integrating the model equations, the

operational equations will be linear in the parameters to be estimated, whereby linear least-squares or weighted linear least-squares methods can be used to estimate the parameters of interest. While the measurement errors are typically statistically independent in time, integration introduces correlation of measurement errors, which can introduce bias into the parameter estimates [83]. The generalized linear least-squares method was designed to remove bias in the estimates resulting from integration of measurements and has been extended to multicompartment models and has been found useful in fast generation of parametric images [84–86].

2.14.6 Spectral Analysis

In compartmental model fitting, the number of compartments and their interconnection are defined *a priori*. This implies that the physiological or biochemical pathways are somewhat known. Yet, *a priori* knowledge about the behavior of novel anticancer drugs may not be available. Further, the compartmental modeling approach assumes well-mixed, homogeneous tracer distribution within the tissue or the ROI. This may not be true for tumor which normally has high degree of heterogeneity. *Spectral analysis* does not rely on tracer assumptions and the number of compartments and their connectivity; it is particularly useful for tracer kinetics studies.

Spectral analysis [87] fits the model defined in equation (2.22) with a predefined set of basis functions, $e^{\beta_j t} \otimes C_p(t)$, where β_j can take on a discrete set of values so that a large number (100 or more) of basis functions are generated. The fitting to tissue data is accomplished by nonnegative least squares (NNLS) algorithm with a constraint $\alpha_i \geq 0$ [88]. Typically, a linear combination of only two or three basis functions from the complete set of basis functions are identified which can best describe the observed tissue data. From the fitted basis functions, the impulse response function and other physiological parameters can be estimated. Spectral analysis can also be applied to projection data directly, but it may not produce results equivalent to those obtained from reconstructed images because the NNLS fitting may not be linear [89].

Since spectral analysis does not require any *a priori* definition of the numerical identifiable components present in the PET data, it is more flexible than compartmental model fitting. However, the assumption on the nonnegativity coefficients of exponentials may not be valid in a generic compartmental model as

negative coefficients of exponentials are also possible if the input and output are not taken from the same compartment [90]. Furthermore, repeated eigenvalues may be inherent in the data and the impulse response function of the underlying system could have different formats [91].

2.14.7 Weighted Integration

Another approach for dynamic PET data analysis is by means of *weighted integration* or *integrated projection* methods. Different from the techniques mentioned before which calculate parameters of interest from a series of *reconstructed images* based on the framework of compartmental model fitting, these methods eliminate the need to reconstruct the dynamic projection data. Instead, the parameters of interest are calculated from the weighted integrals of the projection data based on the relationship between the time-dependent weighting functions and the projection data [92–95]. Indeed, the weighted integration approach can be applied to the reconstructed data by integrating the full set of dynamic images. Reconstruction load, however, can be significantly reduced if the weighted integration is performed on the projection data because only the integrated projection data is reconstructed. This relies on the fact that image reconstruction is a linear operation in the spatial domain and the linear operations in time is commutative with image reconstruction [96, 62]. Therefore, the order of reconstruction and integration can be interchanged. It is apparent that integration of the projection data prior to image reconstruction has an attractive computational advantage.

Typically weighted integration or integrated projection methods are used to determine the rate of blood flow and the tissue-to-blood partition coefficient. In fact, the use of weighted integration does not limit to one-compartment model such as that used to measure cerebral blood flow. With the use of additional weighting functions, physiologic parameters in two- and three-compartment models can also be estimated. As might be expected, the formulation is considerably more complicated.

2.15 Applications of PET

As PET is a physiological/functional imaging modality, which provides information about the physiology in tissue, it is therefore complementary to the

traditional tomographic imaging techniques, such as CT and MRI, that can provide anatomical (or structural) information of the tissue only. These latter techniques are method of choice when normal anatomy is expected to be disrupted by disease. However, there are many situations where functional changes precede anatomic changes or anatomic changes may be absent. Examples include cancers in their early stage, and various neurodegenerative diseases such as Alzheimer's, Huntington's, and Parkinson's diseases, epilepsy and psychiatric disorders, [97–99], in addition to a wide variety of neuroreceptor studies [100].

Historically, clinical applications of PET were centered around neurology and cardiology. The clinical role of PET has evolved considerably during the past 10 years, and it is well recognized that PET has a preeminent clinical role in oncology. Currently, oncological PET studies contribute to over 80% of clinical studies performed worldwide [101]. It is well recognized that PET is useful for monitoring patient response to cancer treatment and assessing whether lesions seen with CT and MRI are cancerous, and is capable of grading degree of malignancy of tumors, detecting early developing disease, staging the extent of disease, detecting primary site of tumor, measuring myocardial perfusion, differentiating residual tumor or recurrence from radiation-induced necrosis or chemonecrosis, and monitoring cancer treatment efficacy [102–107]. FDG is the primary radiopharmaceutical used in oncological PET studies to assess glucose metabolism. Improvements in instrumentation in the late 1980s overcame the limitation of the restricted imaging aperture and enabled three-dimensional whole-body to be imaged. Whole-body PET imaging has been proven highly accurate in the detection of a number of different malignancies, particularly in cancers of the colon, breast, pancreas, head and neck, lungs, liver, lymphoma, melanoma, thyroid, and skeletal system, depending on the use of specific radiotracers. Figures 2.11 and 2.12 show examples of neuro-oncologic and whole-body coronal FDG-PET images.

As mentioned in Section 2.3, PET offers some unique features that cannot be found in other imaging modalities. The radiolabeled compounds used in PET are usually carbon (^{11}C), nitrogen (^{13}N), oxygen (^{15}O), and fluorine (^{18}F), which can be used to label a wide variety of natural substances, metabolites, and drugs, without perturbing their natural biochemical and physiological properties. In particular, these labeled compounds are the major elemental constituents of the body, making them very suitable to trace the biological processes in the body. As the measurements are obtained noninvasively using external detectors,

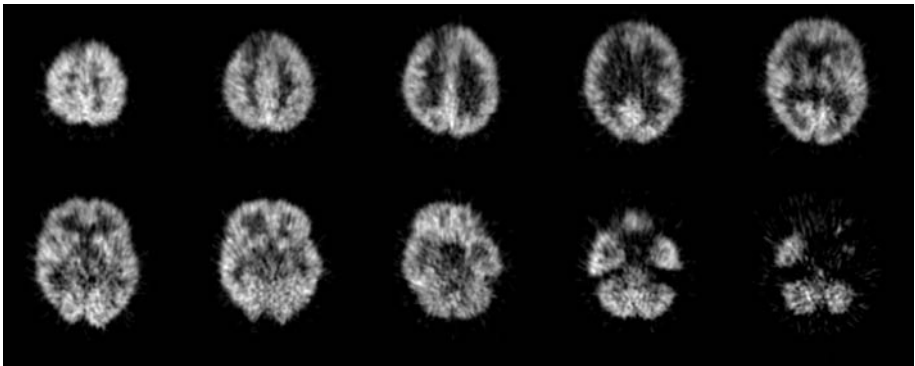


Figure 2.11: Neuro-oncologic FDG-PET images.

experiments can be performed repeatedly without sacrificing the small laboratory animals, such as mice and rats. This is not possible with *in vitro* tests which involve sacrifice of the animal at a specified time after radiotracer injection and preclude the kinetics of the radiotracer to be studied in the same animal. The greater flexibility in producing natural labeled probes for imaging on a macroscopic level in PET has raised the possibility of *in vivo* imaging on a cellular or genetic level. Recent advances in this field appear promising, particularly in the imaging of gene expression. Progress is being made and PET is expected to assume a pivotal role in the development of new genetic markers [108].

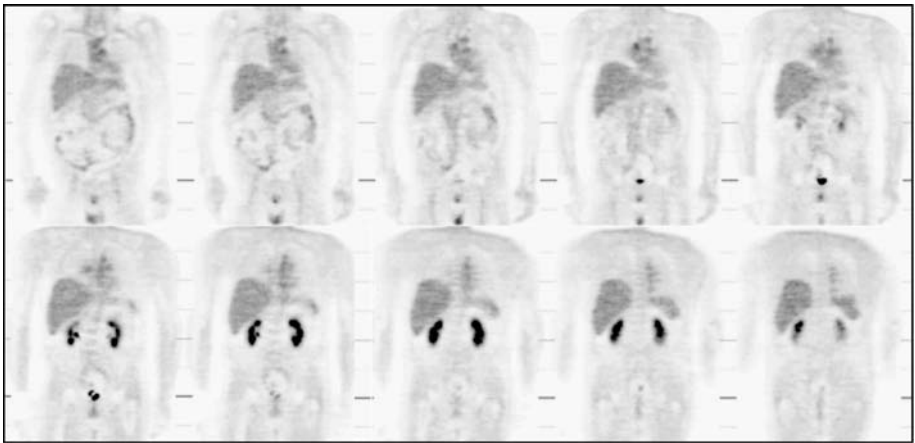


Figure 2.12: Whole-body coronal FDG-PET images.

2.16 Discussion and Concluding Remarks

This chapter presented an overview of quantitative PET imaging, including the basic principles and instrumentation, methods of image reconstruction from projections, and some specific correction factors necessary to achieve quantitative images. Techniques for absolute physiologic parameters estimation based on the framework of tracer kinetic modeling are also introduced. Recent advances in instrumentation and detector technologies have partially resolved some of the challenges. Research is still under way to develop new detector materials not only for improving the spatial resolution but also the system sensitivity, while keeping the cost of the instrument acceptable. Another active research area is image reconstruction. Although the filtered backprojection method is still widely used nowadays, it is not an optimal reconstruction technique. Iterative algorithms have been shown to improve the signal-to-noise ratio of the reconstructed images and provide more accurate image reconstruction for low count studies, but the computational complexity and appreciably long reconstruction times as compared with filtered backprojection, as well as the requirement of using some *ad hoc* techniques to control the visual quality of the reconstructed images, remain a substantial obstacle for routine implementation. It should be noted, however, that a good reconstructed image is not only dependent on the reconstruction algorithm, but also dependent on the sufficient axial and angular sampling of projection data such that the reconstruction artifacts can be minimized. In order to achieve accurate absolute or relative quantification, appropriate data corrections have to be applied prior to image reconstruction.

The use of SPECT for studying physiologic functions deserves mention here. SPECT is another form of emission computed tomography which had its beginning in the early 1960s, initiated by the work of Kuhl and Edwards on transverse- and longitudinal-section scanning with single-photon-emitting radiotracers [15], and the work of Anger on the development of scintillation camera (also known as gamma camera or Anger camera) coupled with photomultiplier tubes [108]. Although the innovative idea of how gamma rays interact with inorganic crystal (scintillator) to produce scintillation light and how optical coupling with photomultiplier tubes helps amplify the scintillation signal may shed some light on detector design in PET, the development of SPECT imaging, however, has been overshadowed by PET for reasons to be detailed. The principles for detecting

the emerging radiation and the formation of tomographic images are similar for both PET and SPECT but the underlying physics and the instruments employed are completely different. In SPECT, nuclear decay results in the emission of one or a few uncorrelated photons in the 100–200 keV energy range. A lead collimator drilled with small holes is used to mechanically collimate the incoming photons by allowing those traveling in one particular direction to interact with the scintillator, while all others are absorbed. Rotating gamma cameras with single or multiple crystal detectors are used to form a tomographic image. Here lies the sensitivity differences between PET and SPECT, and partly explains why PET has received much more attention than SPECT for *in vivo* assessment and quantification of physiologic functions in the body.

Despite the fact that both PET and SPECT suffer from attenuation and Compton scattering of the photons inside the body which can result in image artifacts and loss of quantitative accuracy, SPECT has been largely considered to be nonquantitative and limited to providing qualitative or relative functional images. This is because correction of attenuation and scatter in SPECT are not easy as compared to PET, where attenuation correction is routine (with the exception of whole-body PET). In addition, the spatial resolution of SPECT is inferior to that in PET. Even with triple-headed gamma cameras, the resolution is approximately 8–10 mm FWHM but the theoretical limit of 1–2 mm FWHM can be achieved for PET with new generation of detector technology. Further, typically higher signal-to-noise ratio and lower scatter with PET also helped establish PET as the favorable method for quantitative measurements of physiological parameters.

Although PET will continue to provide insights into biochemical and physiological processes *in vivo*, access to PET is limited due to the requirement of a cyclotron and high operation costs. Recent advances in quantitative SPECT and the widespread application of multidetector SPECT systems with improved sensitivity and dynamic imaging capabilities have made absolute physiological parameter estimation possible with the much more widely available SPECT. One of the major applications of dynamic SPECT is to quantify myocardial perfusion, which is important for the diagnosis and clinical management of patients with coronary artery disease where a perfusion defect after an intervention may indicate incomplete reperfusion or persistent coronary occlusion. Similar to dynamic PET, compartmental modeling is used in dynamic SPECT to quantify physiologic

parameters of interest. It has been demonstrated that myocardial blood flow can be studied by dynamic SPECT imaging of ^{99m}Tc -teboroxime [109, 110]. Another interesting application of dynamic cardiac SPECT is the measurement of perfusion and the distribution volume in the heart using ^{201}Tl [111, 112]. Other applications include brain imaging with ^{99m}Tc and ^{123}I flow agents and some neuroreceptor studies, such as [^{123}I]iomazenil for benzodiazepine receptors [115], [^{123}I]iododexetimide for cholinergic muscarinic neuroreceptors [114], and [^{123}I]iodo-A-85380 for central neuronal nicotinic acetylcholine receptors [115, 116], etc., both in human and nonhuman primates.

While PET and SPECT allow absolute measurements of radioactivity concentration in tissue and have the capability to relate the *in vivo* measurements to physiological functions, there are a number of limiting factors which can impact on their ability to produce reliable physiologic parameters. In particular, the small tissue volume, limitation on the total amount of activity that can be administered, physical artifacts, and low sensitivity (particularly for SPECT) result in measurements with high noise levels. The measurement noise (and mechanical rotation of the detectors in the case of SPECT) has imposed an upper limit on the fastest sampling rate which is >1 sec/sample for PET and >5 sec/sample for SPECT. Furthermore, the short half-life of the radiopharmaceuticals used for PET and SPECT and the need for the patient to remain still on the scanner bed usually limit duration of the PET and SPECT experiments to relatively short periods of time, which vary from minutes to hours rather than days. Slow physiologic processes are therefore difficult to be reliably estimated with both PET and SPECT. All these factors limit the number of parameters which can be reliably estimated from PET or SPECT measurement and complicated compartmental models used for analysis must be simplified. Huang *et al.* provide detailed guidelines on kinetic model development for PET applications and these principles are equally applicable to SPECT tracer studies.

One intrinsic limitation of PET imaging is the presence of physiological sites of tracer accumulation (secretion or excretion). It is sometimes difficult to differentiate real pathology from an unusual pattern of physiologic accumulation. In contrast, some organs of the body, for instance, the pancreas, uterus, and ovary, do not have significant physiological uptake. Spatially differentiation among these organs is sometimes difficult, particularly when they are closely related to one another. In this respect, structural imaging modalities such as X-ray CT can serve as an excellent anatomical roadmap for the functional PET images.

Integration of anatomical and functional images began to emerge during the late 1980s using software-based registration techniques [117, 118], although it was pointed out much earlier by Wagner [119] that this would be the trend for clinical imaging. These software-based registration techniques are applied to register images obtained with different imaging modalities or tracers (in case of registering emission tomographic images), of the same subject at different times. Registration techniques are successful for a rigid organ, such as the brain but they have been found to be problematic for other parts of the body.

Recognizing the advantages of combining the information provided by anatomic imaging and functional imaging, a prototype of an integrated scanner for PET and CT was designed in the early 1990s and a commercial hybrid scanner, named “PET/CT,” has been developed recently [120]. PET/CT is a completely new imaging technique that will likely revolutionize the conventional habit of acquiring and reading PET and CT data separately in the clinical environment. PET/CT imaging will enhance the combined utilization rate of what used to be PET-only or CT-only imaging as it provides, simultaneously, co-registered (fusion) images of both functional and anatomical information in a single acquisition. A potential advantage is the use of CT images for attenuation correction of PET emission data, thereby the need for acquiring a separate, lengthy transmission scan can be completely eliminated. Figure 2.13 shows a combined PET/CT

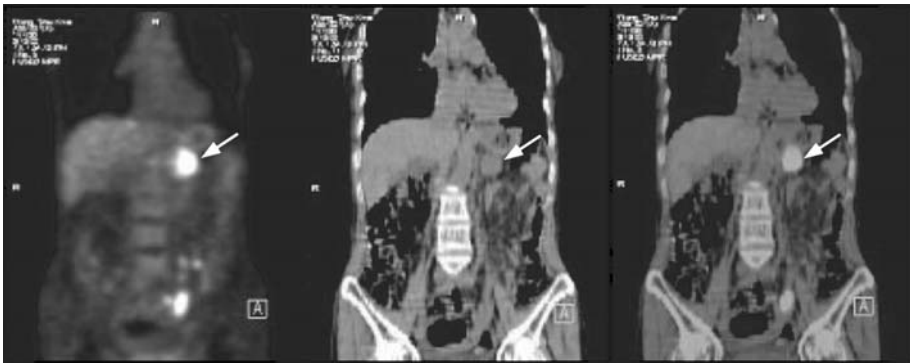


Figure 2.13: (Color slide) Combined PET/CT scan on a 72-year-old woman with a primary pancreatic cancer. From left to right: FDG-PET image, CT image, and fused PET and CT images. The scan shows abnormal FDG uptake in the pancreas (arrow). The fused image shows good alignment of two modalities and enables uptake to be localized to pancreas (arrow).

scan performed on a 72-year-old woman for investigation of unknown primary malignancy, and the combined PET/CT revealed a primary pancreatic cancer. Although the PET/CT imaging is still in its infancy and a number of technical problems remain to be solved, it is anticipated that the combination has the potential to solve many of the present diagnostic challenges associated with whole-body oncologic imaging and has an important role in surgical planning, radiation therapy, treatment monitoring, and diagnosis of disease.

Acknowledgment

This work was supported by the Hong Kong Polytechnic University under Grant G-YX13.

Questions

1. *Medical imaging modalities can be broadly classified into structural and functional. Briefly account for their meanings and give some examples for each case.*
2. *What are the major differences in the reconstructed images of emission computed tomography and X-ray CT?*
3. *PET and SPECT are collectively known as emission computed tomography. Briefly describe some of the major differences between them in terms of physics and instrumentation.*
4. *List the various events and their meanings in PET detection.*
5. *List some of the current applications of PET in clinical oncology.*
6. *Photon attenuation effect is common to both PET and SPECT imaging. Why is it so important to correct for this effect?*
7. *Briefly describe the main difference in correction of photon attenuation in PET and SPECT.*
8. *Attenuation correction in PET is usually performed using a transmission scan. What are the major drawbacks of this approach?*

9. *What is the basic assumption of filtered backprojection reconstruction? What is its major drawback?*
10. *What are the main factors that determine the suitability of a scintillator for PET imaging?*
11. *What is the main goal of tracer kinetic modeling?*
12. *Briefly describe some parametric imaging techniques for quantitative physiological parameter estimation in dynamic PET and SPECT.*

Bibliography

- [1] Cormack, A. M., Representation of a function by its line integral, with some radiological applications, II, *J. Appl. Phys.*, Vol. 35, pp. 2908–2913, 1964.
- [2] Hounsfield, G. N., A method and apparatus for examination of a body by radiation such as X or gamma radiation, The Patent Office, London, 1972, patent 1283915.
- [3] Hounsfield, G. N., Computerized transverse axial scanning (tomography). I: Description of system, *Br. J. Radiol.*, Vol. 46, pp. 1016–1022, 1973.
- [4] Radon, J., Über die bestimmung von funktionen durch ihre integralwäerte längs gewisser männigfaltigkeiten, *Bertichte Sæechsiche Akad. Wissenschaften (Leipzig), Math. Phys. Klass*, Vol. 69, pp. 262–277, 1917.
- [5] Warburg, O., *The Metabolism of Tumors*, Arnold and Constable, London, 1930.
- [6] Rutherford, E. and Soddy, F., The cause and nature of radioactivity, *Philos. Mag.*, Vol. 6th series, No. 4, pp. 370–396, 1902.
- [7] Dirac, P. A. M., A theory of electrons and protons, *Proc. R. Soc. A*, Vol. 126, pp. 360–365, 1930.
- [8] Anderson, C. D., Energies of cosmic-ray particles, *Phys. Rev.*, Vol. 40, pp. 405–421, 1932.
- [9] Joliot, F., Preuve experimentale de l'annihilation des electons postifs, *C. R. Acad. Sci.*, Vol. 197, pp. 1622–1625, 1933.
- [10] Thibaud, J., L'annihilation des positrons au contact de la matiere et la radiation qui en resulte, *C. R. Acad. Sci.*, Vol. 197, pp. 1629–1632, 1933.
- [11] Beringer, R. and Montgomery, C. G., The angular distribution of positron annihilation radiation, *Phys. Rev.*, Vol. 61, pp. 222–224, 1942.

- [12] Wrenn, F. R., Jr., Good, M. L., and Handler, P., Use of positron-emitting radioisotopes for localization of brain tumors, *Science*, Vol. 113, pp. 525–527, 1951.
- [13] Sweet, W. H., Use of nuclear disintegrations in the diagnosis and treatment of brain tumors, *N. Engl. J. Med.*, Vol. 245, pp. 875–878, 1951.
- [14] Brownell, G. L. and Sweet, W. H., Localization of brain tumors with positron emitters, *Nucleonics*, Vol. 11, pp. 40–45, 1953.
- [15] Kuhl, D. E. and Edwards, R. Q., Image separation radio-isotope scanning, *Radiology*, Vol. 80, pp. 653–661, 1963.
- [16] Kuhl, D. E. and Edwards, R. Q., Reorganizing data from transverse section scans using digital processing, *Radiology*, Vol. 91, pp. 975–983, 1968.
- [17] Todd-Pokropek, A. E., The formation and display of section scans, In: *Proc. Symp. Amer. Congress Radiol.*, pp. 545–556, 1972.
- [18] Burham, C. A. and Brownell, G. L., A multi-crystal positron camera, *IEEE Trans. Nucl. Sci.*, Vol. NS-19, pp. 201–205, 1972.
- [19] Anger, H. O., Multiple plane tomographic scanner, In: *Tomographic Imaging in Nuclear Medicine*, Freedman, G. S., ed., Society of Nuclear Medicine, New York, pp. 2–18, 1973.
- [20] Ter-Pogossian, M. M., Phelps, M. E., Hoffman, E. J., and Mullani, N. A., A positron-emission transaxial tomograph for nuclear medicine imaging (PETT), *Radiology*, Vol. 114, pp. 89–98, 1975.
- [21] Phelps, M. E., Hoffman, E. J., Mullani, N. A., and Ter-Pogossian, M. M., Application of annihilation coincidence detection to transaxial reconstruction tomography, *J. Nucl. Med.*, Vol. 16, pp. 210–214, 1975.
- [22] Hoffman, E. J., Phelps, M. E., Mullani, N. A., Higgins, C. S., and Ter-Pogossian, M. M., Design and performance characteristics of a whole body transaxial tomography, *J. Nucl. Med.*, Vol. 17, pp. 493–503, 1976.
- [23] Phelps, M. E., Hoffman, E. J., Coleman, R. E., Welch, M. J., Raichle, M. E., Weiss, E. S., Sober, B. E., and Ter-Pogossian, M. M., Tomographic

- images of blood pool and perfusion in brain and heart, *J. Nucl. Med.*, Vol. 17, pp. 603–612, 1976.
- [24] Phelps, M. E., Hoffman, E. J., Mullani, N. A., and Ter-Pogossian, M. M., Design considerations for a positron emission transaxial tomograph (PETT III), *IEEE Trans. Nucl. Sci.*, Vol. 23, pp. 516–522, 1976.
- [25] Phelps, M. E., Hoffman, E. J., Huang, S. C., and Kuhl, D. E., ECAT: A new computerized tomographic imaging system for positron emitting radiopharmaceuticals, *J. Nucl. Med.*, Vol. 19, pp. 635–647, 1978.
- [26] Bailey, D. L., Data acquisition and performance characterization in PET, In: *Positron Emission Tomography: Basic Science and Clinical Practice*, Valk, P. E., Bailey, D. L., Townsend, D. W., and Maisey, M. N., eds., Springer, London, pp. 69–90, 2003.
- [27] Cho, Z. K. and Farhiki, M. R., Bismuth germanate as a potential scintillator in positron cameras, *J. Nucl. Sci.*, Vol. 18, pp. 840–844, 1977.
- [28] Casey, M. E. and Nutt, R., A multicrystal two dimensional BGO detector system for positron emission tomography, *IEEE Trans. Nucl. Sci.*, Vol. 33, pp. 460–463, 1986.
- [29] Defrise, M. and Kinahan, P. E., Data acquisition and image reconstruction for 3D PET, In: *The Theory and Practice of 3D PET*, Bendriem, B. and Townsend, D. W., eds., Kluwer Academic, Dordrecht, pp. 1–53, 1998.
- [30] Kak, A. C. and Slaney, M., *Principles of Computerized Tomographic Imaging*, IEEE Press, New York, 1988.
- [31] Ramachandran, G. N. and Lakshminarayanan, A. V., Three-dimensional reconstruction from radiograph and electron micrographs: Application of convolutions instead of Fourier transform, *Proc. Natl. Acad. Sci. U.S.A.*, Vol. 67, pp. 2236–2240, 1971.
- [32] Budinger, T. F., Derenzo, S. E., Greenberg, W. L., Gullberg, G. T., and Huesman, R. H., Quantitative potentials of dynamic emission computed tomography, *J. Nucl. Med.*, Vol. 19, pp. 309–315, 1978.

- [33] Herman, G., *Image Reconstruction from Projections*, Academic Press, New York, 1980.
- [34] Llacer, J., Veklerov, E., Baxter, L. R., Grafton, S. T., Griffeth, L. K., Hawkins, R. A., Hoh, C. K., Mazziotta, J. C., Hoffman, E. J., and Metz, C. E., Results of a clinical operating characteristic study comparing filtered backprojection and maximum likelihood estimator images in FDG PET studies, *J. Nucl. Med.*, Vol. 34, pp. 1198–1203, 1993.
- [35] Wilson, D. W. and Tsui, B. M. W., Noise properties of filtered-backprojection and ML-EM reconstructed emission tomographic images, *IEEE Trans. Nucl. Sci.*, Vol. 40, pp. 1198–1203, 1993.
- [36] Hebert, T. and Leahy, R., A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors, *IEEE Trans. Med. Imaging*, Vol. 8, pp. 194–202, 1989.
- [37] Green, P. J., Bayesian reconstruction from emission tomography data using a modified EM algorithm, *IEEE Trans. Med. Imaging*, Vol. 9, pp. 84–93, 1990.
- [38] Shepp, L. A. and Vardi, Y., Maximum likelihood reconstruction for emission tomography, *IEEE Trans. Med. Imaging*, Vol. MI-1, pp. 113–122, 1982.
- [39] Lange, K. and Carson, R. E., EM reconstruction algorithms for emission and transmission tomography, *J. Comput. Assist. Tomogr.*, Vol. 8, pp. 306–316, 1984.
- [40] Hudson, H. M. and Larkin, R. S., Accelerated image reconstruction using ordered subsets of projection data, *IEEE Trans. Med. Imaging*, Vol. 13, pp. 601–609, 1994.
- [41] Meikle, S. R., Hutton, B. F., Bailey, D. L., Hooper, P. K., and Fulham, M. J., Accelerated EM reconstruction in total body PET: potential for improving tumour detectability, *Phys. Med. Biol.*, Vol. 39, pp. 1689–1704, 1994.
- [42] Cherry, S. R., Meikle, S. R., and Hoffman, E. J., Correction and characterization of scattered events in three-dimensional PET using scanners with retractable septa, *J. Nucl. Med.*, Vol. 34, pp. 671–678, 1996.

- [43] Thompson, C. J., The problem of scatter correction in positron volume imaging, *IEEE Trans. Med. Imaging*, Vol. 12, pp. 124–132, 1993.
- [44] Bailey, D. L. and Meikle, S. R., A convolution-subtraction scatter correction method for 3D PET, *Phys. Med. Biol.*, Vol. 39, pp. 411–424, 1994.
- [45] Levin, C. S., Dahlbom, M., and Hoffman, E. J., A Monte Carlo correction for the effect of Compton scattering in 3D PET brain imaging, *IEEE Trans. Nucl. Sci.*, Vol. 42, pp. 1181–1185, 1995.
- [46] Huang, S. C., Hoffman, E. J., Phelps, M. E., and Kuhl, D. E., Quantitation in positron emission computed tomography: 2. Effects of inaccurate attenuation correction, *J. Comput. Assist. Tomogr.*, Vol. 3, pp. 804–814, 1979.
- [47] Dahlbom, M. and Hoffman, E. J., Problems in signal-to-noise ratio for attenuation correction in high-resolution PET, *IEEE Trans. Nucl. Sci.*, Vol. 34, pp. 288–293, 1987.
- [48] Hooper, P. K., Meikle, S. R., Eberl, S., and Fulham, M. J., Validation of post injection transmission measurements for attenuation correction in neurologic FDG PET studies, *J. Nucl. Med.*, Vol. 37, pp. 128–136, 1996.
- [49] Huang, S. C., Carson, R. E., Phelps, M. E., Hoffman, E. J., Schelbert, H. R., and Kuhl, D. E., A boundary method for attenuation correction in positron computed tomography, *J. Nucl. Med.*, Vol. 22, pp. 627–637, 1981.
- [50] Xu, E. Z., Mullani, N. A., Gould, K. L., and Anderson, W. L., A segmented attenuation correction for PET, *J. Nucl. Med.*, Vol. 32, pp. 161–165, 1991.
- [51] Meikle, S. R., Dahlbom, M., and Cherry, S. R., Attenuation correction using count-limited transmission data in positron emission tomography, *J. Nucl. Med.*, Vol. 34, pp. 143–144, 1993.
- [52] Phelps, M. E., Hoffman, E. J., and Huang, S. C., Effect of positron range on spatial resolution, *J. Nucl. Med.*, Vol. 16, pp. 649–652, 1975.

- [53] Hoffman, E. J. and Phelps, M. E., Positron emission tomography: Principles and quantitation, In: Positron Emission Tomography and Autoradiography: Principles and Applications for the Brain and Heart, Phelps, M. E., Mazziotta, J. C., and Schelbert, H. R., eds., Raven Press, New York, pp. 237–286, 1986.
- [54] Derenzo, S. E., Budinger, T. F., and Vuletich, T., High resolution positron emission tomography using small bismuth germanate crystals and individual photosensors, *IEEE Trans. Nucl. Sci.*, Vol. NS-30, pp. 665–670, 1983.
- [55] Wong, W. H., Mullani, N. A., and Wardworth, G., Characteristics of small barium fluoride (BaF_2) scintillation for high intrinsic resolution time-of-flight positron emission tomography, *IEEE Trans. Nucl. Sci.*, Vol. 31, pp. 381–386, 1984.
- [56] Takagi, K. and Fukazawa, T., Cerium-activated Gd_2SiO_5 single crystal scintillator, *Appl. Phys. Lett.*, Vol. 42, pp. 43–45, 1983.
- [57] Melcher, C. L. and Schweitzer, J. S., Cerium-doped lutetium oxyorthosilicate: A fast, efficient, new scintillator, *IEEE Trans. Nucl. Sci.*, Vol. 39, pp. 502–505, 1992.
- [58] Brooks, R. A. and Di Chiro, G., Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging, *Phys. Med. Biol.*, Vol. 21, pp. 689–732, 1976.
- [59] Farquhar, T. H., Chatziioannou, A., Chinn, G., Dahlbom, M., and Hoffman, E. J., An investigation of filter choice for filtered back-projection reconstruction in PET, *IEEE Trans. Nucl. Sci.*, Vol. 45, pp. 1133–1137, 1998.
- [60] Levin, C. S. and Hoffman, E. J., Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution, *Phys. Med. Biol.*, Vol. 44, pp. 781–799, 1999.
- [61] Finkelstein, L. and Carson, E. R., Mathematical Modelling of Dynamic Biological Systems, 2nd ed., Research Studies Press Ltd, Letchworth, 1984.

- [62] Huang, S. C. and Phelps, M. E., Principles of tracer kinetic modeling in positron emission tomography and autoradiography, In: *Positron Emission Tomography and Autoradiography: Principles and Applications for the Brain and Heart*, Phelps, M. E., Mazziotta, J. C., and Schelbert, H. R., eds., Raven Press, New York, pp. 287–346, 1986.
- [63] Godfrey, K., *Compartmental Models And Their Application*, Academic Press, New York, 1983.
- [64] Bard, Y., *Nonlinear Parameter Estimation*, Academic Press, New York, 1974.
- [65] Huang, S. C., Barrio, J. R., Yu, D. C., Chen, B., Grafton, S., and Melega, W. P., Modeling approach for separating blood time-activity curves in positron emission tomographic studies, *Phys. Med. Biol.*, Vol. 36, pp. 749–761, 1991.
- [66] Iida, H., Jones, T., and Miura, S., Modeling approach to eliminate the need to separate arterial plasma in oxygen-15 inhalation positron emission tomography, *J. Nucl. Med.*, Vol. 34, pp. 1333–1340, 1993.
- [67] Phelps, M. E., Huang, S. C., Hoffman, E. J., Selin, C., Sokoloff, L., and Kuhl, D. E., Tomographic measurement of local cerebral glucose metabolic rate in humans with (F-18)2-fluoro-2-deoxy-D-glucose: Validation of method, *Ann. Neurol.*, Vol. 6, pp. 371–388, 1979.
- [68] Huang, S. C., Phelps, M. E., Hoffman, E. J., Sideris, K., Selin, C., and Kuhl, D. E., Noninvasive determination of local cerebral metabolic rate of glucose in man, *Am. J. Physiol.*, Vol. 238, pp. E69–E82, 1980.
- [69] Carson, R. E., Yan, Y., and Shrager, R., Absolute cerebral blood flow with ^{15}O -water and PET: Determination without a measured input function, In: *Quantification of Brain Function using PET*, Myers, R., Cunningham, V., Bailey, D., and Jones, T., eds., Academic Press, San Diego, pp. 185–190, 1996.
- [70] Di Bella, E. V. R., Clackdoyle, R., and Gullberg, G. T., Blind estimation of compartmental model parameters, *Phys. Med. Biol.*, Vol. 44, pp. 765–780, 1999.

- [71] Wong, K. P., Feng, D., Meikle, S. R., and Fulham, M. J., Simultaneous estimation of physiological parameters and the input function—*In vivo* PET data, *IEEE Trans. Inform. Technol. Biomed.*, Vol. 5, pp. 67–76, 2001.
- [72] Logan, J., Fowler, J. S., Volkow, N. D., Wang, G. J., Ding, Y. S., and Alexoff, D. L., Distribution volume ratios without blood sampling from graphical analysis of PET data, *J. Cereb. Blood Flow Metab.*, Vol. 16, pp. 834–840, 1996.
- [73] Lammertsma, A. A. and Hume, S. P., Simplified reference tissue model for PET receptor studies, *Neuroimage*, Vol. 4, pp. 153–158, 1996.
- [74] Gunn, R. N., Lammertsma, A. A., Hume, S. P., and Cunningham, V. J., Parametric imaging of ligand-receptor binding in PET using a simplified reference region model, *Neuroimage*, Vol. 6, pp. 279–287, 1997.
- [75] Patlak, C. S., Blasberg, R. G., and Fenstermacher, J., Graphical evaluation of blood-to-brain transfer constants from multiple-time uptake data, *J. Cereb. Blood Flow Metab.*, Vol. 3, pp. 1–7, 1983.
- [76] Patlak, C. S. and Blasberg, R. G., Graphical evaluation of blood-to-brain transfer constants from multiple-time uptake data: Generalizations, *J. Cereb. Blood Flow Metab.*, Vol. 5, pp. 584–590, 1985.
- [77] Logan, J., Fowler, J. S., Volkow, N. D., Wolf, A. P., Dewey, S. L., Schlyer, D. J., MacGregor, R. R., Hitzemann, R., Bendriem, B., Gatley, S. J., and Christman, D. R., Graphical analysis of reversible radioligand binding from time-activity measurements applied to [N-¹¹C-methyl]-(-)-cocaine PET studies in human subjects, *J. Cereb. Blood Flow Metab.*, Vol. 10, pp. 740–747, 1990.
- [78] Yokoi, T., Iida, H., Itoh, H., and Kanno, I., A new graphic plot analysis for cerebral blood flow and partition coefficient with iodine-123-iodoamphetamine and dynamic SPECT validation studies using oxygen-15-water and PET, *J. Nucl. Med.*, Vol. 34, No. 3, pp. 498–505, 1993.

- [79] Yokoi, T., Iida, H., and Kanno, I., A comparative study of the three fast algorithms to estimate cerebral blood flow and distribution volume using N-isopropyl- p -[^{123}I]iodoamphetamine and two SPECT scans, *Phys. Med. Biol.*, Vol. 40, pp. 1499–1515, 1995.
- [80] Blomqvist, G., On the construction of functional maps in positron emission tomography, *J. Cereb. Blood Flow Metab.*, Vol. 4, pp. 629–632, 1984.
- [81] Kety, S. S. and Schmidt, C. F., The nitrous oxide method for the quantitative determination of cerebral blood flow in man: Theory, procedure, and normal values, *J. Clin. Invest.*, Vol. 27, pp. 476–483, 1948.
- [82] Evans, A. C., A double integral form of the three-compartmental, four-rate-constant model for faster generation of parameter maps, *J. Cereb. Blood Flow Metab.*, Vol. 7, No. suppl., p. S453, 1987.
- [83] Feng, D., Wang, Z., and Huang, S. C., A study on statistically reliable and computationally efficient algorithms for the measurement of local cerebral blood flow with positron emission tomography, *IEEE Trans. Med. Imaging*, Vol. 12, pp. 182–188, 1993.
- [84] Feng, D. and Ho, D., Parametric imaging algorithm for multi-compartmental models dynamic studies with positron emission tomography, In: *Quantification of Brain Function: Tracer Kinetics and Image Analysis in Brain PET*, Uemura, K., Lassen, N. A., Jones, T., and Kanno, I., eds., Elsevier Science, Amsterdam, pp. 127–136, 1993.
- [85] Feng, D., Huang, S. C., Wang, Z., and Ho, D., An unbiased parametric imaging algorithm for non-uniformly sampled biomedical system parameter estimation, *IEEE Trans. Med. Imaging*, Vol. 15, No. 4, pp. 512–518, 1996.
- [86] Chen, K., Lawson, M., Reiman, E., Cooper, A., Feng, D., Huang, S. C., Bandy, D., Ho, D., Yun, L. S., and Palant, A., Generalized linear least squares method for fast generation of myocardial blood flow parametric images with N-13 ammonia PET, *IEEE Trans. Med. Imaging*, Vol. 17, pp. 236–243, 1998.

- [87] Cunningham, V. J. and Jones, T., Spectral analysis of dynamic PET studies, *J. Cereb. Blood Flow Metab.*, Vol. 13, pp. 15–23, 1993.
- [88] Lawson, C. L. and Hanson, R. J., *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [89] Meikle, S. R., Matthews, J. C., Cunningham, V. J., Bailey, D. L., Livieratos, L., Jones, T., and Price, P., Parametric image reconstruction using spectral analysis of PET projection data, *Phys. Med. Biol.*, Vol. 43, pp. 651–666, 1998.
- [90] Carson, E. R., Cobelli, C., and Finkelstein, L., *The Mathematical Modeling of Metabolic and Endocrine Systems: Model Formulation, Identification and Validation*, John Wiley and Sons, New York, 1983.
- [91] Fagarasan, J. T. and DiStefano, J. J., III, Hidden pools, hidden modes and visible repeated eigenvalues in compartmental models, *Math. Biosci.*, Vol. 82, pp. 87–113, 1986.
- [92] Huang, S. C., Carson, R. E., and Phelps, M. E., Measurement of local blood flow and distribution volume with short-lived isotopes: A general input technique, *J. Cereb. Blood Flow Metab.*, Vol. 2, pp. 99–108, 1982.
- [93] Alpert, N. M., Eriksson, L., Chang, J. Y., Bergstrom, M., Litton, J. E., Correia, J. A., Bohm, C., Ackerman, R. H., and Taveras, J. M., Strategy for the measurement of regional cerebral blood flow using short-lived tracers and emission tomography, *J. Cereb. Blood Flow Metab.*, Vol. 4, pp. 28–34, 1984.
- [94] Carson, R. E., Huang, S. C., and Green, M. V., Weighted integration method for local cerebral blood flow measurement with positron emission tomography, *J. Cereb. Blood Flow Metab.*, Vol. 6, pp. 245–258, 1986.
- [95] Yokoi, T., Kanno, I., Iida, H., Miura, S., and Uemura, K., A new approach of weighted integration technique based on accumulated images using dynamic PET and H_2^{15}O , *J. Cereb. Blood Flow Metab.*, Vol. 11, pp. 492–501, 1991.

- [96] Tsui, E. and Budinger, T. F., Transverse section imaging of mean clearance times, *Phys. Med. Biol.*, Vol. 23, pp. 644–653, 1978.
- [97] Phelps, M. E., Mazziotta, J. C., and Huang, S. C., Study of cerebral function with positron computed tomography, *J. Cereb. Blood Flow Metab.*, Vol. 2, pp. 113–162, 1982.
- [98] Mazziotta, J. C. and Phelps, M. E., Positron emission tomography studies of the brain, In: *Positron Emission Tomography and Autoradiography: Principles and Applications for the Brain and Heart*, Phelps, M. E., Mazziotta, J. C., and Schelbert, H. R., eds., Raven Press, New York, pp. 493–579, 1986.
- [99] Grafton, S. T. and Mazziotta, J. C., Cerebral pathophysiology evaluated with positron emission tomography, In: *Diseases of the Nervous System: Clinical Neurobiology*, Asbury, A. K., Mckhann, G. M., and McDonald, W. I., eds., Saunders, Philadelphia, pp. 1573–1588, 1992.
- [100] Frey, K. A., PET studies of neurochemical systems, In: *Positron Emission Tomography: Basic Science and Clinical Practice*, Valk, P. E., Bailey, D. L., Townsend, D. W., and Maisey, M. N., eds., Springer, London, pp. 309–327, 2003.
- [101] Bar-Shalom, R., Valdivia, A. Y., and Blaurox, M. D., PET imaging in oncology, *Semin. Nucl. Med.*, Vol. 30, pp. 150–185, 2000.
- [102] Rhodes, C. G., Wise, R. J., Gibbs, J. M., Frackowiak, R. J., Hatazawa, J., Palmer, A. J., Thomas, D. G. T., and Jones, T., *In vivo* disturbance of the oxidative metabolism of glucose in human cerebral gliomas, *Ann. Neurol.*, Vol. 14, pp. 614–626, 1983.
- [103] Di Chiro, G., Positron emission tomography using [^{18}F]fluorodeoxyglucose in brain tumors: a powerful diagnostic and prognostic tool, *Invest. Radiol.*, Vol. 22, pp. 360–371, 1987.
- [104] Doyle, W. K., Budinger, T. F., Valk, P. E., Levin, V. A., and Gutin, P. H., Differentiation of cerebral radiation necrosis from tumor recurrence by [^{18}F]FDG and ^{82}Rb positron emission tomography, *J. Comput. Assist. Tomogr.*, Vol. 11, pp. 563–570, 1987.

- [105] Strauss, L. G. and Conti, P. S., The applications of PET in clinical oncology, *J. Nucl. Med.*, Vol. 32, pp. 623–648, 1991.
- [106] Glasby, J. A., Hawkins, R. A., Hoh, C. K., and Phelps, M. E., Use of positron emission tomography in oncology, *Oncology*, Vol. 7, pp. 41–46, 1993.
- [107] Coleman, R. E., Clinical PET in oncology, *Clin. Pos. Imaging*, Vol. 1, pp. 15–30, 1998.
- [108] Anger, H. O., Scintillation camera, *Rev. Sci. Instrum.*, Vol. 29, pp. 27–33, 1958.
- [109] Smith, A. M., Gullberg, G. T., Christian, P. E., and Datz, F. L., Kinetic modeling of teboroxime using dynamic SPECT imaging of a canine model, *J. Nucl. Med.*, Vol. 35, pp. 484–495, 1994.
- [110] Smith, A. M., Gullberg, G. T., and Christian, P. E., Experimental verification of technetium 99m-labeled teboroxime kinetic parameters in the myocardium with dynamic single-photon emission computed tomography: Reproducibility, correlation to flow, and susceptibility to extravascular contamination, *J. Nucl. Cardiol.*, Vol. 3, pp. 130–142, 1996.
- [111] Iida, H. and Eberl, S., Quantitative assessment of regional myocardial blood flow with thallium-201 and SPECT, *J. Nucl. Cardiol.*, Vol. 5, pp. 313–331, 1998.
- [112] Eberl, S., Quantitative Physiological Parameter Estimation from Dynamic Single Photon Emission Computed Tomography (SPECT), Ph.D. Thesis, University of New South Wales, Australia, 2000.
- [113] Laruelle, M., Baldwin, R. M., Rattner, Z., Al-Tikriti, M. S., Zea-Ponce, Y., Zoghbi, S. S., Charney, D. S., Price, J. C., Frost, J. J., Hoffer, P. B., and Innis, R. B., SPECT quantification of [^{123}I]iomazenil binding to benzodiazepine receptors in nonhuman primates. I: Kinetic modeling of single bolus experiments, *J. Cereb. Blood Flow Metab.*, Vol. 14, pp. 439–452, 1994.

- [114] Boundy, K. L., Rowe, C. C., Black, A. B., Kitchener, M. I., Barnden, L. R., Sebben, R., Kassiou, M., Katsifis, A., and Lambrecht, R. M., Localization of temporal lobe epileptic foci with iodine-123 iododexetimide cholinergic neuroreceptor single-photon emission computed tomography, *Neurology*, Vol. 47, pp. 1015–1020, 1996.
- [115] Chefer, S. I., Horti, A. G., Lee, K. S., Koren, A. O., Jones, D. W., Gorey, J. G., Links, J. M., Mukhin, A. G., Weinberger, D. R., and London, E. D., *In vivo* imaging of brain nicotinic acetylcholine receptors with 5-[¹²³I]iodo-A-85380 using single photon emission computed tomography, *Life Sci.*, Vol. 63, pp. PL355–PL360, 1998.
- [116] Kassiou, M., Eberl, S., Meikle, S. R., Birrell, A., Constable, C., Fulham, M. J., Wong, D. F., and Musachio, J. L., *In vivo* imaging of nicotinic receptor upregulation following chronic (-)-nicotine treatment in baboon using SPECT, *Nucl. Med. Biol.*, Vol. 28, pp. 165–175, 2001.
- [117] Pelizzari, C. A., Chen, G. T. Y., Spelbring, D. R., Weichselbaum, R. R., and Chen, C. T., Accurate three-dimensional registration of CT, PET and/or MR images of the brain, *J. Comput. Assist. Tomogr.*, Vol. 13, pp. 20–26, 1989.
- [118] Woods, R. P., Mazziotta, J. C., and Cherry, S. R., MRI-PET registration with automated algorithm, *J. Comput. Assist. Tomogr.*, Vol. 17, pp. 536–546, 1993.
- [119] Wagner, H. N., Jr., Images of the future, *J. Nucl. Med.*, Vol. 19, pp. 599–605, 1978.
- [120] Beyer, T., Townsend, D. W., Brun, T., Kinahan, P. E., Charron, M., Roddy, R., Jerin, J., Young, J., Byars, L., and Nutt, R., A combined PET/CT scanner for clinical oncology, *J. Nucl. Med.*, Vol. 41, pp. 1369–1379, 2000.

Chapter 3

Advances in Magnetic Resonance Angiography and Physical Principles

Rakesh Sharma¹ and Avdhesh Sharma²

3.1 Introduction

In this chapter, we will discuss the physical principles of magnetic resonance angiography (MRA). MRA may at first appear very complicated, but we shall try to present the major concepts in the simplest form. The first part concentrates on physical principles of flow magnetization and flow characteristics in human vascular system. The later part is devoted to various magnetic resonance angiography techniques from the MRA physics as well as angiography technique refinement points of view.

MRA is a technique for obtaining information on blood motion mainly in the cardiovascular and cerebrovascular systems. Let us consider how motion or flow in the vessels generates the angiographic effect for creating magnetic resonance (MR) images.

3.1.1 Principles of Magnetization and Flow

The vascular system experiences motion of blood due to continuous flow of blood inside. Precession frequency and gradient field vectors are related. These vectors are represented as spin isochromats. The behavior of the moving spin

¹Department of Medicine, Columbia University, New York, NY 10032, USA

²Electrical Engineering Department, Indian Institute of Technology, New Delhi 10016, India

isochromats can be explained as follows:

$$\delta\Phi/\delta t = \omega_0 = \gamma(B_0 + xG_x + yG_y + zG_z) \quad (3.1)$$

where γ is gyromagnetic ratio, B_0 is magnetic field strength, x, y, z are position vectors of a spin isochromat, and G is the applied gradient field vector. This vector has components viz. G_x, G_y , and G_z along the x, y , and z directions, respectively. Inside the vessels, slight variations in magnetic field make the spin isochromats precess at different speeds. The spin isochromat precessing in different directions can be represented as different points on a precession circle. Simultaneously, they lose phase coherence in this process that results in loss of MR signal. However, two methods are commonly used to recover MR signal loss viz. refocusing 180° RF pulse and gradient recalled echo (GRE). Spin isochromat magnetization is inverted by applying excitation time less than TE i.e. $T = TE/2$. Refocusing 180° RF pulse in spin echo (SE) sequence sent after time $T = TE/2$ inverts isochromat magnetization. The refocusing 180° RF pulse creates a head start. So, it refocuses the slow moving spins to reach the x axis as shown in Fig. 3.1. This whole process is known as dephasing or defocusing.

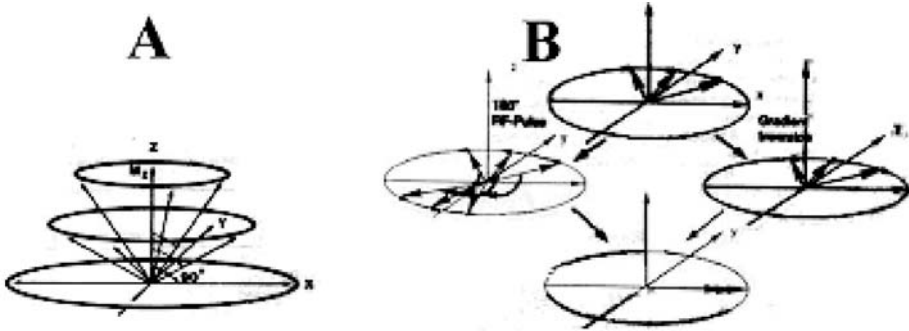


Figure 3.1: RF pulse is shown to flip the magnetization out of its orientation along the z -axis by a variable flip angle θ , magnetization vector starts to precess, describing a isochromat circle in the x,y plane (Figure A) for spin-echo imaging at flip angle 90° . After 90° pulse, the isochromats precess with different Larmor frequencies due to experience of different magnetic fields (shown with arrows). A typical spin-echo pulse is shown with RF pulse flipping magnetization 180° and back to create an echo (middle row). In GRE sequence, inverted readout gradient is used to invert precession and result refocusing pulse.

Alternatively, the gradient field-recalled echo or gradient inversion method inverts the precession direction of spin isochromats. Interestingly, slice-selection gradient is not needed after initial phase in this process. So, re-focusing is achieved by using a negative read-out gradient for the first echo, a positive one for the second echo, and so on. In both methods, all the precessing isochromats point along the x direction after time TE. It results in first spin echo generation.

3.1.1.1 Spin Isochromats in Motion

Let us consider the case of time-dependent position $x(t)$ of a spin isochromat in motion. The position may be represented as Taylor series expansion in the x direction:

$$x(t) = S + Vt + A_x t^2 + \text{higher order terms}$$

where S is initial position of spin isochromat, V is velocity, and A is acceleration in time t .

For simplicity, assume a spin isochromat moves along the x axis (y axis and z axis assumed zero) and read-out occurs along the x axis. In that case, according to Eq. (3.1), G_x gradient will have an effect on spin isochromat to generate precession phase of moving spin isochromat relative to stationary spin isochromat (see Fig. 3.2). This precession phase can be represented as

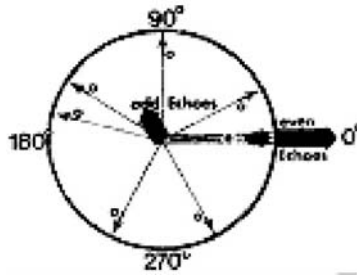


Figure 3.2: Precessing isochromats are shown in motion to result nonzero phase angle at odd echoes (arrows with label “o”). The isochromat magnetization vectors within a voxel add up to a small resultant vector (short thick arrow) if the isochromats within the voxel have different velocities. On even echoes, all isochromat magnetization vectors point in the 0° direction (along the x -axis) independent of velocity (arrows labeled “e”).

follows:

$$\delta\Phi = \gamma[(G_x S_x + G_x V_x t + G_x A_x t^2/2) + (\text{higher order terms})]\delta t \quad (3.2)$$

The phases of precession and motion under the influence of gradient G_x may be explained to generate spin echo and even echo refocusing phenomena. Gradient field G_x is turned on. Precession phase of moving spin isochromat is integrated over different time intervals on a precession circle. It will show stationary spin isochromat pointing along the x direction at the first echo. Moving spin isochromats will point in any direction in the xy plane. Let us consider the basis of ‘even echo refocusing phenomenon’ in these spin isochromats. The phase angle Φ in these spin isochromats is proportional to the velocity and gradient field strength G_x . However, the second and other even echoes ($n = 2, 4, 6, \dots$) have phase angle zero. The phase angles of even echoes are independent of velocity in the case of constant-velocity motion and symmetrical echoes.

These concepts explain the behavior of phase and motion. Variations in phase and motion of flowing blood inside vessels appear with variable spin-phase appearance of flowing blood. Similarly for accelerated motion, the phase angle is proportional to acceleration. In this case, even echo refocusing does not happen. Interestingly, velocity-induced phase changes are proportional to the time t_p . t_p is defined as the time during which the gradient field G_x is switched on, and is a function of the echo time ($T = TE/2$). Acceleration-induced phase changes are functions of the echo time TE and t_p .

3.1.1.2 Flow Information in Spin Isochromats

In spin echo pulse sequence, gradient vectors are represented in the x , y , and z directions as G_x , G_y , and G_z gradient fields. In an earlier section, motion in the gradient field G_x was explained. Let us consider the case of motion along the other gradient fields G_y and G_z . Similar spin isochromat effects and relationship may be explained. These flow effects are stronger along the slice-selected gradient. These flow effects are negligible along phase encoding gradient. For read gradient, area under G_z , before and after 180° refocusing pulse are equal. On the contrary, for GRE sequence, read gradient is opposite. This read gradient is equal to 1, just prior to read-out gradient. So, the refocusing effect is generated. For it, during read-out, gradient is turned on for twice as long as that at the beginning of the pulse sequence.

Motion inside the vessels produces predictable changes in the precession phases of moving spin isochromats relative to stationary spin isochromats. Inside the vessel, for each voxel, phase angles can be determined based on the projections of magnetization M_{xy} along the x and y axes. Precessing spins in the voxel exhibit different phase angles. These phase angles in the voxel generate real and imaginary images. In general, the images may be represented as modulus or amplitude images in different voxels. These phase image amplitudes correspond to the length of magnetization vector M_{xy} . So, these images represent voxel-by-voxel velocity for applied gradient fields. In other words, motion can be identified as areas on phase images where phase is nonzero. However, in the voxel, spin phases and image generation suffer from magnetic field inhomogeneity artifacts. These inhomogeneity artifacts affect the entire magnetic field. An abrupt change in phase along a smaller intravascular area exhibits phase variation due to intravascular signal. This abrupt phase variation along a smaller area is used for generating image flow abnormality.

3.1.1.3 Laminar, Turbulent and Pulsatile Flow in Human Vascular System

Blood flows in a human body in a well-defined physiological closed circulatory system. The flow is regulated by the heart and exhibits different flow properties known as flow patterns. Blood flow patterns are different at different locations in the intravascular system. MR signal intensities from such intravascular locations in cardiovascular and cerebrovascular systems appear dependent on hemodynamic properties of the cardiovascular or cerebrovascular system. The other important property of vascular system is flow velocity. In general, blood velocity inside a vessel is the largest at the center and zero around the walls. The flow velocity and vessel diameter plots are known as flow profile. This concept is significant in the analysis of MR signal loss.

Three types of flow velocities are representative viz. laminar, turbulent, and pulsatile flow (see Fig. 3.3). Laminar flow is defined as a flow pattern in which adjacent layers of fluid glide past each other without mixing different flowing blood layers. This type of flow may be called parabolic flow. The velocity varies quadratically with the distance from the center of the vessel. At the center, the

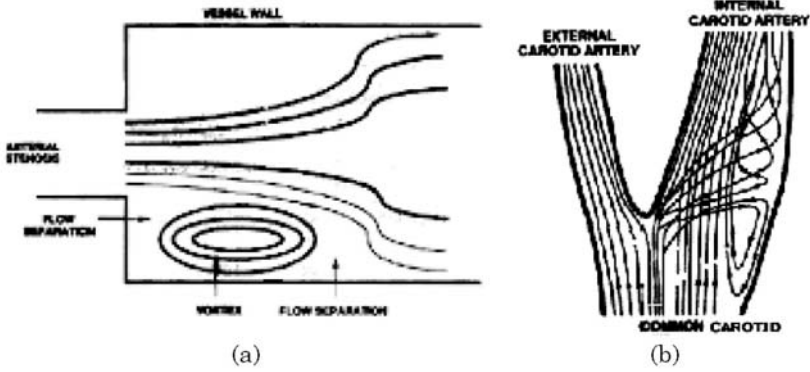


Figure 3.3: The central streamline flow separating from the vessel wall to produce a vertex or flow eddy stagnant blood to cause hemodynamic condition in VMRI (on left). The flow pattern at carotid bifurcation shows countercurrent flow and flow separation phenomena within the carotid bifurcation.

flow is maximum. Turbulent flow is defined as a rectangular flow pattern. The flow velocity is high in the whole region and vortices do appear. Adjacent layers are mixed. The flow is known as 'plug flow' otherwise velocity as a function of spin position is defined by Laminar flow as following:

$$V(\vec{r}) = V_{max}[1 - (r/a)^2] \quad (3.3)$$

where a is radius of vessel as cylinder. So, the plug flow for every phase-encoding step may be defined at constant flow as:

$$\rho(x, y) = e^{i\gamma G_0 v(x, y)\tau/2} \cdot \rho(x, y)^\tau \quad (3.4)$$

where G_0 is bipolar pulse strength and τ is length of time and phase is $\gamma G_0 v\tau^2$ with flow along x . In case of velocity as function of spin position for the flow along x when vessel is in-plane the laminar flow may be defined as:

$$\rho(x, y) = e^{i\gamma G_0 v(x, y)\tau/2} \cdot \rho(x, y) \quad (3.5)$$

These flow characteristics are interrelated by Reynolds number, Re , as:

$$Re = 2R_0 v_{av} \rho / \eta \quad (3.6)$$

where ρ is density and η is viscosity of fluid.

For $Re > 2000$, the flow is defined as turbulent flow. For $Re > 7000$, the flow is defined as pulsatile flow as observed in arteries for a transition state between

laminar and turbulent flow. First, laminar flow facilitates the acceleration of the blood flow to reach peak flow velocity. Later, the transition from laminar flow to turbulent flow appears as early phase in the deceleration phase soon after the peak velocity. In such situations, the transition flow depends upon the curvature and radius of a vessel. This flow generates forces parallel to the vessel wall termed as 'shear force'. For example, shear forces are common at the points of atherosclerotic plaque in the arterial wall. The shear force can be represented as: $s = \eta \delta v / \delta r$ where η is coefficient of viscosity and $\delta v / \delta r$ is the radial variation of velocity in the vessel. In the vessel, the shear force is greater close to the vessel wall. The reason for this is that the radial spatial variation in velocity is largest there.

In humans, laminar flow is common in veins and capillaries. This flow varies due to respiratory motion and arterial contractions. The flow velocity in the veins varies on the order of 10–20 cm/sec. In the arteries, blood flow is pulsatile with Reynolds number > 7000 . In blood vessels, turbulence is rarely observed. However, turbulence may be seen in large arteries and systolic motion in the heart. Typical flow velocities in large arteries vary from zero in the end-diastolic phase of cardiac cycle to 50–100 cm/sec in the mid-systole. Larger spatial variations in flow velocity are also observed at the vessel walls near vascular bifurcation sites at which atherosclerotic plaque appears. In arteries, blood flow in cardiac chambers is pulsatile because cardiac chambers are large open spaces. In these chambers, R_0 is large and inflow and outflow of blood result in vortex formation and also in large spatial velocity variations. This flow characteristic is known as 'cine ventriculography'. Vortex formation is related to rapid inflow and outflow of blood in the cardiac chambers. In the diastolic phase, little flow and small volume changes are observed as short-lived phase. This short-lived phase of cardiac cycle depends on the heart rate. These are common in patients with low heart rates. In these patients blood is approximately stagnant during late diastole, while systolic events are less affected by heart rate. At heartbeats above 70 beats per minute, patients show appearance of vortices and spatial variation in flow velocity in cardiac chamber. These spatial variations affect systole and diastole. On the other hand, microvascular circulation occurs at flow velocities 0.5–1.0 cm/sec and is pulsatile in the arterioles up to the precapillary sphincter. It is continuous in the capillaries and venules distal to it. Vessel walls experience high shear forces. Microcirculation vessels do form a network of vessels with changing orientation inside the vessel.

3.1.1.4 Factors Influencing the Appearance of Flowing Blood on MR Images

The magnetization M_{xy} , after application of 90° nutation pulse in an SE experiment, can be explained as:

$$M_{xy} = F\{v, a, \dots\}H\{1 - \exp(t/T1)\}(\exp(-t/T2)) \quad (3.7)$$

where the expressions in parentheses characterize longitudinal (T1) and transverse (T2) dependent magnetizations and are known as relaxation times. H is proton density. The factor $F(v, a, \dots)$ is flow factor which depends on spin isochromats. Velocity (v), acceleration (a), slice transition, spin phase phenomena, and high-order motion terms are the main variables.

3.1.2 Flow Physical Principles

Let us discuss the basic flow patterns in blood vessels and related MRA signal magnitudes. Mathematical models of laminar and plug flow for signals from spin echo pulses are well established. Flow causes a physical displacement of spins between successive excitations. This time-of-flight effect leads to different series of RF pulses producing different echo amplitudes for a number of different spin populations. The fraction volume of each population can be expressed as a function of the interpulse interval length during which each population enters the slice. So, the total signal is the sum of the echo amplitudes from each spin population. Different pulse sequences generate image signal intensity (SI) as a function of velocity, TR, TI, TE and slice thickness.

Fractional volume segments. The cylindrical volume of the vessel cut by a slice is equal to fraction of flowing blood volume of imaging (VOI).

MRA image signal. The evolution of magnetization for each spin population can be described by Bloch equations. For simplicity, magnetization may be expressed for transforming to a reference rotating frame at Larmor frequency ($-\gamma H_0$) according to Bloch equations as follows:

$$M_x(t) = M_{0x} \exp(-t/T2) \quad (3.8)$$

$$M_y(t) = M_{0y} \exp(-t/T2) \quad (3.9)$$

$$M_z(t) = M_0[1 - \exp^{(t-t/T1)}] + M_{0z} \exp^{(-t/T1)} \quad (3.10)$$

where M_{0x} , M_{0y} , and M_{0z} are the values of M_x , M_y , and M_z at $t = 0$, and M_0 is the steady-state magnetization in the z direction. This condition is good for spins rotating in the transverse plane aligning through the x axis at $M_{0y} = 0$.

For fundamental understanding for the signal generation, we describe different signal intensities of main pulse sequences as follows:

$$90^\circ \text{ selective} - TE/2 - 180^\circ: S = M_0 \exp(-TE/T_2)$$

$$180^\circ - (TR - TE/2) - 90^\circ - TE/2 - 180^\circ:$$

$$S = M_0 \{1 - 2\exp[-(TR - TE/2)/T_1]\} \exp(-TE/T_2)$$

$$90^\circ - TE/2 - 180^\circ - (TR - TE/2) - 90^\circ - TE/2 - 180^\circ:$$

$$S = M_0 \{1 - 2\exp[-(TR - TE/2)/T_1] + \exp(-TR/T_1)\} \exp(-TE/2)$$

The complete expression for the echo signal has four separate velocity boundary conditions:

Plug flow signal will be:

$$SV(TR - TE/2)/D + SVTE/(2D) + S[1 - V(TR + TE/2)D]$$

$$\text{if } 0 < V < D/(TR + TE/2)$$

$$SV(TR - TE/2)/D + S(1 - VTR/D) \quad \text{if } D/TR + TE/2 < V < (TR$$

$$S[1 - VTE/(2D)] \quad \text{if } D/TR < V < D/(TE/2)$$

$$0 \quad \text{if } V > D/(TE/2)$$

Laminar flow signal will be:

$$S_a[V_m(TR - TE/2)/(2D)] + S_b V_m TE/(4D) + S_c[1 - V_m(TR + TE/2)/(2D)]$$

$$\text{if } 0 < V_m < D/(TR + TE/2)$$

$$S_a[V_m(TR - TE/2)/(2D)] + S_b\{1 - D/2V_m(TR + TE/2)\} - V_m TR/(2D)\}$$

$$+ S_c D/[2V_m(TR + TE/2)] \quad \text{if } D/(TR + TE/2) < V_m < D/TR$$

$$S_a[1 - D/(2V_m TR) - V_m TE/4D] + S_b\{D/(2V_m TR) - D/[2V_m(TR + TE/2)]\}$$

$$+ S_c D/[2V_m(TR + TE/2)] \quad \text{if } D/(TR) < V_m < D/(TE/2)$$

$$S_a[D/V_m TE] - D/2V_m(TR) + S_b\{D/(2V_m TR) - D/[2V_m(TR + TE/2)]\}$$

$$+ S_c D/[2V_m(TR + TE/2)] \quad \text{if } V_m > D/(TE/2)$$

90° selective-TI- 90° selective- $TE/2$ - 180° selective pulse:

a. 90° - $TE/2$ - 180° : $S_a = M_0 \exp(TE/T_2)$

b. 90° -TI- 90° - $TE/2$ - 180° : $S_b = M_0[1 - \exp(-TI/T_1) \exp(-TE/T_2)]$

3.1.2.1 Slice-Transition Phenomenon

It is known as the time-of-flight effect. The time-of-flight effect occurs whenever a vessel takes course in part perpendicular to the imaging plane. During this course, substantial amount of blood volume is replaced within the imaged slice for several hundred milliseconds. Different pulse sequences capture these events of blood volume turnover in selected slices and voxels of the circulatory system. For example, images are captured in several hundred milliseconds by spin echo (SE) pulse sequence and a few tens of milliseconds in GRE imaging. Normally, the flow direction is selected perpendicular to the imaged slice axis in the xy plane. The flow is chosen along the z direction. Time-of-flight depends upon the slice thickness, position of slices in stack region of interest, multislice acquisition order, flow velocity, and flow direction of spin isochromats. Assume a volume element or voxel that moves with constant velocity V_0 . If it moves a distance z_0 during time t , we can write

$$V_0 = z_0/t.$$

Two types of slice-transition phenomenon can occur. In the first case, blood moves so fast that it leaves the slice between the 90° nutation pulse and the 180° refocusing pulse in an SE sequence. It results in signal loss and has been termed as high-velocity signal loss. In the second case, the flow velocity is small enough that only a small fraction of blood leaves the slice between the 90° and 180° pulses in the SE sequence, but a part of the blood is replaced by fully magnetized blood from outside of the imaged volume during the repetition time (TR). It leads to increased intravascular signal and has been termed flow-related enhancement.

3.1.2.1.1 High-Velocity Loss. Spin isochromats leave the imaged slice in an SE sequence before the rephasing pulse for a given echo is applied. These spin isochromats will not contribute to this echo. These spin isochromats continue with their dephasing process or rephasing. This dephasing process or rephasing will not be experienced by spin isochromats. In general, the rephasing pulse occurs at time T_e , which is equal to $TE/2$ for the first echo, or $(n - 1/2)TE$ if the n^{th} echo in a multiecho sequence with regular echo spacing is used. Here T_e is equal to $T_{E1} + (T_{E2} - T_{E1})/2$ if a sequence with two asymmetrical echoes (at times T_{E1} and T_{E2}) is used. The maximum velocity (V_z) depends on slice

thickness (s). The maximum velocity of blood (V_z) above which blood leaves the slice will be more than s/T_e . For lower velocities, only a fraction of blood leaves the voxel up to the rephrasing pulse time T_e . The fraction of blood (g) will be dependent on maximum velocity and slice thickness as:

$$g = V_z/s.$$

Thus, the recorded magnetization decreases linearly with the velocity down to zero. The blood flow may be observed to be very slow. As a result, intravascular signal is seen on first echo, but no longer on second echo. It indicates blood has moved out of the slice during the time interval of $TE/2$ and $3TE/2$. If we know the slice thickness, we can calculate blood flow velocity. The signal loss on the first echo at the rim of the vessel is commonly observed due to spin-phase effect in SE sequences. On the contrary, in GRE sequences, slice-selective rephrasing pulse is not applied. So, slice-transition effects are not observed and do not produce any intravascular signal loss. At low blood velocities, only few out of all spin isochromats leave the slice during the time T_e . Hence, during the repetition time $TR \gg T_e$ applied, some or all spin isochromats can be replaced by still fully magnetized spin isochromats. These spin isochromats represent from outside the imaged volume. In this region, these spin isochromats have not undergone repetitive 90° pulse in SE or alpha pulse in GRE sequences. This gives rise to 'entry slice phenomenon'. We shall discuss entry slice phenomenon in the following section.

3.1.2.1.2 Entry Slice Phenomenon. If $T1 \gg TR$ for blood, spin isochromats cannot fully recover their magnetization along the z axis. If spin isochromats in blood move perpendicular to a stack of slices with velocity v , they are subjected to 90° in SE imaging and to an alpha pulse in GRE pulse sequence. Due to flow, they are partly replaced by spin isochromats from outside imaging volume. This outside volume has not been subject to such pulse. So, net magnetization M available for the next 90° pulse or alpha pulse is the sum of the magnetization of the remaining spin isochromats. It leads to increased intravascular signal intensity. The temporal dependence of M over time will represent an increase in M in such a way that the first linear increase will be due to incomplete magnetization recovery. The later phase in the increase in M as a plateau will represent flow-related enhancement. The magnitude of this effect is again dependent on the flow velocity (v), the slice thickness, and TR . The fraction of blood g in the

voxel that is replaced is given by

$$g_1 = vT/s \quad (3.11)$$

while other one remaining is given by $1 - g_2$. The sum of the two magnetization components in the vessel is thus

$$M(TR) = M\{(1 - g_1)[1 - \exp(-TR/T_1)] + g_2\}\{(1 - g_1)[1 - \exp(-TR/T_1)]\} \quad (3.12)$$

It represents previous voxel in slice and g_2 represents fresh voxel. The flow-related enhancement is prominent when a significant fraction of blood in a slice is replaced during the time TR. With optimized values for slice thickness, s , and repetition time, TR, in a sequence, such flow velocities, v , are of the order of a few centimeters per second. At TR = 500 msec, $s = 0.5$ cm, the flow velocity will measure 1 cm/sec. At higher velocities the combination of higher velocity signal loss and flow-related enhancement tends to reduce the intravascular signal intensity.

3.1.2.1.3 Flow-Related Enhancement. During multislice acquisition, it can be operative in several slices of a stack. The spin isochromats moving at the center of a vessel are generally faster than those close to the vessel wall. Therefore, centrally located spin isochromats move deeper onto the stack during the repetition time (TR) than peripherally located ones. Suppose the planes of entry into different slices are separated by a distance q ($q \gg$ slice thickness), then fully magnetized spin isochromats moving with a velocity v will enter the j th slice after a time, $t = j \cdot q/v$ after entering the first slice. Spin isochromats that move a distance $j \cdot q$ after the j th slice and before the $(j + 1)$ th slice are irradiated with RF pulses. These contribute their full magnetization to the signal measured in the $(j + 1)$ th slice from outside the stack without being disturbed by RF irradiation. Here slices are acquired in the sequence parallel to flow. The deeper slice in the stack indicates that faster blood flow enhances the signal in that slice. The fast flow causes high-velocity signal loss in SE images. Thus flow-related enhancement cannot be observed in all slices of a stack. Suppose $q = 10$ mm, TE = 30 msec, and adjacent slices are excited 100 msec apart, the total signal loss will occur for velocities of approximately 65 cm/sec and above according to $V = K \cdot s/TE$ where K and s are slice thickness. Blood moving through the first slice between the refocusing pulse at T = 15 msec and into the

next slice before it is excited at $t = 100$ msec will contribute full magnetization to the second slice. Such spin isochromats must move with a velocity V of at least $1 \text{ cm}/(100 - 15 \text{ msec}) = 12 \text{ cm/sec}$, which is lower than 65 cm/sec (velocity needed for total signal loss). Such spin isochromats contribute to signal enhancement in the second slice. In fact, isochromats moving at approximately 24 cm/sec and 36 cm/sec can reach the third and fourth slices, respectively, to contribute to the signal enhancement. However, spin isochromats getting to slices farther into the stack move progressively faster and these isochromats approach the velocities at which high-velocity signal loss occurs. On this basis, it is possible to detect a bright spot of signal with decreased diameter at multiple slices into a stack. On the contrary, if the order of acquisition is reversed in another way to minimize crosstalk between slices, signal enhancement may occur in slices even deeper into the stack. Entry slice effects are principal causes for the high signal intensity of blood vessels on gradient-recalled-echo (GRE) images. This is a result of the fact that the short TR in GRE sequences does not permit the z magnetization to regrow to the values close to its maximum M_{z0} . It is only a small fraction when the next alpha (α) pulse is applied. If a substantial fraction of blood is replaced during the TR of the sequence, entry slice effects can lead to a very strong signal increase. It results in invisible high-velocity signal loss. It is due to the fact that no slice-selective rephrasing pulse is applied. As a result, intravascular signal will be very bright.

3.1.2.1.4 Slice Transition Effects. These slice transition variations measure the flow in a vessel. The velocity of flowing spins depends upon the distance traveled by the flowing spin isochromats and their travel time. Hence, their velocity may be calculated by dividing these two quantities. Velocity may be measured as the number of excited spins present inside the voxel of interest as a function of time. For this, one way is to apply a slice-selective 90° pulse and then to apply a 180° rephasing pulse in the slice-displaced phase along the direction of flow at some distance. Any signal measured in this second-slice duration will represent spin isochromats that have been washed-in by the flow in the vessel. This type of method of determination of the flow velocity is advantageous over the spin-phase method. In this method, flow sensitization occurs by selecting a read-out slice either proximal or distal to the tagging slice. However, the disadvantage of this approach is that it measures the flow in positive, negative, or in both directions. So, clinically this method is not acceptable.

3.1.2.2 Inflow Related Artifacts and Their Suppression

High intravascular signal intensity on GRE images shows the vascular ghost appearance. These ghosts appear due to pulsation artifacts. These pulsation artifacts affect the image quality, if the images are acquired without electrocardiographic (ECG) triggering. These artifacts may be suppressed by the use of phase encoding steps. For this, the principle frequency is selected as superimposed with inspiration rate. It will shift the ghost to the edge of the image. Motion-insensitive pulse sequence is also used to suppress the artifacts. In general, second echo and velocity-insensitive gradients are applied. These methods compensate for high-order motions and thereby minimize the vascular ghosting. Hence, these methods are known as flow-compensated techniques. However, for quantification of vascular features, the spatial-presaturation method is a current practice. This method uses 90° RF pulse. The 90° RF pulse is applied outside the imaged volume. This imaged volume represents the stack of slices in any region. In any selected region, the inflowing spins may produce entry-slice phenomenon. After applying 90° RF pulse, the z magnetization of inflowing spins is completely abolished just before their flow in the imaging volume. This results in no flow-related enhancement effects and the vessels appear black on GRE images. The disadvantage of this method is that it cannot completely suppress signal from stagnant blood imaged with pulse sequences even using the long repetition times (TR).

In order to overcome this problem, MR-projection angiograms of vessels have emerged as an alternative method. In this method, many adjacent thin slices perpendicular to the principal vessel orientation are acquired consecutively. The method is known as 2D Fourier transform gradient recalled echo (2D-FT GRE). As a result of entry-slice phenomenon, the vessels appear very bright in these slices, particularly if they have been acquired using flow compensation to suppress the dephasing. These acquired data are three dimensionally reconstructed using maximal-intensity-projection or surface-rendering methods to generate vascular angiograms. This method is described later in the section on techniques. If presaturation pulse is applied in distal to imaged slice, the signals from veins are suppressed. Similarly, if presaturation pulse is applied in the proximal direction to the imaged slice, it will suppress arterial signals. This method was further refined by the use of two image sequences of the veins and arteries being imaged. It was done by subtracting a flow compensated sequence

from the uncompensated sequence or by subtracting a fully presaturated image from a unilaterally presaturated image. The image acquisition in the interleaved fashion will further minimize the motion artifacts.

3.1.2.3 Spin Phase Phenomenon

This effect is based on the motion in a vessel in the direction of magnetic field gradients. It leads to the precession phases different from zero in bulk motion, while the magnitude of the magnetization vector remains unaffected. All of the moving spin isochromats within the voxel experience the same phase change. Interestingly, the moving fluid will have a different phase. Flowing blood gives rise to a velocity profile in a vessel, divided into different voxels. Due to phase change along the vessel wall and surrounding regions, velocity variation is observed due to phase changes either 90° or 180° . It causes considerable signal loss in the voxel at the location of fat tissue.

Suppose a velocity difference of 1 cm/sec within a voxel produces precessional phase changes of approximately 360° , it will lead to complete signal loss by use of SE sequence with typical gradient values. For slower blood flow, intravascular signal is seen less dephased and is more prominent at the center of the vessel such as accelerated blood flow. With acceleration, the signal loss that results from the dephasing of spin isochromats increases in proportion to the echo number (see Fig. 3.4). For constant velocity motion, this method may be known as even-echo rephrasing or even-echo refocusing for the flow along the

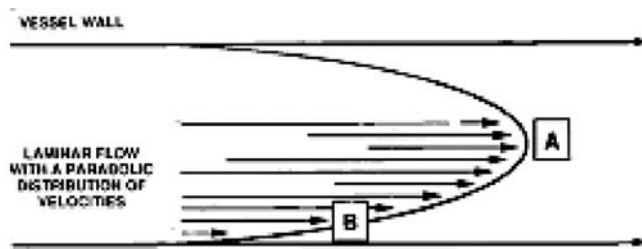


Figure 3.4: Intravoxel spin-phase dispersion due to incoherence is shown near the center of the vessel (point A) for minimal phase dispersion. Point B near the vessel wall encompasses a large range of velocities resulting with intravoxel dephasing and signal loss.

direction of a symmetrical gradient field. This is the case of read-out gradient when multiple echoes with constant echo spacing are obtained. In a multiecho sequence obtained in this fashion, echoes occur at all multiple integers of the echo time (TE) such that even echoes correspond to phase change zero for stationary and moving spin isochromats independent of their velocity. The intravascular signal does not show up and results in no dephasing at even echoes. So, the rephrasing phenomenon is dramatic, as vessels without signal on the first echo can have very bright signal on the second echos or even echos. Arteries do have the pulsatile blood flow and experience the velocity and acceleration dephasing effects. During diastole, arterial blood is almost stagnant and leads to high signal intensity. There is complete signal loss during diastole when blood is moving fast in the veins.

Both slice-transition and spin-phase phenomena are responsible simultaneously for the suppression of intravascular signal. In in-plane flow, spin-phase effects must be dominant, whereas for in-flow perpendicular to the imaging plane, slice-transition effects must be prominent. In gradient echo imaging, intraluminal signal loss occurs due to spin-phase phenomenon. It does not show high-velocity signal loss. This intraluminal signal loss results from spatial variation in velocities and accelerations at the site of increased shear stress in the vascular tree such as arterial bifurcation, arterial proximal parts, pulmonary arteries, and venous confluences around cardiac valves. However, dephasing effects and therefore the signal losses on GRE images are pulse sequence dependent.

In general, MR imaging requires the product of the gradient field G and the time to turn-on (t_p) to be constant for spatial resolution. So, the dephasing effect on constant velocity is dependent on t_{p2} and t_{p3} , and shorter TE. In other words, shorter TE and t_p will result in less dephasing effect and less signal loss. The SE and GRE pulse sequences are standardized for using dephasing effect in the evaluation of valvular diseases.

3.1.2.4 Turbulence-Induced Signal Loss

On GRE images around vascular stenoses, turbulence-induced signal loss further deteriorates the capacity of this approach in flow quantification in valvular diseases. Spin dephasing and consecutive signal loss make the lesions appear more prominent. The turbulence-induced effects are corrected by flow-compensated

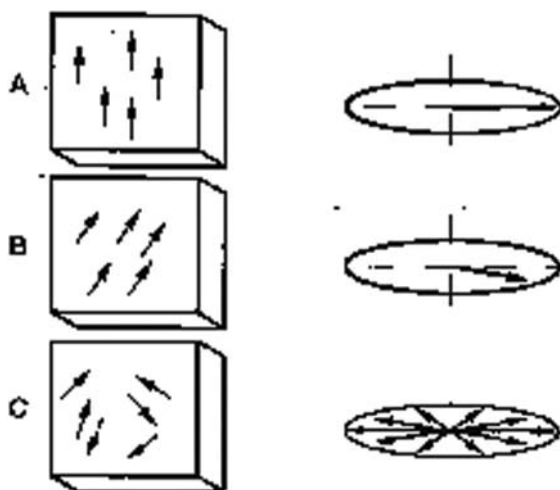


Figure 3.5: Velocity-induced phase shifts. Stationary pins do not show any velocity-induced phase shift or intravoxel dephasing (box A). Constant velocity flow throughout voxel results in net phase shift (box B). Turbulence and shear stress within a voxel produce randomly oriented velocity vectors, resulting in a loss of signal.

sequences at short echo times (see Fig. 3.5). These recent advances are described later in this chapter. The velocity distribution inside the voxels is not the result of an intravascular flow profile, but there are multiple small vessels inside a single voxel that take course in all directions. In the case of vessels taking many turns, intravascular velocities and accelerations inside the voxel are due to spatial arrangement rather than the flow profile. Spin isochromats experience destructive interference of isochromat vector components. This results in signal loss and the magnitudes depend upon the cardiac cycle. These signal losses may be avoided by the use of subtraction of flow-sensitive images from the flow-insensitive images to demonstrate tissue perfusion and tissue diffusion.

3.1.2.5 Quantification of Flow by Spin-Phase Effect

Intravascular signals of voxel within the vessel are characterized by the magnitude and phase angle of the magnetization vector. Flow-compensated pulse sequences are sensitive to velocities because the phase angle is a measure of blood flow velocity or acceleration. The phase angle is adjusted below 360° for

all the possible velocities in the vessels. This concept is also used in Doppler sonography. The sequence uses the gradient reversal. Reversed gradient suppresses the signal from stationary tissue but does not affect phase angle of the flowing tissue. It also corrects magnetic field inhomogeneity and zero phase determination.

3.2 Techniques and Principles of Magnetic Resonance Angiography

There are two main methods of acquiring angiographic images. We shall describe them from the point of view of intravascular flow imaging. Later, we shall describe recent refinements and modifications in the angiography methods.

TOF MRA

PC MRA

Each technique can be performed using 2D-FT. Thus, there are mainly four different methods:

1. 2D-TOF MRA
2. 2D-PC MRA
3. 3D-TOF MRA
4. 3D-PC MRA.

Each of these techniques is advantageous to a different type of clinical application.

Before discussing these techniques, let us understand the basic physical principles of flow inside the vessel which is the fundamental principle of angiography. This information is important for MR instrument physicists in day to day practice.

3.2.1 TOF MRA

This approach is based on flow-related enhancement in which 2D or 3D GRE techniques are applied (see Fig. 3.6). Usually, FC (flow compensation) is used perpendicular to the lumen of the vessel. So, it relies primarily on flow-related

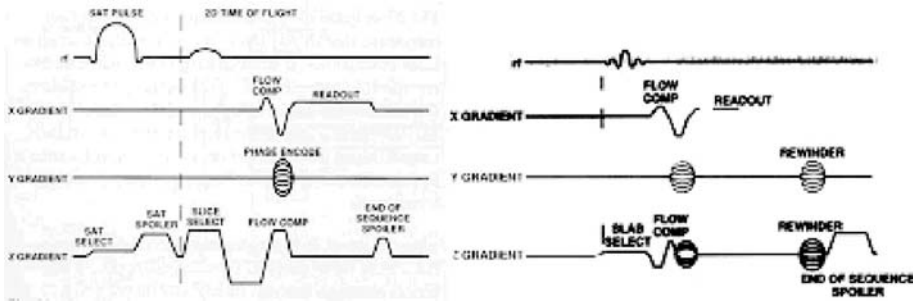


Figure 3.6: Two-dimensional TOF MRA pulse sequence (left) and 3D TOF pulse sequence (right) are shown. Several gradient waveforms are combined to reduce TE.

enhancement to distinguish two types of spins i.e., moving spins and stationary spins. The advantage of this flow-related enhancement is mainly fast scanning time. It uses minimized FRE because each slice represents an entry slice.

3.2.1.1 2D TOF MRA

A typical pulse sequence is represented for TOF MRA. A presaturation pulse is applied above or below each slice to eliminate signal from overlapping venous or arterial structures. Usually a short TR (about 50 msec), a moderate flip angle ($45\text{--}50^\circ$), and a short TE (a few msec) are used. This method has the following advantages. It generates high SNR as signal is generated from a large volume. It improves spatial resolution. During time of flight, blood flowing into the imaging slice is fully magnetized and appears brighter than the partially saturated stationary tissues. Optimized scan parameters to acquire angiographic images are described.

3.2.1.1.1 Image Contrast. To get better image contrast, repetition times (TR) must be kept short with respect to the stationary tissues' T1 times. It helps to suppress the signal from stationary tissue and to maximize the vessel contrast due to flow-related enhancement (see Fig. 3.7). TR of 45–50 msec are adequate for suppressing the signal from stationary spins within the slice. During this time, the fully relaxed blood moving into the slice remains unsaturated. As a result, blood appears bright (high signal intensity) compared to the low-signal-intensity stationary tissues. Other flow and imaging parameters influencing the

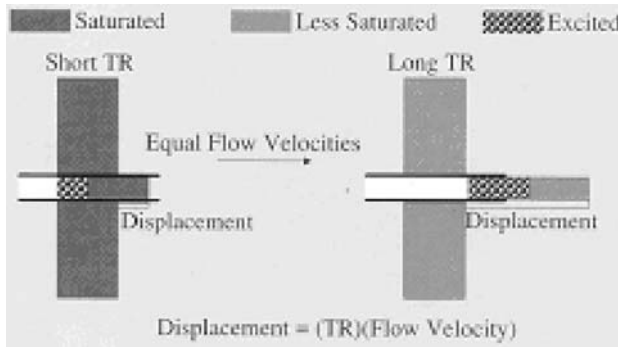


Figure 3.7: Effect of TR is shown. At short TR, stationary tissue is partially saturated causing saturated blood flowing out of slice and replaced with unsaturated blood appearing as high signal in the blood relative to surrounding tissue (on left). At long TR, stationary tissue may recover between excitations and excited blood flows out of the slice before it is refocused to form echo or high signal of stationary tissue with no signal from flow.

image contrast in 2D TOF images include flow velocity and direction, vessel geometry, T1 of blood and stationary tissues, flip angle, TE, and slice thickness.

3.2.1.1.2 Flow Velocity. Initially, flow-related enhancement increases with the flow velocity. At moderate flow velocity rates, there is flow-related enhancement for a complete new set of the spins. Later, no further increase is possible in image contrast or signal intensity.

3.2.1.1.3 Vessel Geometry. The orientation of the blood vessel to the slice plane also affects vascular signal intensity. Maximum inflow enhancement occurs when blood flow is perpendicular to the imaging plane. When a vessel travels obliquely through the slice or the vessel lies within the slice plane, the flowing spins are subjected to multiple RF pulses. As a result, spins begin to become saturated. As a result, intravascular signal intensity decreases and the vessel may be incompletely visualized.

3.2.1.1.4 Slice Thickness. The thinnest slices maximize inflow enhancement. These thin slices reduce the effects of in-plane flow. Typically, for our carotid bifurcation imaging, a nominal 1.5 mm slice is obtained using a narrow bandwidth RF pulse at 625 Hz and gradient amplitudes of 1 G/cm or 10 mT/m.

The larger slice-select gradient amplitudes require larger flow compensation gradients. These gradients in turn limit the minimum TE to 8–9 msec.

3.2.1.1.5 Longitudinal Relaxation Times (T₁). T₁ values of blood may have a significant effect on intravascular signal intensity, particularly in slow-flow conditions. Suppose the velocity is not sufficient to completely refresh spins within the imaged slice; in this case the blood will begin to become saturated. In the 2D TOF procedure described for the carotid bifurcation, saturation would occur at flow velocities of approximately 3 cm/sec. It can be represented as 1.5 mm divided by 50 msec. The ability of TOF angiography to detect slow flow may be improved by shortening the T₁ of blood through the use of MR contrast agents.

3.2.1.1.6 Flip Angle. Flip angle affects the slice-saturation rate. It is selected to saturate the stationary tissues without compromising intravascular signal intensity. Normally, flip angle of 45–60° is selected for 2D TOF angiography.

3.2.1.1.7 Asymmetric Echo Acquisitions. These acquisitions are obtained using the fractional echo or partial echo times. Shortest echo times are obtained for optimizing MR angiographic images. These short echo times are obtained by the use of asymmetric echo acquisition. In this approach, the gradient echo may offset in the acquisition window by shortening the duration and size of the dephasing and flow-compensation gradient pulses. This permits shortening of the sequence and reduced echo times. Asymmetric echo acquisition has the advantage of reducing the size of the refocusing gradients on the read-out axis. It results in decreased signal loss and reduces artifacts from accelerations or higher order moments.

3.2.1.2 Presaturation Pulses

Blood may flow into the imaging plane from any direction. The blood may produce flow-related enhancement i.e. 2D TOF images of carotid artery show blood flowing to the imaging plane from above or below. As a result, the carotid arteries appear bright. This brightness is due to overlapping of the carotid arteries and jugular arteries. To eliminate the signal from overlapping vascular structures in the projection images, a presaturation pulse is applied as a 3 cm wide saturation band at the level of 0.5 cm superior to the slice. The saturation pulse

moves superiorly with each successive tissue slice. As a result, the image data set emphasizes exclusively arterial structures.

3.2.1.3 2D TOF Angiography of the Carotid Bifurcations

In patients with vascular diseases, the 2D TOF imaging technique is an effective method of imaging the carotid artery bifurcation. We acquired typically 50–70 contiguous axial slices, each approximately 1.5 mm thick. The acquisition is performed by using flow compensation in both slice-select and read-out directions. For this, typically a gradient echo pulse sequence is employed, with $TR = 45\text{--}50$ msec, a flip angle of $45\text{--}60^\circ$, $NEX = 1$, 128×256 matrix, and minimum available echo time. The field of view (FOV) may vary from 16 to 20 cm, depending on the patient size. As a result, axial image slices show the blood vessels as bright (see Fig. 3.8). Other surrounding tissues appear with much lower signal intensity. However, the 2D TOF angiography method has limitations.

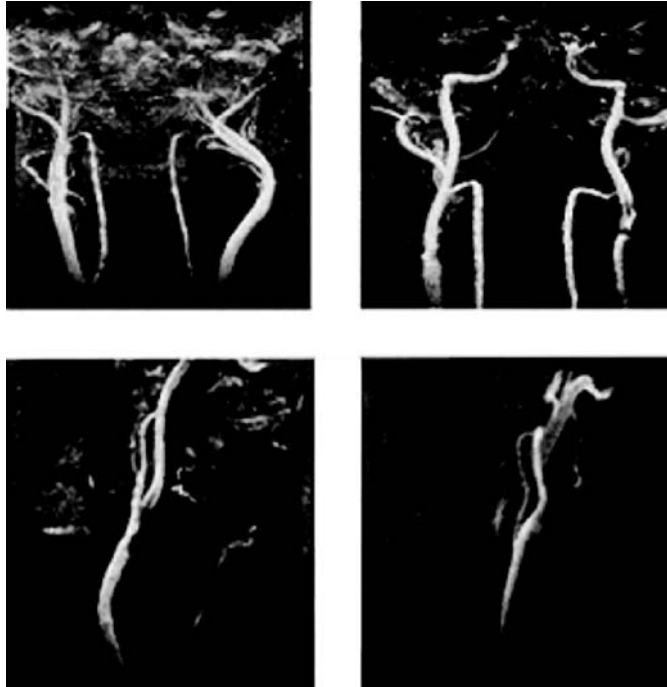


Figure 3.8: In carotid artery, glomus tumor vasculature is shown in pre- and postsurgery (left and right panels at the top). Carotid stenosis (left on bottom) and carotid aneurysm (right at bottom) are highlighted.

The presence of vascular stenosis may produce areas of increased flow velocity within the stenosis segment. So, these regions (having relatively less flow) further slow the flow in the poststenotic vortices. Turbulence is also frequently encountered distal to the stenosis. These factors will produce intravoxel incoherence, spin saturation in the slowly flowing eddies, and loss of signal intensity. Large ulcerations are often difficult to image because of the slow and complex flow within the location of ulceration. Use of the shortest TE and smallest voxel size can substantially minimize signal loss. In spite of these measures, some signal will be lost adjacent to the stenosis region. It is possible to overestimate the degree of stenosis when interpreting the carotid MR angiograms (see Fig. 3.9). In-plane flow can also compromise the quality of 2D TOF angiograms. When a blood vessel runs parallel to the imaging plane, the blood will experience multiple RF pulses. As a result, the blood will eventually become saturated. The portion of the vessel coursing through the imaging plane may exhibit little or no signal intensity. These vessels appear as artificially narrowed or



Figure 3.9: Three-dimensional coronal inflow targeted MIP angiography of carotid arteries is shown. Boxes in both rows represent isolated single carotid arteries at 1.5 mm 32 partitions.

discontinuous. Artificial loss of signal is usually easy to identify if caused by the in-plane flow. It is due to the fact that the vessel geometry can be appreciated above and below the in-plane segment. On several occasions, vessels may become so intertwined that vascular loops are formed. Presaturation pulse obliterates the signal intensity from inferior vessel taking course. Similarly, reversed flow in the internal or external carotid arteries will be undetectable. The invisibility is because of the superior presaturation pulse. For better results, the maximum intensity pixel (MIP) approach was suggested as described in the following section.

3.2.1.4 2D-TOF Imaging of Venous Anatomy

Images of cortical veins have been generated with the use of 2D TOF imaging. When superimposed on axial MR images, a map of the cortical veins can be produced to relate intraparenchymal pathology with cortical venous anatomy. This approach has been successful for preoperative localization of lesions prior to surgery. For this, a presaturation pulse is applied to eliminate arterial signal using slice thickness 2.0–2.9 mm. The method allows the relatively fast coverage of a large anatomical region without compromising in-flow enhancement. The trade-off in this approach is the decrease in image-resolution.

3.2.1.5 MIP Projection Ray Tracing Technique

The MIP projection ray tracing technique is used for viewing 2D TOF angiograms after data acquisition and reconstruction (see Fig. 3.10). As a result, multiple projection images are generated from volume of stacked axial slices' image data. When volume data is projected onto a two-dimensional plane, each pixel in the projection image depends on the pixels along each line, or ray, through the volume of data. Several procedures can be used to determine pixel intensity in the projection image. Using MIP, the projection pixel is assigned the maximum pixel intensity found along a ray traversing the imaging volume. Additional parallel rays are passed through the volume until a complete projection image of the vessels is obtained. Once all pixel intensities for a projection have been calculated, the process may be repeated for any other projection angles (see Fig. 3.11). Typically, 18 projection angiograms are obtained at 10° increments. The collections of projected images may be viewed as a cine loop to give the

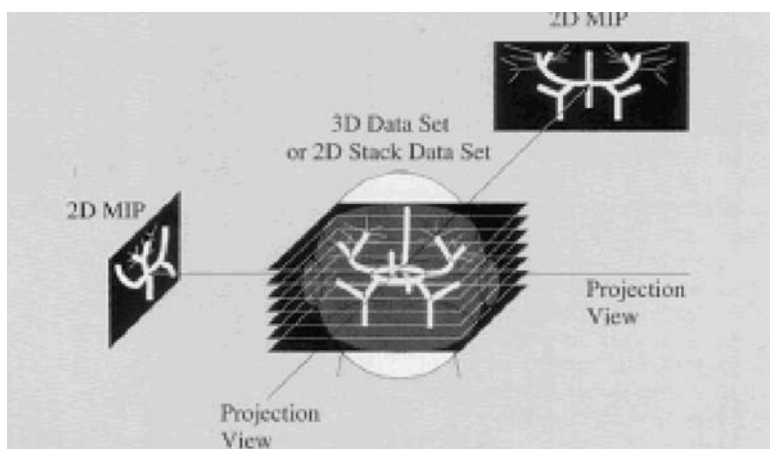


Figure 3.10: Maximum intensity projection: 3D structure is represented on a 2D plane based on the maximum signal intensity. Projecting successive views from different angles results in an apparent rotation when later displayed in a cine loop.

appearance of rotation and depth. These images result in a three-dimensional representation of the vascular structures. Angiographic imaging parameters are selected to maximize the signal intensity of vascular structures. For this, the MIP procedure produces projection images in which vessels appear bright and the background signal intensity is reduced. The MIP technique has limitations despite improved contrast. It results in a slight decrease in vessel size. So, this technique contributes to an overestimation of stenotic regions. The MIP process also slightly reduces the diameter of normal vessels. The MIP projection images lack increased signal at points of vessel overlap. A ray tracing through a region of vessel-overlap selects the brightest pixel along the ray. It presents the overlapped vessels as a single vessel.

3.2.2 3D TOF MRA

A pulse sequence is represented for 3D TOF MRA (see Fig. 3.6). A slab of several cm (usually about 5 cm) is obtained which contains up to 28–60 slice 3D volumes in axial plane through region of interest. The slice thickness is 0.7–1.0 mm, repetition time is 40 msec, and flip angle is 15–20° with FOV of 16–20 cm, depending on the patient size and region of interest. Depending upon the desired resolution

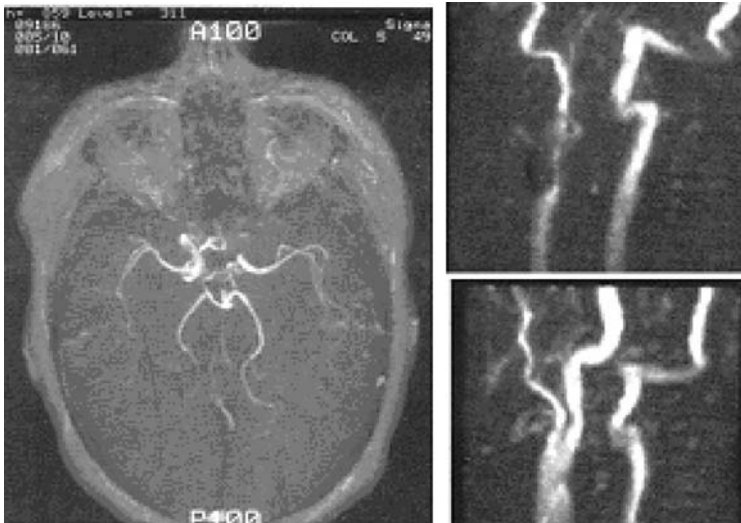


Figure 3.11: Three-dimensional TOF angiogram showing circle of Willis confirming the occlusion of the left internal carotid artery (left panel). Two-dimensional TOF angiograms demonstrating an internal carotid artery occlusion. A sagittal projection of right carotid bifurcation reveals a patent vessel post endarterectomy (top row on right). The sagittal projection of the left carotid bifurcation reveals stenosis of the proximal external carotid artery and occlusion of the internal carotid artery (bottom row on right).

and imaging time, 128×128 , 192×256 , or 256×256 matrix can be used with $NEX = 1$. Very short echo times may be attained with flow compensation. These optimized scan parameters permit adequate penetration of inflowing, fresh, fully magnetized spins into the imaging volume. The resultant 3D data set initially is displayed as a series of slices, acquired in the axial plane. Later, it is subjected to the MIP ray tracing technique to create coronal and sagittal projections. A series of projections may also be generated to “rotate” the vascular structures around a single axis. Cine loop display can provide the perception of depth. Advantages of 3D techniques are appreciable as these techniques are more susceptible to saturation effects and less sensitive to slow flow. Thus, 3D volume acquisition techniques offer superior signal-to-noise ratios (SNR). 3D TOF MRA offers a prescription of very thin slices, thereby reducing the voxel size and decreasing the intravoxel dephasing. 3D TOF MRA maximizes the flow-related enhancement.

3.2.2.1 Optimization of Image Parameters of 3D TOF MRA

Optimization parameters are blood velocity, vessel orientation in relation to the slab, the size of the imaging volume, TR, slice thickness, voxel size, and flip angle. Flow velocity should ideally permit fresh, fully magnetized spins to traverse the entire imaging volume between successive RF pulses. This results in optimal signal enhancement because of in-flow effects. For instance, at normal flow velocity saturation effects will be minimal. At lower velocity, slow flowing blood becomes saturated as it moves through the imaging volume, and signal intensity decreases. Slow flow conditions may be encountered in the cases of vascular occlusive disease, venous thrombosis, and aneurysms with complex flow patterns.

3.2.2.1.1 Imaging Flow Orientation. It should be selected to minimize the saturation of moving spins as they course through the volume. For instance, axial orientation permits imaging of 'circle of Willis' using a small volume, thereby reducing the imaging time (see Fig. 3.11). In practice, coronal and sagittal orientations have been used to image both extracranial and intracranial carotid arteries in a single acquisition. Larger flip angles of 35–60° maximize signal in the extracranial carotids, but result in saturation of the intracranial vessels. Smaller flip angles of 15–30° improve visualization of the intracranial vessel because of the reduced saturation. As a result, trade-off is the decreased intensity of intravascular signal from the extracranial carotid arteries.

3.2.2.1.2 Repetition Time (TR). At short TR, stationary tissues exhibit greater saturation. It increases the tissue contrast between vessel and the surrounding tissues (see Fig. 3.12). However, at short TR, spins flowing through the imaging volume become saturated, resulting in loss of intravascular signal intensity. These saturation effects can be somewhat reduced by using a smaller flip angle or by shortening the T1 of blood through the use of MR contrast agents. Nonetheless, when the 3D acquisition is optimized for normal intracranial arterial flow (flip = 15–20°, TR = 40), slower flow will become saturated, reducing the delineation of venous anatomy and slow flow within aneurysm or diseased arteries. Despite this, 3D TOF MRA does not distinguish flowing spins from subacute hemorrhage. For instance, methemoglobin within a subacute hematoma has a short T1 and does not become saturated during the 3D acquisition. The

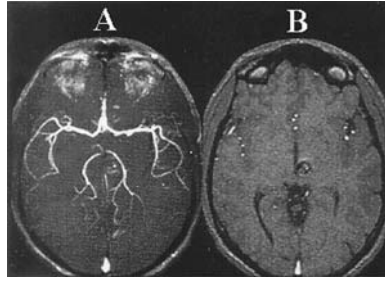


Figure 3.12: Three-dimensional TOF angiogram (left panel) shows cavernous angioma with visible methemoglobin due to short T1 due to simulated blood flow. For comparison, SPGR images are shown with high signal intensity center representing methemoglobin.

result is bright signal intensity in the images, which may simulate flow-related enhancement.

3.2.2.1.3 Echo Time (TE). Lower TE reduces motion-induced phase errors. Partial RF pulses reduce the minimum TE while these RF pulses preserve an acceptable slab profile. Very low TE may be achieved by removing flow compensation from the gradient waveform. Thus there is a trade-off between minimum echo time at the cost of flow compensation. This approach is currently used for clinical imaging.

3.2.2.1.4 Flip Angle. Flip angle has an effect on intravascular signal intensity and background suppression. Smaller arteries may be visualized at flip angles of $15\text{--}20^\circ$ with TR of 40 msec. Stationary tissues exhibit greater saturation at a larger flip angle. For example, small 3D volumes of 28 slices show intravascular signal intensity of larger arterial structures at flip angles $20\text{--}35^\circ$ with rapid flow. Arterial flow begins to saturate at flip angles greater than 40° . It results in reduced intravascular signal intensity (see Fig. 3.13).

3.2.2.1.5 Flow Compensation. Flow compensation is critical in 3D TOF MRA. Motion-induced phase dispersion results in signal void areas. These areas are frequently identified within the juxtaseptal carotid arteries and proximal middle cerebral arteries. These signal void areas can be minimized by the use of shortest possible TE with flow compensation applied in the slice-select and read-out directions. This combined approach reduces the phase dispersion and

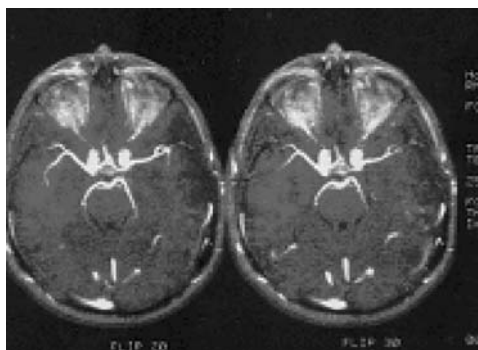


Figure 3.13: Effect of flip angle in 3D TOF angiogram images shows at different flip angles 20° (left) and 30° (right).

therefore maximizes intravascular signal. Intraluminal signal loss may still occur in spite of the use of first-order motion compensation. High order motions such as jerks and acceleration may still produce regions of signal loss due to the phase dispersion. For instance, blood flow in carotid siphon experiences centripetal acceleration along the carotid vessel's outer wall. In 3D TOF images, the effects of acceleration are not compensated and result in linear regions of signal loss at curves in the carotid artery and proximal middle cerebral artery. However, magnetic susceptibility effects from the adjacent paranasal sinuses play a minor role in the loss of signal intensity in the juxtaseellar carotid artery and proximal middle carotid artery at short TE. Mostly, signal intensity losses are observed at the bends of these arteries during diastole. Higher order motion compensation gradients extend TE. At extended TE, susceptibility effects are significant and the signal loss is more apparent.

3.2.2.1.6 Slice Thickness. Slice thickness also contributes to signal loss. Thicker slices show significant signal loss. Thin slices exhibit phase dispersion within the voxel which minimizes signal loss and effects of intravoxel dephasing. However, thin slices reduce signal-to-noise ratio and the volume of interest.

Other important refinements in this technique are described in Section 3.4.

3.2.3 Phase Contrast MRA

Phase contrast (PC) MRA is based on the fact that the phase gain of flowing blood through a gradient is proportional to its velocity (assuming constant velocity).

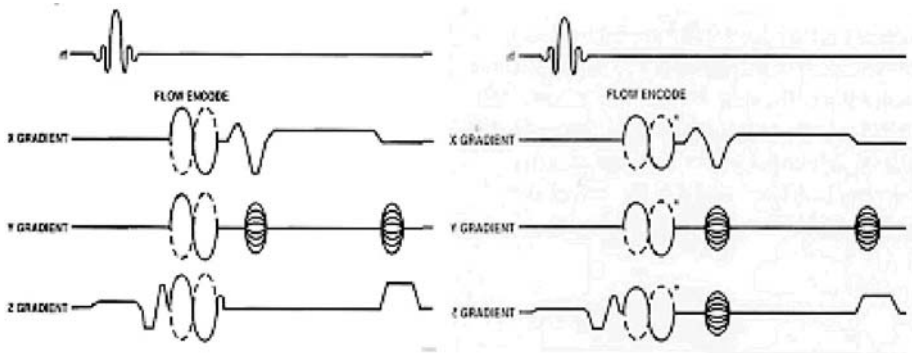


Figure 3.14: Two-dimensional Phase contrast pulse sequence (left) and 3D phase contrast pulse sequence (right) are shown with velocity-induced phase shift to distinguish stationary and flowing spins. In both 2D/3D PC MRA, two or more acquisitions with opposite polarity of the bipolar flow-encoding gradients are subtracted to produce image of vasculature while these gradients are not applied to all three axes simultaneously.

Phase (ϕ) and velocity (v) are related by

$$\phi = \int \omega dt = \int (\gamma G v t) dt = \frac{1}{2} \gamma G v t^2 \quad (3.13)$$

Therefore, knowledge of the phase at any point in time allows us to calculate the velocity. The most common method for PC MRA is the use of bipolar gradient (see Fig. 3.14). This process is called flow encoding. Because the two lobes in this bipolar gradient have equal areas, stationary tissues observe no net phase change. However, flowing blood will experience a net phase shift proportional to its velocity (assuming a constant flow velocity). This is how flow is distinguished from stationary tissue in PC MRA (see Fig. 3.15).

PC MRA is illustrated for 2D PC and 3D PC MRA, respectively in the following section. At this point, it is important to describe “flow phase,” “velocity-dephasing,” and the distinction between “magnitude” image and “phase” image.

Flow image results from phase changes in transverse magnetization of spins moving along a magnetic field gradient. These phase shift effects can be used to generate flow images to quantify flow velocities. These phase effects are also present in stationary spins due to differences in their precession frequency. Stationary tissues dephase over time in a spatially-dependent magnetic field gradient. This dephasing can be exactly compensated to form an echo using a

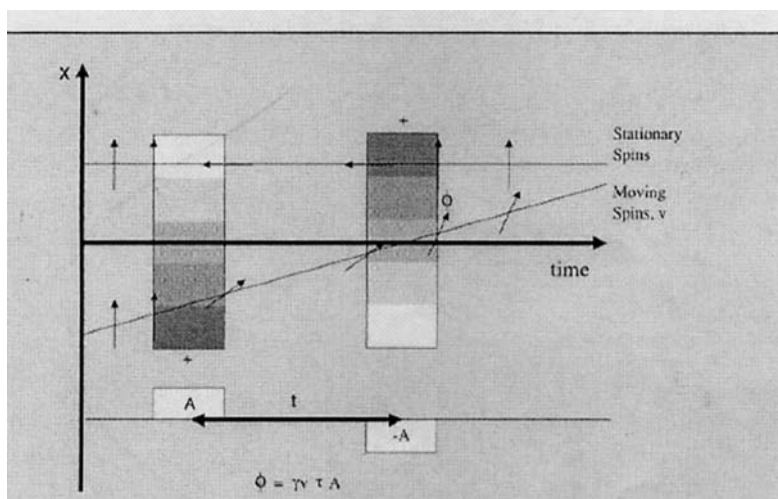


Figure 3.15: Flow is encoded in one direction using bipolar gradients. Through encoding, stationary tissue receives zero phase shift, ϕ , while moving spins receive a phase shift proportional to their velocity, v .

second gradient in the opposite direction. Flowing spins change the position during application of the dephasing and rephasing gradients (see Fig. 3.16). As in PC MRA, both magnitude and phase images can be obtained with information on direction of flow in the vessel (see Fig. 3.17).

The effect of velocity-dephasing on phase information indicates the flow direction in right-left (R/L), superior-inferior (S/I), or anterior-posterior (A/P). This effect may be described as follows: flow-induced different phase shifts are generated due to spins moving in-plane along frequency gradient in different directions (see Fig. 3.18 shown by zig-zag arrows). Conventionally, spin flow is higher at the center than near the wall due to laminar flow. Due to this difference, spins at the center cause larger phase shift than the phase shift by slower peripheral coherence. This results in velocity dephasing and total signal loss called “flow void.” In this way, phase information is transferred to a magnitude contrast.

3.2.3.1 2D Phase Contrast Angiography

The primary advantage is that a variety of velocity encoding may be opted in a short period of time (within a few minutes). If limited angiographic information

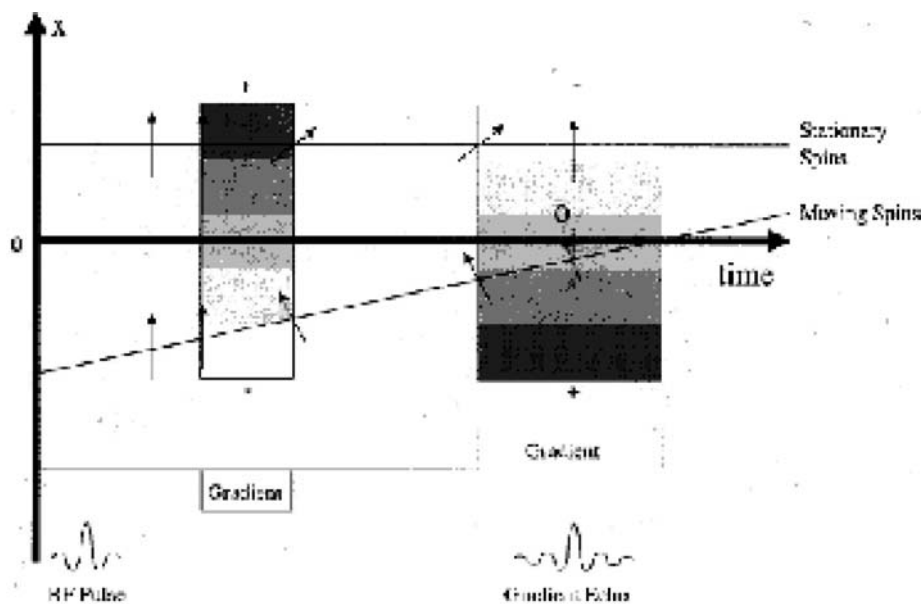


Figure 3.16: Flow phase dephasing and rephasing: stationary spins are dephased between the first gradient pulse that later rephases in opposite direction at the echo time. Moving spins acquire an additional phase shift ϕ as they move along a spatially dependent gradient.

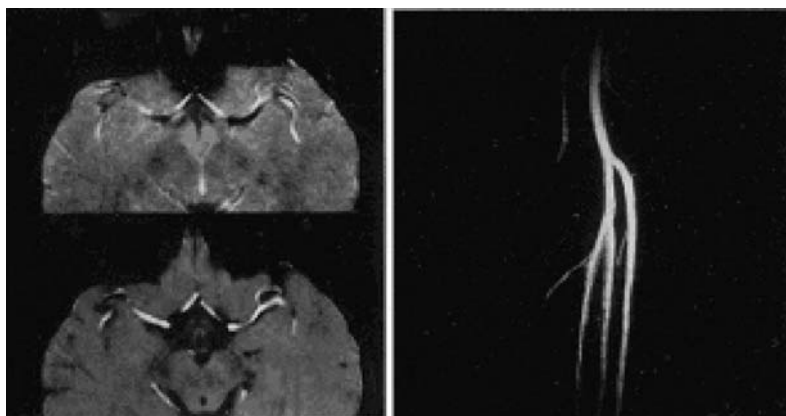


Figure 3.17: Two-dimensional MRA (left on top) and 3D MRA (bottom on left) angiogram images are shown to highlight the limitations and advantages of each. Magnitude contrast provides signal enhancement over long distance (on right).

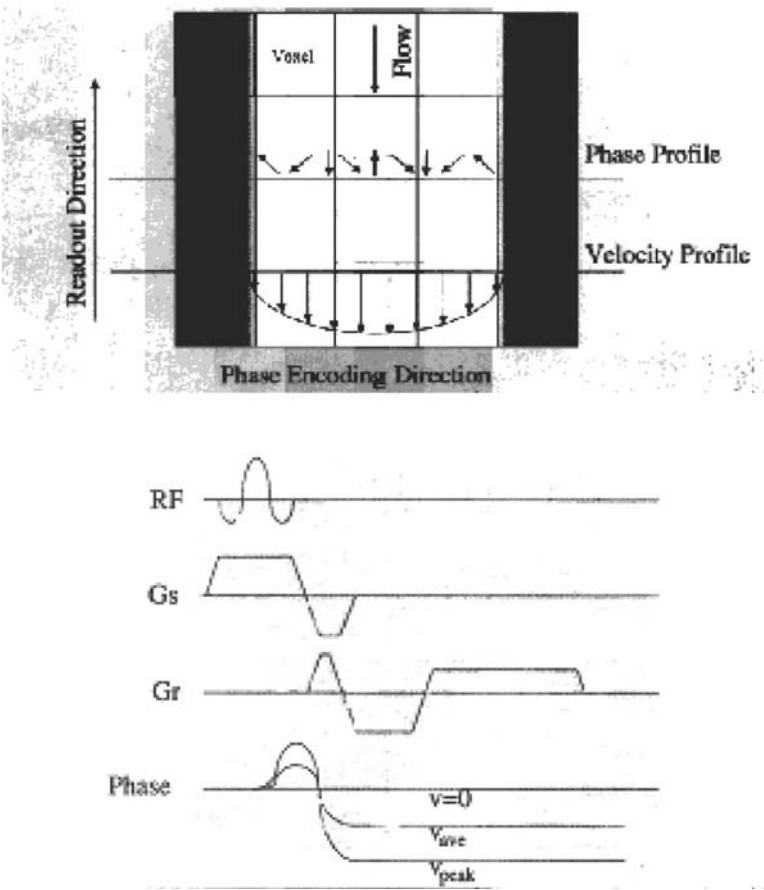


Figure 3.18: (top): Velocity dephasing: Moving spins show different velocity profile. They acquire a net velocity dependent phase shift indicated by the zig-zag arrows. Different flow phases within one volume element cause signal attenuation in that region. A representative RACE pulse sequence is shown (bottom).

is needed, 2D phase contrast images may be sufficient to define the anatomy of interest. To overcome this problem, RE pulse sequence is used to generate velocity profiles inside the vessels (see Fig. 3.19). It uses slice selective gradient and modulated RF pulse to excite a slice perpendicular to the flow direction. An echo is read out thereafter. Spins flowing in this slice experience a phase shift moving along the slice selection gradient because a read-out gradient is also used perpendicular to flow. It allows correlation of phase shift with position across the vessel diameter for generating a phase-shift profile which is linear with velocity profile (see Fig. 3.19). Another advantage of 2D phase contrast

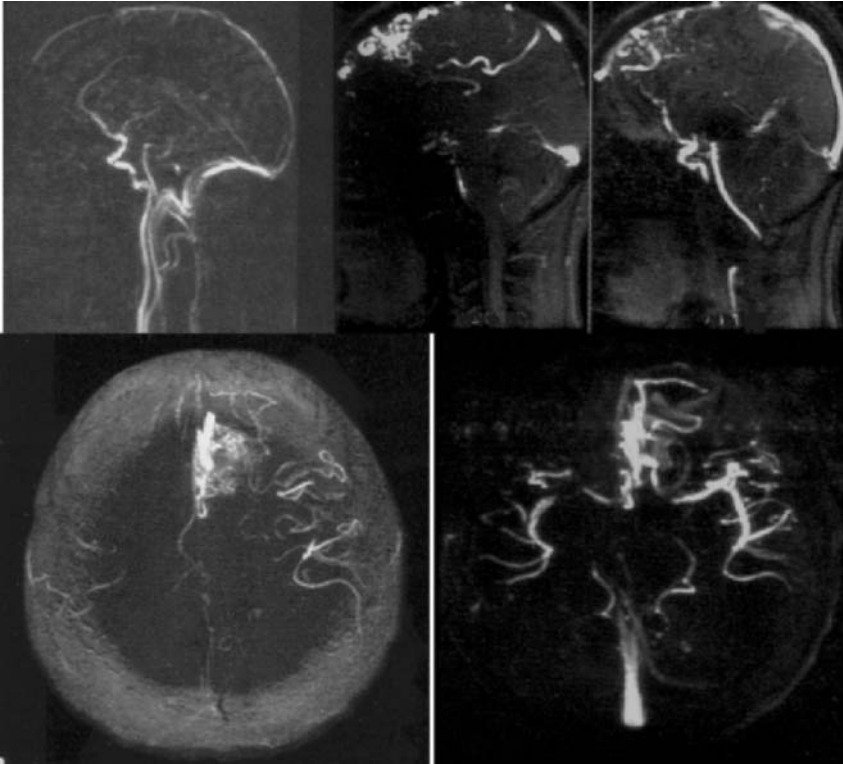


Figure 3.19: Two-dimensional phase contrast localizer as guide image to apply multiple velocities in a short period of time (left panel on top row). Sagittal 2D phase contrast angiogram images are shown at VENC 20° to highlight draining veins of AVM (panel in center) and VENC 80° to highlight arterial supply (right panel on top row). Axial collapsed image from 3D TOF angiogram shows clear delineated AVM nidus (left on bottom row). Axial collapsed images from 3D phase contrast angiogram is shown with greater signal intensity in the left hemisphere arterial structures (right panel on bottom row).

angiography is that, by varying the VENC, the user can generate images of the arteries or the veins. Vascular occlusions can also be confirmed by imaging at slower flow rates. However, 2D techniques are faster than 3D techniques while 3D PC MRA generates better SNR.

3.2.3.1.1 Image Acquisition. Here bipolar phase-encoding gradients are used in a fashion analogous to 3D PC techniques. Instead of 3D volume, however,

the system collects and displays the data as a series of thick slices or a single slab. The slices, or slab, are then projected onto a single plane. The most efficient way to acquire 2D PC angiogram is to use the shortest possible repetition time and a large number of excitations, e.g. $NEX = 2-16$.

The optimized NEX permits data emerging during an R-to-R interval for each set of the phase-encoding gradients. Since the data is averaged during the entire cardiac cycle, the resulting angiogram will be a measure of the average flow. Averaging also reduces ghosting artifacts. 2D phase contrast images may also be obtained without the use of flow compensation, to minimize echo time.

3.2.3.1.2 Dynamic-Range Compression. Compared with the signal coming from all the other spins in the regions being imaged, the MR signal from moving blood is very small. Since the primary mechanism of stationary-spin suppression is the subtraction of two excitations drawing most of their MR signal from nonmoving spin, minor errors in the stationary-spin signal prohibit good background suppression.

To overcome this problem, a projection-dephasing gradient can be applied to diminish signal from thick objects. This gradient has little effect on vascular signal because the vessels are small with respect to the head and neck. For this reason, the signal from stationary tissues is substantially suppressed, while the MR signal from vessel is only slightly diminished (see Fig. 3.20). This results in a reduction in dynamic range (an important imaging enhancement for PC angiography). Projection dephasing may also modify the appearance of vessels separated in the direction of projection. Under these circumstances, the region of overlapped vessels may exhibit enhanced signal intensity, reduced signal intensity, or no change in intensity.

3.2.3.1.3 Cardiac-Gated 2D Phase Contrast Angiography. Cardiac gated 2D phase contrast angiography is based on cine MR acquisition methods in which TR remains constant, and each step in phase encoding is initiated by the ECG trigger. The cine gradient echo pulse sequence is modified to include bipolar gradients for positive and negative flow encoding in a fashion analogous to multislice/slab 2D phase contrast angiography. The velocity encoding is also similar to nongated 2D PC MRA. With this approach, up to 32 points in the cardiac cycle are retrospectively sorted from the scan data. Magnitude and phase images are then generated for each point in the cardiac cycle. On the phase

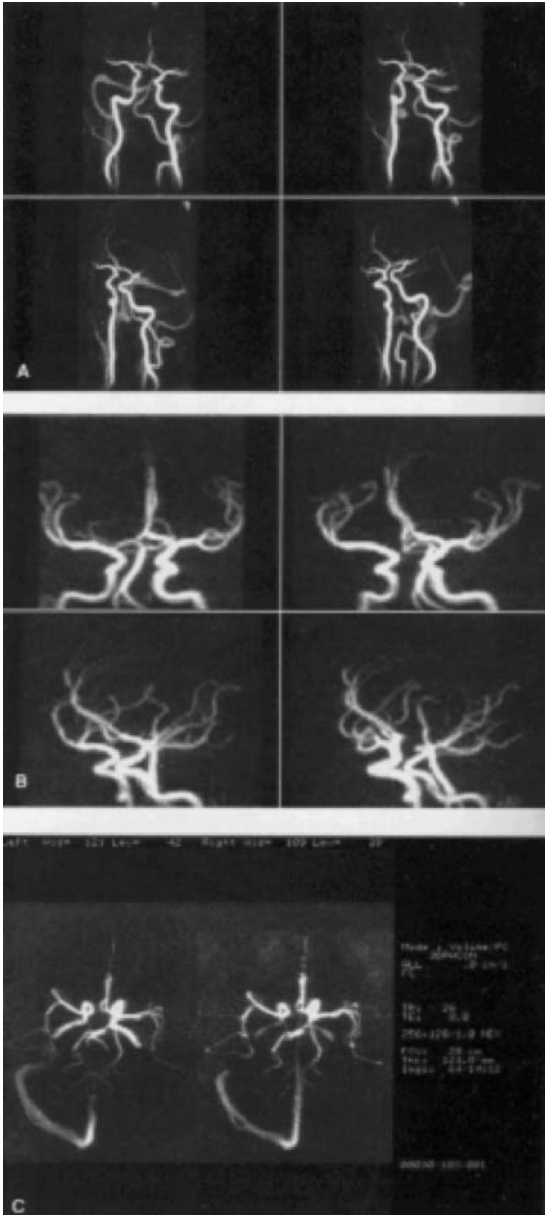


Figure 3.20: Three-dimensional Phase contrast angiograms are shown as normal four projection images for normal vasculature with no signal loss (panels A); 3D PC angiogram of intracranial arteries showing well visualized middle cerebral artery and anterior cerebral artery (panels B); 3D PC angiogram axial projection image from 2 mm slice (left) and 5 mm slice (right) (panels C). Using single projection, thicker slices can be achieved at less imaging time.

images, signal intensity is proportional to blood flow velocity for each phase of the cardiac cycle. The phase images also display the direction of blood flow; a bright vascular structure represents flow in the same direction as the flow-encoding gradient, while dark vessels indicate flow in the opposite direction. Scan information may be acquired in a multislice or projection/slab format. This technique allows the assessment of alterations in hemodynamic conditions during the cardiac cycle. Phase contrast angiograms may also be obtained using a projection technique, to create a single-projection angiogram through a selected volume of interest.

3.2.4 3D Phase Contrast Angiography

3D PC MRA technique relies on velocity-induced phase shifts to distinguish flowing blood from surrounding tissue. The phase contrast approach can be made sensitive to slow flow in small vessels. The reason for this can be attributed to the fact that the contrast between flowing blood and stationary-tissues is related to blood velocity, rather than stationary-tissue T1. PC angiography also permits excellent cancellation of stationary tissues and offers the potential for quantitative measurements of blood velocity.

3.2.4.1 Image Acquisition

To understand the mechanisms of image acquisition during phase contrast vascular imaging, conventional, rotating-frame vector of spin magnetization provides a simple picture. This is described in the following section. After a 90° pulse has been applied to the equilibrium magnetization, the spin is rotated into the transverse plane, on an axis perpendicular to the static magnetic field B . Here, it precesses at the Larmor frequency; at any moment in time, the magnetization can be described by the length of the magnetization vector and its phase in relation to reference positions. As this precession continues, the phase of the magnetization vector or the phase angle varies. It is primarily because of the effect of the main magnetic field, although smaller contributions may also change the rate of phase evolution. For example, if the local magnetic field of spin is changed by a magnetic field gradient, the spin's Larmor frequency will be slightly different and the rate of change of the phase will be altered. To detect flow, phase contrast angiography uses a bipolar gradient to encode a spin's

velocity as a change of phase. The phase accumulation associated with such a gradient is expressed as:

$$\phi = \gamma V \cdot T \cdot A \quad (3.14)$$

where ϕ is phase shift induced by flow in the transverse spin magnetization, γ is the gyromagnetic ratio of the spin, V is the component of the spin's velocity in the applied gradient's direction, T is the center-to-center time interval between the two gradient lobes, and A is the area of each gradient lobe. This equation describes only the phase shift induced by constant velocity flow when a bipolar gradient is applied and not phase shifts due to such higher orders of motion as acceleration or jerk. Since the flow-induced phase shift is directly proportional to velocity, a stationary spin with zero velocity will have no net phase accumulation. For subsequent acquisitions, this pulse sequence inverts the polarity of the bipolar flow-encoding gradients. The polarity of the gradient (A) is now negative, giving the equation for the second acquisition as $\phi = -\gamma VTA$. When the image data from the first acquisition is subtracted from the second acquisition, the remaining data is from the signal that is different in two acquisitions i.e., the intravascular signal from moving blood. The procedural difference in these two acquisitions is the negation of bipolar gradients. A stationary spin will have identical (zero) phase shifts for each polarity of the flow-encoding pulse, resulting in a zero net phase shift. So, the subtraction of two vectors result in zero. The vector subtraction of signals from the spins moving with constant velocity is quite different.

Suppose two signals have the same magnitude but different phases. Consequently, when the vectors are subtracted, the resulting vector is not zero. The result is signal originating from vascular structures with nearly complete elimination of stationary tissues from the MR angiogram. In MRA, the imager acquires the equivalent of three raw data sets for three flow-encoding directions. The magnitudes of these data sets are combined into a total flow angiogram.

3.2.4.2 Image Contrast

Image contrast in PC angiography is influenced by several factors such as flow direction, velocity encoding and aliasing, phase dispersion and flow compensation, and saturation effects.

3.2.4.3 Flow-Encoding Gradients

In PC angiography, the bipolar flow-encoding gradients may be applied in a single direction (e.g. superior/inferior) or in all directions—S/I, A/P, and R/L. In some anatomic regions, the carotid bifurcation such as application of a single flow-encoding axis may be sufficient. The resulting images represent flow direction by the sign of the pixel value in an S/I flow image, for example superior-to-inferior flow is represented by positive pixel values (brighter), while inferior-to-superior flow is represented by negative pixel values (darker). In such a case, single flow-encoding directions will not be adequate i.e., intracranial MRA shows blood flow components in all directions. However, the total flow image can be obtained by measurement of individual flow components and combining them mathematically into a composite image called “velocity image.” This image is made of flows in multiple directions and has magnitude (in cm/sec) but no specific direction. Velocity is defined as a vector with a magnitude (in cm/sec) and direction such as S/I, A/P, and R/L. The individual flow measurements can also generate a phase image with velocity and directional flow information. In the phase contrast angiograms, display pixel values are proportional to the product of image magnitude and velocity encoding. This relationship of velocity with image magnitude provides quantitative measurement of velocity.

3.2.4.4 Spatial Misregistration Effects

The reason for spatial misregistration artifacts can be understood with the pulse sequence. In this sequence, phase encoding fixes the position of an isochromat in the phase-encoding direction, which occurs shortly after the 90° nutation pulse. This fixing of the isochromat position in the read-out direction is followed by read-out which occurs only at echo time (e.g. approximately at TE, 2TE, etc.) after phase encoding. If spin isochromats move between these two events in an oblique in-plane direction, their signal is misregistered. Spatial misregistration occurs because the position of the flowing isochromats is identified in the phase-encoding direction prior to the read-out direction. The result is a shift in intravascular signal intensity in the direction of flow along the read-out gradient. Measurements of the displacement of the signal delineating an apparent vessel and the angle between vessel and

read-out direction are used to determine flow velocity. In quantitative terms, the time difference Δt is the time between the phase-encoding and the read-out events. The distance A is the measurement by which the signal is displaced outside a vessel. Measurement of the angle C is the angle between the vessel and the read-out direction, which permits determination of the flow velocity V :

$$V = A/(\Delta t \cos C) \quad (3.15)$$

3.2.5 Velocity-Encoding and Aliasing

Flow encoding in a vessel can be called velocity encoding (VENC). It is a parameter that is selected by the MR operator when using PC MRA. VENC is the maximum velocity present in the imaging volume. Any velocity greater than VENC will be aliased according to the following formula: aliased velocity = VENC – actual velocity. A small VENC is always more sensitive to slow flow (venous flow) and to smaller branches, but it causes more rapid (arterial) flow to get aliased. A larger VENC is more appropriate for arterial flow. So, small and large VENC are important for imaging all flow components. This method has several advantages. PC MRA is capable of generating magnitude and phase images with superior background suppression. VENC is less sensitive to intravoxel dephasing or saturation effects. On the other hand, this method suffers from several disadvantages such as long scan time, sensitivity to signal losses due to turbulence and dephasing on vessel turns (carotid siphon), and dependence on maximum flow velocity in order to select an optimum VENC. To provide quantitative information regarding velocity in PC angiography, the VENC should be selected to encompass the highest velocities that are likely to be encountered within the area of interest (see Fig. 3.21). The normal maximal flow velocities are likely to be encountered within the vessel region of interest. The normal maximal flow velocities of intracranial arteries do not exceed 80 cm/sec. So, the VENC of 80 cm/sec would encompass all flow velocities up to and including 80 cm/sec. When a velocity encoding is selected, the amplitude of the bipolar flow-encoding gradients is adjusted so that all velocities including the selected value can be imaged without aliasing. Aliasing in phase contrast occurs when high flow velocities are incorrectly represented in the velocity image as lower flow velocities.

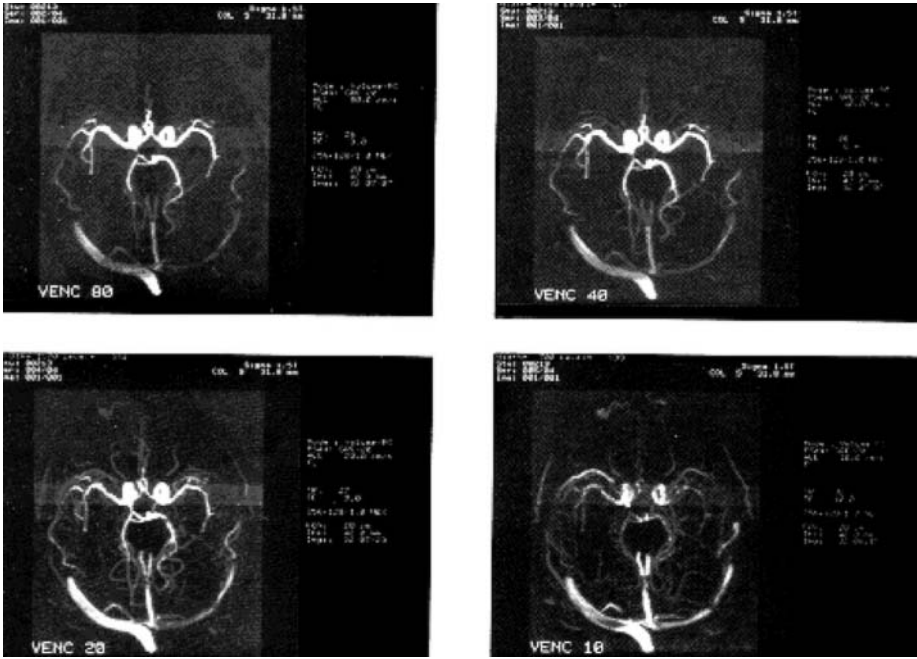


Figure 3.21: Three-Dimensional PC angiogram at multiple velocity encoding (VENC) shows the effect of high velocity encoding (cm/sec) at 80° (left panel), 40° (right panel) on top row and 20° (left panel), 10° (right panel) on bottom row to emphasize the better venous anatomical appearance with clear sphenoparietal sinus at low VENC.

3.2.5.1 Aliasing in Speed Images

When the velocity-encoding set below the peak velocities is encountered within the vessel lumen, the higher velocities will be aliased and appear as lower signal intensities from the lower velocities. Since the highest velocities are usually present at the center of the vessel, aliasing may result in a decrease in signal intensity within the center of the vessel. If a very low velocity encoding (VENC = 20 cm/sec) is used, the higher flow velocities will be aliased and the slower velocities will have greater signal intensity. The advantage of aliasing in magnitude and velocity images is also noticeable to bring out slower flow along the walls of arteries, structures, or to emphasize venous anatomy. VENC may be set lower than the peak velocity. Aliasing artifacts makes the flow information at the center of the artery meaningless but this part of the vessel is often not seen in the MIP projection images.

3.2.5.2 Aliasing in Phase Images

When peak velocity in a vessel is equal to the VENC value, the bipolar gradients give either a 180° or 180° phase shift, depending on the direction of flow. When velocity exceeds the VENC value and the phase shift exceeds 180° , it becomes indistinguishable from the phase shift produced by flow in the opposite direction. The result is phase aliasing. Here aliasing flow seems to change direction, since the $+190^\circ$ phase shift is equivalent to a -170° phase shift (see Fig. 3.22). For this reason, aliasing in individual flow-axis images is often recognized by adjacent white and black pixels. In addition, the measured phase shift increases with velocity up to a value of 180° , at which point it is aliased with an equal negative velocity. This sets a limit on the usable degree of flow encoding for quantitative

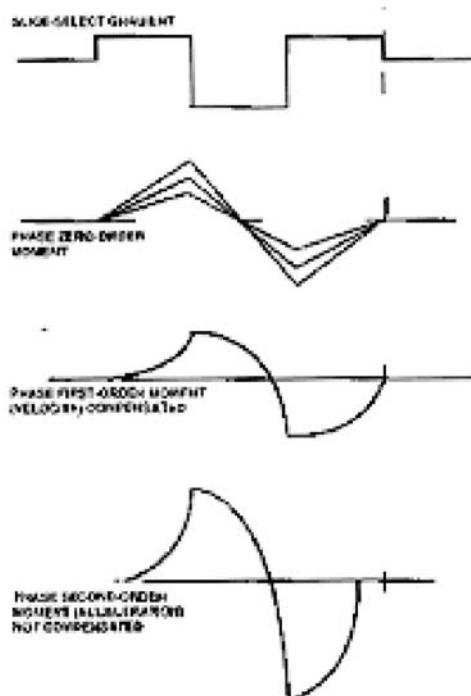


Figure 3.22: Phase plot shows the effect of a gradient on transverse magnetization at three different locations along the frequency axis. The gradient echo is formed by first dephasing the transverse magnetization along the frequency-encoding axis. The first half of the read-out gradient refocuses the magnetization, producing an echo at time TE.

studies. With higher velocity encoding, pulse is wrapped. Magnitude and speed images show a drop in signal intensity with increasing velocity.

For quantitative studies, one sets the flow encoding to produce a phase shift just below 180° for the highest velocities present. The quantitative relationship between velocity and phase shift reduces the detectability of small vessels and some aneurysms and reduces the apparent diameter of large vessels.

3.2.5.3 Phase Dispersion and Flow Compensation

Intravoxel spin-phase dispersion is called intravoxel incoherence or loss of spin-phase coherence. It imposes a limitation for vascular MRI. This loss of signal intensity can occur whenever any of the three conditions exists: (1) A wide spectrum of flow velocities exists within an imaging voxel; (2) higher orders of motion, such as acceleration and jerk, are not compensated; and (3) local variations in magnetic field homogeneity are present, such as those produced by magnetic susceptibility effects. In a long straight vessel with no bifurcation, blood flow is typically laminar flow. That is, the velocity profile across the vessel is not constant, but varies across the vessel lumen. The flow at the center of the lumen of the vessel is faster than that at the vessel wall, where resistance slows down the blood flow. As a result, the blood velocity is almost zero near the wall, and increases toward the center of the vessel. The velocity profile becomes more complicated when the flow is pulsatile and the vessel curves or bifurcates. In general, shear rate increases near the vessel wall, resulting in greater velocity variations, intravoxel phase dispersion, and loss of signal intensity. Decreasing the voxel size is one important strategy for minimizing intravoxel dephasing in vascular MRI studies. Smaller voxels encompass a smaller range of flow velocities. This reduced size of voxel also reduces SNR in a linear fashion. The loss of SNR can be offset by the use of long acquisition times. SNR is proportional to the square root of the imaging time. The other alternative is employing the stronger magnetic fields, as SNR is proportional to magnetic field strength. Thus, voxel-size reduction will improve nonturbulent flow only such as vascular structures with well-characterized distribution of velocities within a vessel. It will not eliminate signal loss due to true turbulence. The reason for this is that turbulence flow has randomly oriented the velocity vectors. The lower voxel-size strategy offers similar improvements in the regions with magnetic susceptibility

changes due to magnetic field gradients. Phase shift induced by flowing blood in the presence of a flow-encoding gradient is directly proportional to the velocity. A dispersion of velocities in a vessel, therefore, results in a dispersion of phase shifts. Consequently, a projection measurement of phase through a vessel with laminar flow will represent the average velocity provided that the flow-encoding gradient is not too strong. If the flow becomes complex or turbulent, the dispersion of velocity components along the projection may cause an attenuation of the signal, or even zero signal. Turbulent flow is the flows with different velocities that fluctuate randomly. The difference in velocities across the vessel changes erratically.

3.2.5.4 Flow Compensation

Spin echoes recover the loss of signal because of magnetic field inhomogeneity or susceptibility gradients. However, these spin echoes with longer echo times are less effective in overcoming the phase dispersion due to spins moving at different velocities. Flow compensation is a first-order gradient moment nulling. It employs the refocusing gradients to re-establish phase coherence. For this, lobes are added to the read-out and slice-select gradient waveforms. As a result, the loss of phase coherence due to different velocity distributions is minimized and velocity-induced phase shifts are canceled. This strategy results in an acquisition at constant velocity. However, high-order motions such as acceleration and jerks are compensated by the use of waveform complexity. As a result of additional lobes of gradient waveforms, the echo time and degrade image quality are increased.

3.2.5.5 First-Order Gradient Moment Nulling

It means that the system applies gradient pulses so that constant velocity spins and stationary spins have no net phase accumulation at each echo time. For stationary spins, the signs of the gradients are reversed so that the phase advance experienced at a given location is compensated by appropriate phase retardation. The first-order gradient moment nulling balances the phase for both stationary spins and spins moving with constant velocity. This can be accomplished with the application of a gradient sequence in which the strength and duration of the gradient pulses have a 1:2:1 ratio (see Fig. 3.22). Vascular blood

flow is pulsatile and velocity is not constant between excitation and detection. However, some phase dispersion will normally occur. In addition, in some anatomic regions the effects of acceleration become prominent and “acceleration drop out” signal loss becomes apparent in the resulting images. In peripheral vascular studies, pulsatile motion and jerk are significant causes of artifacts. Although acceleration compensation schemes exist, the inevitable trade-off of increased echo time can make them impractical.

3.2.5.6 Phase Dispersion

When magnetic field gradient is applied to a spin system, the spins within the voxel accumulate a phase angle in relation to one another. This phase angle difference is known as “phase dispersion.” To correct for this phase dispersion, the gradient is typically reversed to rephase the spins. This technique is used frequently in imaging sequences to refocus stationary spins. These “bipolar” gradient lobes are of equal strength and duration but have opposite signs (see Fig. 3.23). Spins that are moving in the direction of the magnetic field gradient are not refocused and are left with some residual phase. The motion-induced phase shifts occurring in the presence of magnetic field gradients are arithmetically defined by position/time derivatives called “moments.” The zeroth moment (M_0) describes the effect of a gradient on the phase of stationary spins. Similarly, the

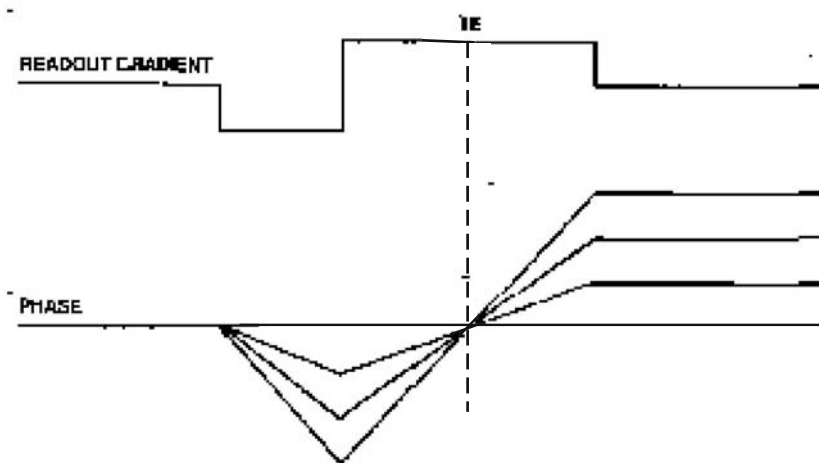


Figure 3.23: Gradient reversal.

first moment (M_1) describes its effect on the phase of a spin with constant velocity. The second moment (M_2) describes the gradient's relationship to the phase of spins experiencing acceleration. The third moment (M_3) defines the effect of jerk on spin phase. Even higher order moments exist, but they are usually less important.

3.2.5.7 Shorter Echo Times

Shorter echo times (TE) may also reduce the problem of signal loss due to phase dispersion. Short TE reduces the time for spins to dephase after the RF pulse. Short TE thereby reduces the signal loss arising from susceptibility gradients, velocity distributions, and higher orders of motion. For all VMRI techniques, flow-related phase errors accumulate as a function of $TE(n + 1)$, where n is the moment (i.e., $n = 1$ for velocity and $n = 2$ for acceleration). Phase error is, proportional to $TE(n + 1)$.

The effects of higher order moments become more significant for long echo delays. This is because the second moment (acceleration) has a cubic dependence on echo time, while the third moment (jerk) has fourth-power dependence. Using the shortest possible TE can therefore minimize signal loss due to these higher order moments. For example, a VMRI exam obtained with $TE = 3$ msec will have approximately one-half the velocity-related phase errors of the same study performed with $TE = 4$ msec.

3.2.5.8 Complex Flow

To minimize the problem of signal loss due to complex flow, several strategies may be employed. The dispersion of velocities along a projection can be greatly reduced by obtaining vessel images in thin cross-sections rather than in full projection. 3D data acquisition overcomes the problem of velocity dispersion within a voxel. Since the phase contrast technique relies on the phase shift induced in moving spins, conventional flow compensation techniques cannot be used on flow-encoding axis. To minimize phase dispersion, the bipolar phase-encoding gradient is placed symmetrically around the first moment (called PC flow compensation). However, a slightly shorter echo time can be achieved by placing this gradient asymmetrically in relation to the first moment. The resulting

technique may be called “minimum TE.” It produces the shortest possible TE with the PC sequence, and is selected by not choosing the flow compensation.

3.2.6 MRA Image Reformation

The MR data from MRA images is reformatted and this reformatting plays a major role in vascular anatomy observed in the MRA imaging. The common method for reformatting TOF-MRA uses the technique known as MIP. This technique also generates 3D images of blood vessels with blood motion. The other method for reformatting MRA images is shaded surface display. This method reformats image data in such a way that it appears as if a light is thrown onto structures to generate 3D appearance of vasculature.

3.2.6.1 Maximum Intensity Projection

The method of reformatting based on ‘maximum intensity projection’ is known as ‘mipping.’ The mipping of blood can be done based upon the blood flow characteristics. Flowing blood in MRA techniques has a high intensity. The intensity of a pixel in a slice is compared with that of the corresponding pixels in all the other slices (as in a channel), and the one with maximum intensity is selected. For example, pixel (1, 1) in slice 1 is compared with other pixels (1, 1) of all other slices. For this, an internal threshold is used, below which no pixel in the channel falls. This threshold process is repeated for all the pixels in the slice to connect high intensity dots in space in order to generate an MRA image. Thus MRA image represents the highest intensities (caused by flowing blood) in the imaging volume. A major drawback of this method is that bright structures other than blood may be included in the mipped image i.e. fat, posterior pituitary glands and subacute hemorrhage. This problem is observed only with TOF MRA and not with PC MRA. PC MRA is a subtraction technique based on velocity-induced phase shifts rather than on tissue T1 and T2 relaxation times.

3.2.6.2 Saturation Effects

The saturation effects can minimize the loss in signal intensity if small (15–20°) flip angles are used. The 3D phase contrast technique can image large volumes, such as the entire head, without serious signal loss due to saturation effects. As

a result of the reduced saturation dependency, short TR of 25–28 msec is used with the minimal saturation of moving spins. It allows the spins to recover from previous RF pulses. Longitudinal recovery occurs exponentially with a time constant T1. Normally, repetition time T_R five times the value of T1 is needed to ensure complete relaxation. However, long TR would limit the amount of data acquired and make 3D imaging difficult. Full relaxation can be achieved with a short pulse delay if smaller flip angles are used. Partially relaxed steady states are easily achieved with shorter pulse delays using the limited flip angles, the so-called gradient-recalled acquisition in the steady state for 2D or 3D volume acquisitions. Further reductions in saturation effects may also be realized by using intravenous contrast agents that shorten the T1 of blood. These refer to the gradual loss of longitudinal magnetization caused by repeated excitation RF pulses. This leads to loss of signal-to-noise ratio during 2D acquisition in which flowing blood has to travel within a slice or in a 3D acquisition in which the blood travels through a thick imaging volume (or slab). In such a situation, saturation effects may prevent the imaging of the distal portion of a vessel.

Contrast mechanism for vessel imaging is based on the differences in saturation between blood and stationary tissue, rather than flow itself. This contrast mechanism is usually dominated by “in flow” effects. TOF techniques differentiate blood only when its magnetization differs from that of surrounding stationary tissue. Longer blood stay in the imaging volume makes it more difficult to detect the vessels. Signal loss therefore occurs whenever slowly moving blood enters the volume of interest and reaches a new saturated steady state. Phase contrast angiography is less susceptible to this problem of signal loss. This may be due to saturation effects arising out of decreased TR and increased α factor.

Let us describe these factors.

3.2.6.2.1 Short TR. Short repetition times (TR) cause less recovery of longitudinal magnetization from one cycle to the next, causing gradual loss of the M_z component. This effect is less pronounced with longer repetition times.

3.2.6.2.2 Larger Flip Angle (α). A large flip angle causes more signal loss due to loss of longitudinal magnetization. Therefore, for a given TR, there is greater gradual loss of M_z with a larger flip angle (α) than with a smaller flip angle (see Fig. 3.24). In GRE, very short TR is selected, as a result saturation effects pose a problem. The uses of small flip angles counteract this effect. These saturation effects become especially important in 2D and 3D-plane flow or in

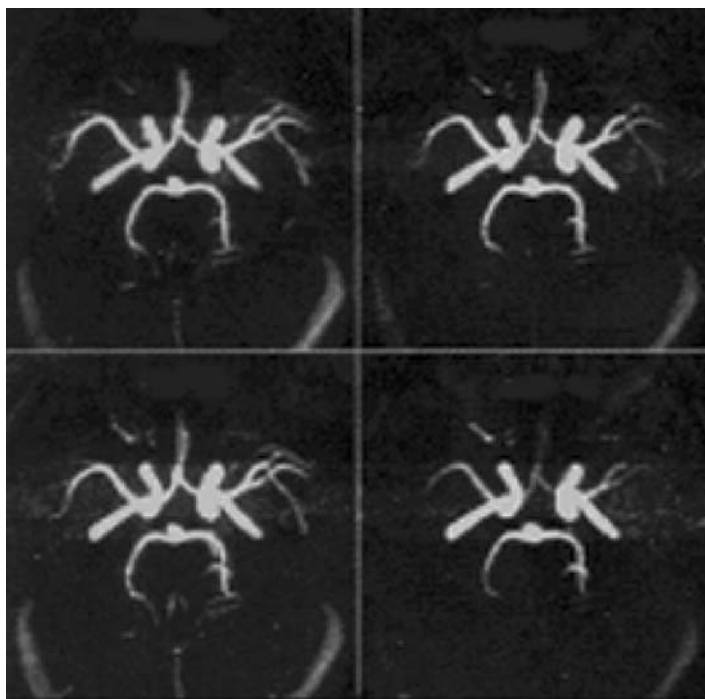


Figure 3.24: The figure represents the method of reformatting of TOF-MRA images by MIP. The technique generates the 3D images of blood vessels with blood motion. Larger flip angle at given TR show loss of magnetization and generate the different 3D appearance of vasculature (shown on right panel at bottom).

3D imaging in which volume imaging is performed over an imaging slab. Signal losses might be significant from one end of the slab to the other end of the slab.

3.2.7 Multislice GRE Techniques

These techniques use longer decrease in repetition time TR. As a result, the saturation effects lead to signal loss with the use of larger flip angles. This approach using longer decrease in TR improves the SNR. Other ways are also used to decrease saturation effects. Common use of paramagnetic contrast agent such as gadolinium chelate causes spin-lattice relaxation time (T1) shortening of blood (see Fig. 3.25). Consequently, the T1 recovery (from short T1 to normal T1 values) is faster with less saturation effects. In routine, multiple overlapping thin-slab acquisition (MOTSA) tilt optimized nonsaturated excitation (TONE) are also promising techniques to reduce saturation effects.

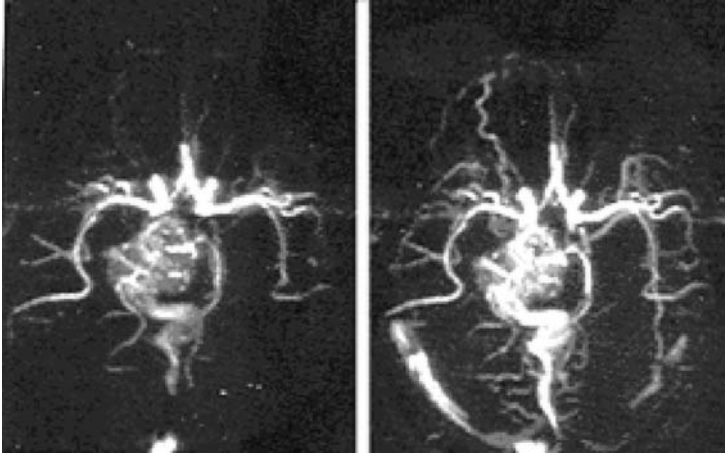


Figure 3.25: A method of multislice gradient echo is represented using short TR and larger flip angles with sufficient SNR (on left panel). However, gadolinium contrast agent shortened the blood with T1 recovery with less saturation effects (on right panel).

3.2.7.1 Multiple Overlapping Thin-Slab Acquisition (MOTSA)

It is a combination of 2D TOF and 3D TOF techniques for the purpose of reducing the saturation effects associated with a thick slab. In this method, multiple thin slabs used, which overlap by 25–50%. Extracting the central slices of each slab creates the final imaging volume and discards the peripheral slices, which are more affected by saturation effects. The main drawback of this technique is the appearance of “Venetian blind” artifact at the points where the slabs overlap.

3.2.7.2 Tilt Optimized Nonsaturated Excitation (TONE)

In this technique, flip angle (α) is increased progressively as the flowing spins move into the imaging volume by using increasing RF pulses. A large flip angle (α) yields higher SNR. Thus, larger flip angle counteracts the saturation effects of slow-flowing blood in deeper slices. This allows better visualization of distal vessels and the slow-flowing vessels. In common practice, ramped flip angle excitation pulse is used. In our commonly used scheme, the center flip angle is 30° and the flip angle at each end varies by 30%. As a result, flip angle changes 20° at the entry slice and 40° at the exit slice.

3.2.8 Magnetization Transfer

This method is based on suppression of the off-resonant protein-bound water protons. If magnetization transfer (MT) is combined with TOF MRA, it helps suppress the background signal at least by 30%. The best-known example is brain parenchyma where MT-TOF MRA increases conspicuity of small and distal branches of vessels with slow flow, and aneurysms. MT can be combined with TONE for further visualization of small vessels.

3.2.9 Flow Eddies

The flow eddies are unique for identification and estimation of stenosis although these cause overestimation of stenosis by MRA. Flow eddies are mainly contributed by turbulent flow and vortex flow as well as stream separation distal to stenosis and carotid siphons (vessel turns). This overestimation of stenosis is the result of accelerated flow through the stenotic area leading to dephasing and flow void during TE. Common examples of overestimation are estimation of length of stenosis in the case of poststenosis or mimicking stenosis in the case of vessel turns.

3.2.10 Bright Blood Imaging

In addition to making the vessels appear black, vascular structures can also be visualized by making them brighter. Several techniques can be used to enhance the signal from flowing blood including gradient echo imaging and/or gradient moment rephasing and/or contrast enhancement. In gradient echo imaging, the flowing spins are refocused by the rephasing gradient. The patent vessels appear brighter on the images. This technique can be referred to as “bright blood imaging” and can be further improved by the application of an imaging option known as “gradient moment rephasing.” It is a first-order velocity compensation technique to visualize slow moving protons with constant velocity (see Fig. 3.26). Protons in venous blood or CSF are put into phase with the stationary protons. So, the intravoxel dephasing is reduced. Gradient moment rephasing compensates the flow by making these vessels containing slow flowing spins appear brighter. It enhances the signal from blood and CSF.

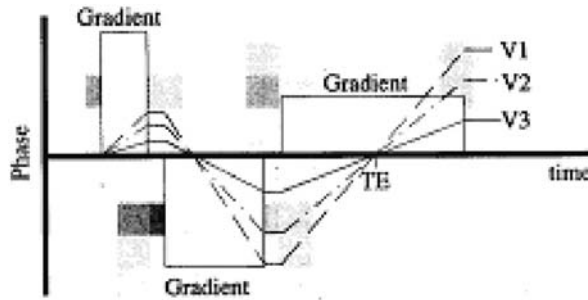


Figure 3.26: V1, V2, and V3 effect.

3.2.11 Black Blood MRA

Black blood MRA is another technique for MRA in which flowing blood appears dark rather than bright. It appears as negative of bright blood MRA. Rapidly flowing blood in arteries exhibits the TOF MRA signal losses. Slow flowing blood in veins appears as higher signal intensity. Various flow presaturation pulses and dephasing methods via gradients are employed in this technique to render flowing blood as black. This technique uses the MIP algorithm. Black blood MRA has several advantages. They offer no overestimation for the degree of stenosis and no dephasing in vessel turns that mimic stenosis. On the other hand, the technique has disadvantages that calcified plaque appears dark. Thus, this technique may underestimate the degree of stenosis or invisible plaques. Other black materials such as air or bone may mimic the blood flow.

Black blood MR angiograms make use of another time-of-flight phenomenon—the signal void observed for flowing blood in spin echo images. Unlike white blood, or INFLOW angiograms, which use a gradient echo sequence to enhance flowing blood and saturate static tissue, black blood angiography uses a spin echo sequence with presaturation to increase the signal of the tissue and to create a signal void (i.e. no MR signal) for flowing blood. The data is then processed using an MIP algorithm to yield the final MR angiogram. Black blood magnetic resonance angiography offers the advantage that signal voids due to turbulent flow are avoided. However, the contrast between vessel and static tissue may be lower, arterial and venous flow cannot be easily distinguished, and several regions of signal void such as nasal sinuses exist on images. Despite these disadvantages, black blood MRA may prove useful in the determination of some pathologies, such as severe stenotic lesions.

3.3 Acquisition Methods

This section describes the basic theory of MR angiography, mainly the INFLOW, FLAG, and rapid sequential excitation (RSE) methods.

3.3.1 INFLOW Method—Time of Flight

This method belongs to a class of MR angiographic techniques known as “time-of-flight.” This technique gives rise to 3D information about the vessels in the volume of tissue being imaged with high contrast between the stationary tissue and the flowing blood. The INFLOW method relies on the flow related image enhancement caused by the movement of fresh, unsaturated blood into an already saturated slab of tissue. The INFLOW method has a number of advantages over other angiographic imaging methods. First, image subtraction is not necessary, thereby reducing scan time and computing requirements while speeding data manipulation. Second, high contrast can be obtained virtually independent of flow velocity. Third, the arteries or veins may be selectively imaged by the use of presaturation slabs. Finally, the technique does not require the use of self-shielded gradients. It is less sensitive to motion than the phase contrast methods. Using the INFLOW technique, angiograms may be obtained in only 10–15 min. For example, the data can be processed by sending a batch job or processed interactively with AP500 within 10 min. The choices are available on selecting INFLOW processing under the ANPROC key. Both batch and interactive processing are discussed later in this section.

To achieve the best possible contrast in the final images, the imaging parameters must provide for maximum refreshment of blood in the imaging volume. The threshold minimum velocity (V_t) is given by:

$$V_t = d/TR, \quad (3.16)$$

where d is the slice thickness and TR is the repetition time. For a typical 2D INFLOW sequence with 2 mm slices and TR = 50 ms, threshold velocity (V_t) will be 0.04 m/sec. For velocities greater than V_t , the signal intensity is essentially independent of the flow velocity. Typical velocities range from 1 m/sec for the aorta and 0.8 m/sec in the carotid artery and 0.03 m/sec in small veins.

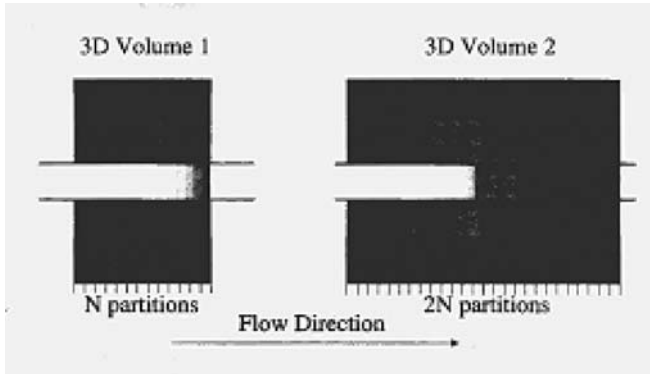


Figure 3.27: Partition effect.

The phase of the transverse magnetization is made independent of the flow velocity by the use of velocity compensated gradients. However, higher order flow terms may cause signal void in the areas of turbulent flow. The use of short echo times compensates for this. If the stationary tissue is selected as the volume of interest, it may be saturated using a short TR and a large tip angle (see Fig. 3.27). During this pulse sequence, fresh unsaturated blood moves into the imaging slice. This results in good contrast between the unsaturated blood and the stationary tissue.

The INFLOW technique may be used with 2D multiple single slice or 3D acquisition with a flow compensated gradient echo sequence. For 2D multiple-single-slice INFLOW, many thin (2–3 mm) contiguous (or over contiguous) slices are collected in a plane that is orthogonal to the blood flow. The optimum contrast between flowing blood and stationary tissue should be obtained with the shortest TE, a TR of the order of 40–60 msec, and a large tip angle of 45–90°, depending on the anatomy being studied and the flow rate of blood. Presaturation of a slab above or below the imaging slice allows selective imaging of the veins or arteries. The single sided, parallel presaturation slab moves with the slice position, ensuring good suppression. The slab thickness is adjusted in the second pass parameters and is typically set to 50 mm. An alternative method for certain imaging protocols employs a presaturation plane that is perpendicular to the imaging slice. An example is the use of a sagittal or coronal slice for imaging the carotid arteries. A perpendicular presaturation slab is necessary to remove the venous flow.

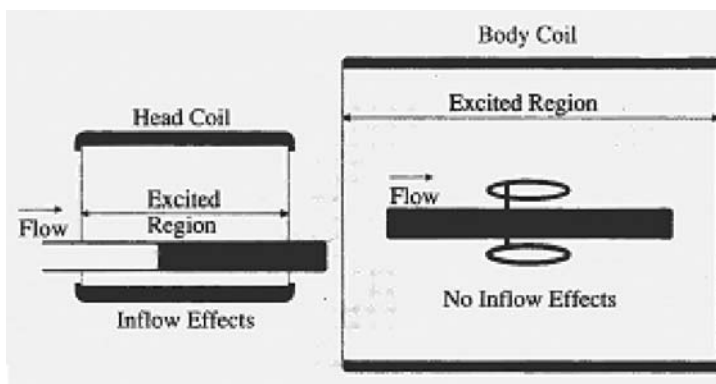


Figure 3.28: Excitation in coil.

Since the 2D method is a multiple-single-slice technique, the slices are reconstructed as they are collected. They may be viewed while subsequent slices are being collected. This feature allows the operator to monitor the data collection. Later, data collection may be stopped to correct the protocol, if necessary, without waiting for all the data to be collected. It also shortens the study time by reconstructing the slices while the acquisition is still in progress. In our experience for routine transverse slices of the carotids, processing methods consist of projecting the stack of slices in a plane orthogonal axis. Top-down projections or perpendicular projections in the AP direction may be generated with appropriate selection of projections in the select procedure menu (see Fig. 3.28). First projection will be generated when slices are reconstructed. INFLOW image processing uses a maximum intensity projection with the interpolation between the slices. The maximum intensity voxel in a given vector is used for that projection view.

3.3.2 FLAG, RSE-Phase Contrast

Flow adjusted gradient (FLAG) and RSE are fast field echo sequences. They have velocity-sensitive gradients that are designed to image flow by adjusting their sensitivities to different flow velocities. The contrast between flowing and stationary tissue is based on the phase of the transverse magnetization of moving spins rather than on time-of-flight effects. Spins moving in the presence of a magnetic gradient accumulate a flow-induced phase shift. This phase shift depends on the strength and duration of the gradient and the velocity of the moving spins.

In our experience, two or more images are collected, one of which is velocity sensitive in a specified direction and the other is velocity compensated image. For example, a thick slab (100 mm) FLAG sequence in the plane of flow (thick coronal slab for imaging the abdominal aorta) will yield a projective MR angiographic image. The resultant phase images are phase corrected and subtracted to yield the projection angiogram. The FLAG sequence can be run with or without cardiac triggering. In the noncardiac triggered version, a “shortest” TR is not recommended. The FLAG sequence interleaves the velocity compensated and velocity-sensitive data in consecutive TR periods. If the scan is gated, FLAG sequence interleaves the velocity sensitive data such as in consecutive heartbeats. For better suppression of respiratory motion, the RSE sequence may be used. RSE interleaves the velocity compensated and velocity-sensitive data in the same heartbeat. The RSE sequence must be run in the cardiac triggered mode.

3.3.3 Digital Subtraction MRA

Digital subtraction MRA has been compared to digital subtraction angiography (DSA) as contrast is selectively produced for moving spins during two acquisitions. These moving spins are then subtracted to remove the signal from the stationary spins, leaving behind an image of the moving spins. An early subtraction angiogram may be performed while gating to the cardiac cycle. An acquisition during systole (fast flow) is generally subtracted from an acquisition during diastole (slow flow). In this case, the stationary spins were subtracted, retaining only the moving spins, such as the vasculature, on the resultant image. This technique is significant as recent techniques were based on same principles.

3.4 Recent Advancement in MRA Techniques

Different newer methods are reported in the literature for MRA from the perspective of different applications applied for flow imaging.

3.4.1 Sensitivity Encoding

Sensitivity encoding (SENSE) is used to increase spatial resolution and decrease venous contamination in peripheral MRA. In this method, single-bolus peripheral

contrast-enhanced (CE)-MRA was performed [1]. Manual table movements combined with SENSE in the upper station allowed for more rapid overall scan coverage such that acquisition of the lower station began 34 sec after aortic contrast arrival. True submillimeter isotropic resolution was achieved in the lower station. Diagnostic MR angiograms of all three stations were obtained. Venous enhancement did not confound interpretation in any case. Submillimeter lower station resolution provided excellent vascular details. Decreased delay time between upper and lower station acquisition in single-bolus peripheral MR angiograms, now possible using parallel imaging techniques, combined with lower station submillimeter resolution, may decrease venous contamination and increase overall interpretability, thus increasing clinical acceptance of peripheral MRA.

3.4.2 Blood Pool Contrast Enhancement

This technique of blood pool contrast-enhanced MRA was used to visualize the arterial and venous vessel tree and to detect deep venous thrombosis of the lower extremities. Patients with pulmonary embolism were randomized to evaluate various doses of NC100150 by T1-weighted (T1W) 3D gradient recalled echo sequence. Qualitative assessment of overall MRA image quality and semi-quantitative vessel scoring revealed good to excellent delineation of venous and arterial vessel segments independent of the dose of NC100150. However, quantitative region of interest analysis revealed a significantly higher signal-to-noise ratio in the high-dose group than in the mid- and low-dose groups of NC100150. Between dose groups, the SNR was independent of vessel type (artery or vein) and vessel segment localization (proximal or distal). Venous thrombi were characterized by very low signal intensity, approximately one tenth the SI in adjacent venous segments. High-quality MR angiograms of the lower extremities can be obtained using low concentrations of NC100150 in combination with a strong T1W 3D GRE sequence. The obvious delineation of venous thrombi suggests that this technique may be potentially used as a noninvasive “one-stop shopping” tool in the evaluation of thrombo-embolic disease [2].

3.4.3 Digital Subtraction Angiography

Contrast-magnetic resonance angiography (CE-MRA) MoBI-trak was used in the evaluation of the peripheral vessels in patients with peripheral vascular

disease [3]. CE-MRA with automated table movement (MoBI-trak) using a 1.5 T superconducting magnet (Philips Gyroscan ACS NT) was equipped with a Power Trak 6000 gradient. Contrast medium (Gd-DTPA) was administered in two sequential boluses-20 cm³ at 0.6 cm³/sec (starting phase) and 20 cm³ at 0.3 cm³/sec (maintenance phase), using a MedRad Spectris automatic injector. DSA was the gold standard and was performed using a Philips Integris 3000, with a brilliance intensifier of 38 sec. DSA and MRA were evaluated on printed films. DSA provided more than 200 diagnostic assessments including stenosis <50–99% occlusions. CE-MRA MoBI-trak exhibited good sensitivity, specificity, positive and negative predictive values and high diagnostic accuracy. Using this technique MoBI-trak has been shown to be a reliable technique for the detection of peripheral vascular disease up to the trifurcation, although it underlines the necessity for more diagnostic investigation and improvements in the technique.

3.4.4 Magnetic Resonance Angiography with Diffusion-Weighted Imaging

This approach was used for intraoperative magnetic resonance imaging, including magnetic resonance angiography and diffusion-weighted imaging [4]. This integrated approach was used to monitor the surgical treatment of a patient with an intracranial aneurysm. Intraoperative imaging was performed with a ceiling-mounted, mobile, 1.5-T magnet (developed in collaboration with Innovative Magnetic Resonance Imaging Systems, Inc., Winnipeg, MB, Canada) that included high-performance 20-mT/m gradients. Pre- and postclipping, intraoperative, T1-weighted, angiographic, and diffusion-weighted magnetic resonance images were obtained from a patient with an incidental, 8-mm, anterior communicating artery aneurysm. T1-weighted images demonstrated brain anatomic features, with visible shifts induced by surgery. Magnetic resonance angiography demonstrated the aneurysm and indicated that, after clipping, the A1 and A2 anterior cerebral artery branches were patent. Diffusion-weighted studies demonstrated no evidence of brain ischemia. For the first time, intraoperative magnetic resonance imaging was reported to monitor the surgical treatment of a patient with an intracranial aneurysm (see Fig. 3.29).

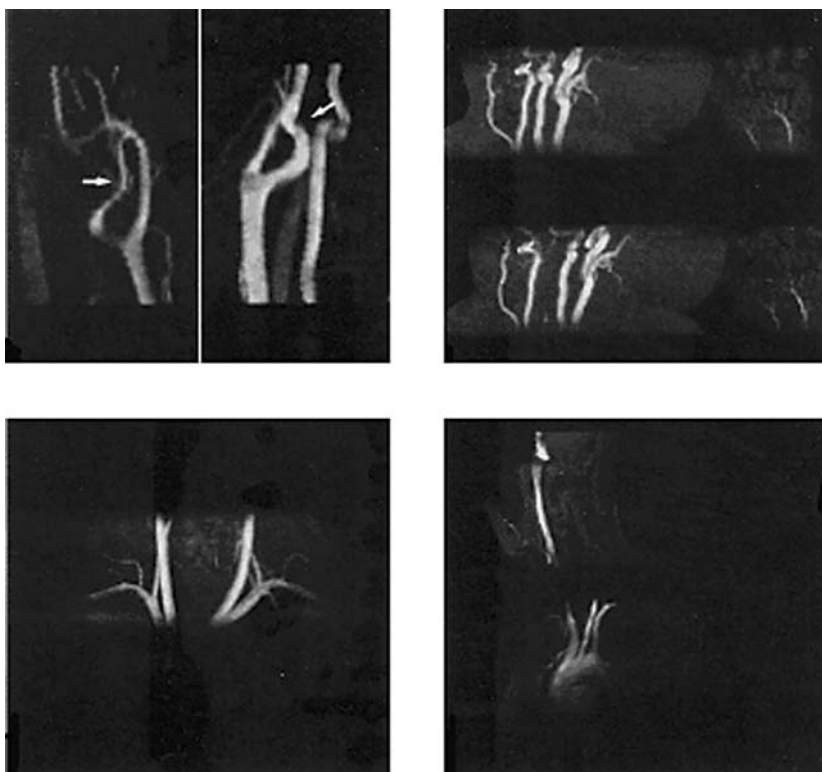


Figure 3.29: Magnetic resonance angiography is shown to demonstrate the aneurysm with both A1 and A2 anterior cerebral artery branches as patent (shown with arrows in left panel on top). Pre- and post-clipping, intraoperative, T1-weighted, angiographic and diffusion-weighted magnetic resonance images were obtained from a patient with an incidental, 8-mm, anterior communicating artery aneurysm. T1-weighted images showed brain anatomic features with visible shifts induced by surgery.

3.4.5 3D-Navigator Echo MRA

A three-dimensional navigator echo (NE) sequence on an MR scanner with a high performance gradient system was used to evaluate MR coronary angiography [5]. For imaging, a navigated gradient-echo (GE) pulse sequence with an in-plane resolution between 0.63×0.63 and $0.78 \times 0.78 \text{ mm}^2$ with two overlapping slabs was acquired. The number of visualized coronary artery segments was estimated. In addition, signal-to-noise measurements were performed in the ascending aorta at the level of the proximal right and left coronary arteries.

This technique visualized the left main, the right coronary artery up to segment 3, the LAD up to segment 8, and the RCX with segments 11 and 13. The average signal-to-noise value at the level of the right coronary artery was 11.4 ± 5.0 , at the level of the left coronary artery 12.3 ± 4.5 with an in-place resolution of $0.63 \times 0.63 \text{ mm}^2$. This resulted in a too low signal-to-noise ratio so that an adequate assessment of coronary arteries was no longer possible. 3D-MR coronary angiography using the navigator technique is limited by the low signal-to-noise ratio.

3.4.6 Navigator Echo and Cine Gradient-Echo Technique

This technique was used to evaluate coronary artery stents with MR [6]. For both sequences the low-signal artifact was used to localize the stent, whereas the flow-related high signal before and distal to the stent was considered as a potency sign. All the stents were recognized as signal void with GE, and all but one with NE. Positive EET, with a stent on the left anterior descending coronary artery, presented low signal distal to the stent at both MR sequences. These suggested the dysfunction stenosis at conventional coronary angiography (CCA). Two sequential stents on the right coronary artery presented lack of signal distal to the stents at both MR sequences. It suggested occlusion (97% stenosis at CCA). However, negative EET, MR high signal before and distal to the stent suggested patency at both sequences. MR seems to be a safe and promising technique for noninvasive evaluation of coronary stents.

3.4.7 MR Phase-Contrast Doppler Flow Quantification

Determination of blood flow volume is useful in assessing ischemic cerebrovascular disease. Blood flow volume measurement was evaluated by three noninvasive imaging techniques, namely color velocity imaging quantification, spectral Doppler imaging quantification, and MR phase-contrast flow quantification, to see how well the flow values determined by each technique agreed with one another [7]. Flow volume quantification was tested experimentally using a flow simulator and three techniques. These techniques evaluated the vertebral and internal carotid arteries of patients with history of cerebral ischemia. In the flow simulation study, the flow values in each technique were compared with the phantom flow using Wilcoxon's signed rank test. Flow volumes were measured by color velocity imaging quantification. MR phase-contrast flow quantification

agreed with the phantom flow simulation within the tested range, and spectral Doppler imaging quantification values were significantly overestimated. In patients, a large variation in the blood flow volume was obtained between each technique. Blood flow volume measurements determined by the three noninvasive imaging techniques on the same vessel can differ widely, and spectral Doppler imaging quantification consistently overestimated the flow volume. Color velocity imaging quantification or MR phase-contrast flow quantification can be used for clinical follow-up investigations in the future.

3.4.8 Breath-Hold Contrast Enhanced MRA

The purpose of this technique was to evaluate the effect of breathing on image quality of the aortic arch and carotid vessels during contrast-enhanced MR angiography. It showed that high-resolution breath-hold contrast-enhanced MR angiography combined with a timing-bolus technique can produce high-quality images of the entire carotid circulation [8]. High-resolution contrast-enhanced MR angiography on a 1.5-T Siemens imager was used for coronal three-dimensional gradient-echo sequence (TR/TE, 4.36/1.64; flip angle, 25°) with asymmetric k-space acquisition. The 136×512 matrix yielded voxel sizes of $1.33 \times 0.64 \times 1.0$ mm³. A timing-bolus acquisition, orientated in the coronal plane to include the aortic arch, was obtained initially during free-breathing. Twenty milliliters of gadopentate dimeglumine was injected at 2 mL/sec. Unenhanced and enhanced 3D volumes were recorded. A subtracted 3D set was calculated and subjected to a maximum-intensity-projection algorithm. Half of the patients held their breath during angiography and the other half did not. Aortic arch motion was measured on the timing-bolus acquisition as the distance moved by a single pixel in both the x and y directions. Two observers assessed MIP MR images independently and vessel sharpness was scored on a scale of 1–5. Sharpness was also assessed quantitatively by generating a signal intensity profile across the aortic arch vessel wall and calculating the average of the upward slope and downward slope at full-width half maximum. Visualization of carotid branch vessels was scored on a scale of 0–5, and venous contamination was scored on a scale of 0–3. Average in-plane aortic arch movement was 10.3 mm in the x direction and 8.7 mm in the y direction. Quantitative and qualitative sharpness of the aortic arch and great vessel origins was better ($p < 0.05$) during breath-holding than during non-breath-holding. No difference in the sharpness of the carotid vessels was noted

between the two groups. Carotid branch vessels were well visualized from the aortic arch to the intracerebral circulation. The average venous contamination score was 0.56. Breath-holding greatly improves the sharpness of the aortic arch and great vessel origins but has no effect on visualization of the carotid vessels. High-resolution breath-hold contrast-enhanced MRA can produce high-quality, artifact-free images of the entire carotid circulation from the aortic arch to the intracerebral circulation.

3.4.9 K-space 3D Navigator-Gated MRA

To acquire the center of k-space while extending three-dimensional free-breathing navigator-gated coronary MRA by an initial single breath-hold [9], this approach was successfully applied. Resulting images were compared with conventionally acquired free-breathing navigator-gated MR angiograms. The acquisition of k-space center during the single breath-hold resulted in an appreciable increase in the signal-to-noise ratio. Visible length of the right coronary artery, as well as contrast-to-noise ratio between blood and the myocardial muscle were identical. The breath-hold extension was shown to be a valuable technique that may be combined with first-pass contrast-enhanced MR imaging. The other advantage was the creation of photographic freeze scans of coronary small arteries and heart within each heartbeat of approximately 80 msec intervals.

3.4.10 MR Arterial Spin Tagging Projection Coronary MRA Technique

Conventional coronary MRA techniques display the coronary blood-pool along with the surrounding structures, including myocardium, the ventricular and atrial blood-pool, and the great vessels [10]. This representation of the coronary lumen is not directly analogous to the information provided by X-ray coronary angiography, in which the coronary lumen displayed by iodinated contrast agent is seen. Analogous “luminographic” data may be obtained using MR arterial spin tagging (projection coronary MRA) techniques. Such an approach was implemented using a 2D selective “pencil” excitation for aortic spin tagging in concert with a 3D interleaved segmented spiral imaging sequence with free-breathing and real-time navigator technology. This technique allows for selective 3D

visualization of the coronary lumen blood-pool, while signal from the surrounding structures is suppressed. However, there are technical hurdles for visualizing the coronal arteries because of small size and near-constant mobility, which is related to cardiac cycle and normal breathing.

3.4.11 Collateral Flow Assessment with Contrast Enhanced MR Velocity Mapping

To correlate quantification of collateral flow in aortic coarctation with the morphological visualization of the collateral vessels and to compare different approaches to the measurement of collateral flow [11], coarctation was examined with T1-weighted spin-echo (T1-W SE) imaging and 3D contrast-enhanced MRA. MR velocity mapping was performed at four levels in the descending aorta. The flow immediately above and below the coarctation did not differ significantly. Measuring within the coarctation resulted in flow overestimation. The increase in flow from proximal to distal aorta was $12 \pm 21\%$ in patients with no or uncertain collaterals and $69 \pm 55\%$ in patients with pronounced collaterals. Spin-echo images and MRA were comparable in visualizing collateral vessels. The visual estimation of collaterals was correlated reasonably well with flow quantification MR velocity mapping. Collateral flow assessment with MR velocity mapping was an accurate technique for evaluating the hemodynamic importance of a coarctation and was recommended if abundant collaterals are not visualized with spin echo or MRA.

3.4.12 (Gd)-Enhanced Three-Dimensional Magnetic Resonance Angiography

The goal of this technique was to evaluate the diagnostic value of gadolinium (Gd)-enhanced three-dimensional MRA in patients with congenital and acquired anomalies of the pulmonary and systemic veins [12]. Gadolinium-enhanced 3D MRA is a fast magnetic resonance imaging technique that has shown great promise in the evaluation of large and medium-sized arteries. However, its application to venous anomalies has not been established. Gd-enhanced 3D MRA examination was used for diagnosis with anomalies of the pulmonary or

systemic veins and had additional diagnostic data available for comparison with the MRA findings. The technique did not detect anomalies of the pulmonary veins that were subsequently diagnosed by MRA. Intervention catheterization procedures and operations followed the 3D MRA diagnoses. 3D MRA either diagnosed previously unsuspected venous anomalies or added new clinically important information. The mechanism of pulmonary vein compression in eight patients was determined by MRA but not by other imaging modalities. Gadolinium-enhanced 3D MRA is rapid and accurate. 3D MRA has been shown to be the premier noninvasive technique for imaging large blood vessels in the body.

3.4.13 3D Time-Resolved Imaging of Contrast Kinetics Technique

This technique based on contrast-enhanced MR angiography was used by direct comparison with the fluoroscopic triggered 3D-elliptical centric view ordering (3D-ELLIP) technique [13]. 3D-TRICKS and 3D-ELLIP were directly compared on a 1.5-T MR unit using the same spatial resolution and matrix. In 3D-TRICKS, the central part of the k-space is updated more frequently than the peripheral part of the k-space, which is divided in the slice-encoding direction. The carotid arteries were imaged using 3D-TRICKS and 3D-ELLIP sequentially in 14 patients. Temporal resolution was 12 sec for 3D-ELLIP and 6 sec for 3D-TRICKS. The signal-to-noise ratio (S/N) of the common carotid artery was measured and the quality of MIP images was then scored in terms of venous overlap and blurring of vessel contours. No significant difference in mean S/N was seen between the two methods. Significant venous overlap was not seen in any of the patients examined. Moderate blurring of vessel contours was noted on 3D-TRICKS in five patients and on 3D-ELLIP in four patients. Blurring in the slice-encoding direction was slightly more pronounced in 3D-TRICKS. However, qualitative analysis scores showed no significant differences. When the spatial resolution of the two methods was identical, the performance of 3D-TRICKS was found to be comparable in static visualization of the carotid arteries with 3D-ELLIP, although blurring in the slice-encoding direction was slightly more pronounced in 3D-TRICKS. 3D-TRICKS is a more robust technique than 3D-ELLIP, because 3D-ELLIP requires operator-dependent fluoroscopic triggering. Furthermore, 3D-TRICKS can achieve higher temporal resolution. For the

spatial resolution employed in this study, 3D-TRICKS may be the method of choice.

3.4.14 Autocorrected MRA for Motion-Induced Artifacts

This technique was used to investigate the efficacy of a retrospective adaptive motion correction technique known as autocorrection for reducing motion-induced artifacts in high-resolution three-dimensional time-of-flight MR angiography of the circle of Willis [14]. Gradient-recalled echo three-dimensional time-of-flight MR angiography sequence was used for MRA of the circle of Willis. Each volunteer was asked to rotate approximately 2° after completion of one third and one half of the acquisition in the axial, sagittal, and oblique planes (45° to the axial and sagittal planes). A single static data set was also acquired for each volunteer. Unprocessed and autocorrected maximum-intensity-projection images were reviewed as blinded image pairs by six radiologists and were compared on a five-point image quality scale. Mean improvement in image quality after autocorrection was 1.4 ($p < 0.0001$), 1.1 ($p < 0.0001$), and 0.2 ($p = 0.003$) observer points (maximum value 2.0), respectively, for examinations corrupted by motion in the axial, oblique, and sagittal planes. All three axes had statistically significant improvement in image quality compared with the uncorrected images. Autocorrection can reduce artifacts in motion-corrupted MR angiography of the circle of Willis without distorting motion-free examinations.

3.4.15 Multiphase Contrast-Enhanced Magnetic Resonance Angiography

A fast pulse sequence with spiral in-plane readout and conventional 3D partition encoding was reported for multiphase contrast-enhanced magnetic resonance angiography (CE-MRA) of the renal vasculature and compared to a standard multiphase 3D CE-MRA with FLASH readout [15]. An isotropic in-plane spatial resolution of $1.4 \times 1.4 \text{ mm}^2$ over $2.0 \times 1.4 \text{ mm}^2$ could be achieved with a high temporal resolution. The theoretical gain of spatial resolution by using the spiral pulse sequence and the performance in the presence of turbulent flow was evaluated in phantom measurements. A deblurring technique corrected the spiral raw data. Thereby, the off-resonance frequencies were determined by

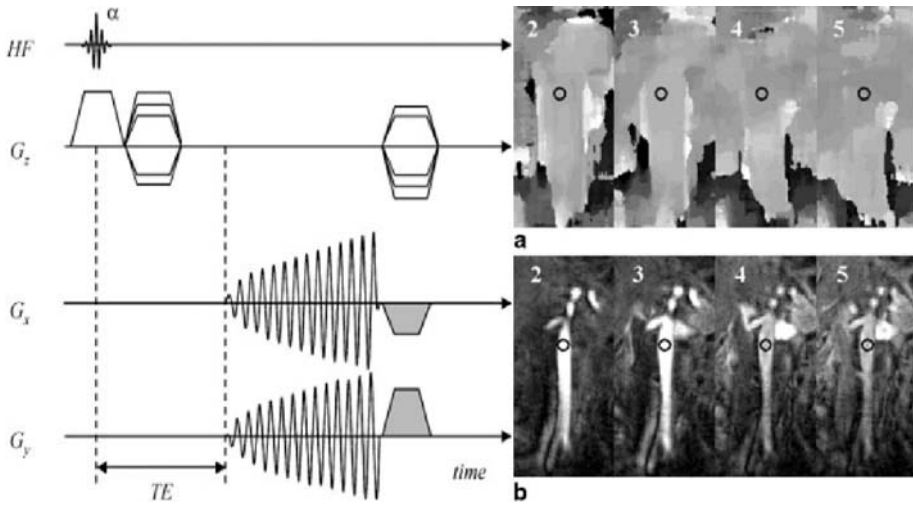


Figure 3.30: FFEP MRA.

minimizing the imaginary part of the data in image space. The correction algorithm reduced image blurring substantially in all MRA phases (see Fig. 3.30). The image quality of the spiral CE-MRA pulse sequence was comparable to that of the FLASH CE-MRA with increased spatial resolution and a reduced contrast-to-noise ratio. Additionally, artifacts specific to spiral MRI could be observed that had no impact on the assessment of the renal arteries.

3.4.16 High-Resolution MRA with Phase/Frequency Flow Compensation

A newly developed pulse sequence 3D TOF-VTE was tested on clinical MRI systems, by performing scans of the cervical carotid artery and intracranial carotid artery at the carotid siphon. It required very long echo delay times (TE). Variable TE (VTE) was implemented into flow-compensated 3D TOF to minimize the effective TE and reduce the flow-related signal void. The k-space of the 3D TOF was divided into segment groups ranging from two to 32 segments with different TE. The TE were minimized and the flow-compensation gradient lobes were calculated to null the total first moment at the peak of the echo for each segment [16]. Possible artifacts and off-resonance effects were evaluated, with respect to the number of TE segments, using the point spread function

(PSF) and corresponding experiments. The optimal number of TE segments for the least artifact was determined to be one-half of the number of slices. Two types of artifacts caused by VTE were predicted and subsequently observed. The signal distribution near the bifurcation and the siphon was much more uniform with VTE, and the flow-related signal loss was greatly reduced (see Fig. 3.30). The resultant MR angiograms provided improved vessel detail. The results show that VTE improved the quality of flow-compensated 3D TOF MRA.

3.4.17 Cardiac-Triggered Free-Breathing 3D Balanced Fast Field-Echo Projection MRA

A two-dimensional pencil-beam aortic labeling pulse was developed for the renal arteries [17]. For data acquisition during free breathing in eight healthy adults and seven consecutive patients with renal artery disease, real-time navigator technology was implemented. This technique allows high spatial resolution and high contrast renal MR angiography and visualization of renal artery stenosis without exogenous contrast agent or breath hold (see Fig. 3.31). Initial promising results warrant larger clinical studies.

3.4.18 Cervical MRA

Initial experience with intracranial and cervical MRA at 3.0 T was reported. Phantom measurements (corrected for relaxation effects) show $S/N(3.0T) = 2.14 + / - 0.08 \times S/N(1.5T)$ in identical-geometry head coils [18]. A 3.0 T TOF intracranial imaging protocol with higher-order autoshimming was developed and compared to 1.5 T 3D TOF in 12 patients with aneurysms. A comparison by two radiologists showed the 3.0 T to be significantly better ($P < 0.001$) for visualization of the aneurysms (see Fig. 3.29). The feasibility of cervical and intracranial contrast enhanced MR angiography (CEMRA) at 3.0 T was also examined. The relaxivity of the gadolinium contrast agent decreased by only about 4–7% when the field strength was increased from 1.5 T to 3.0 T. Cervical 3.0 T CEMRA was obtained in eight patients available for direct comparison. Image comparison suggested 3.0 T to be favorable field strength for cervical CEMRA. Voxel volumes of 0.62–0.73 mm³ were readily achieved at 3.0 T with the use of single-channel transmit-receive head or cervical coil, a 25 mL

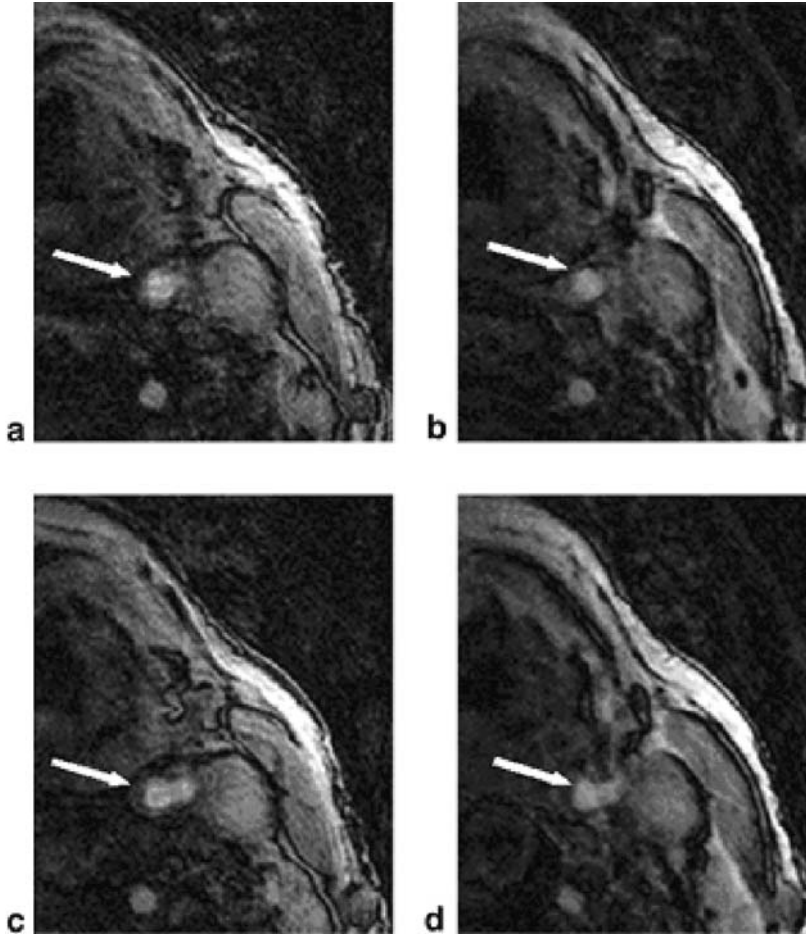


Figure 3.31: Source images of the 3D TOF MRA of the left carotid artery of a volunteer: (a and b) inferior and (c and d) at the carotid bifurcation. Images were acquired (a and c) without VTE and (b and d) with VTE (16 TE segments). The imaging parameters were as follows: matrix = $256 \text{ Å} \sim 256 \text{ Å} \sim 32$, slice thickness = 1 mm, TR = 24 ms, FOV = 14 cm, and TE = 1.8/2.9 ms for VTE on/off. MT was not applied. The reduced signal indicated by arrows in a and c was much more uniform in images b and d with VTE.

bolus of Gadoteridol, and a 3D pulse sequence with a 66% sampling efficiency. This spatial resolution allowed visualization of intracranial aneurysms, carotid dissections, and atherosclerotic disease including ulcerations. Potential drawbacks of 3.0 T MRA are increased SAR and $T(^*)^2$ dephasing compared to 1.5 T.

The dependence of RF power deposition on TR for CEMRA was calculated and described.

3.4.19 Magnetization Transfer MRA with RF Labeling Technique

A method for MT angiography using an RF labeling technique was suggested. The method utilized a slice-selective spin-lock pulse sequence for tagging the spins of inflowing blood [19]. The pulse sequence begins with a spatially selective 90° (x) RF pulses, followed by a nonselective composite locking pulse of 135° (y) – $n[360^\circ$ (y)] – 135° (y) and by a 90° (–x) pulse. A spoiler gradient was then applied. A rapid imaging stage, which yielded a T1 rho-weighted signal from the tagged spins, completed the sequence. Untagged spins were thoroughly dephased and consequently suppressed in the image. Thus, contrast was obtained without an injection of a contrast material or image subtraction. Furthermore, the flow of the tagged bolus could be visualized. The sequence was implemented on phantoms and on human volunteers using a 1.5 T scanner. The results indicated the feasibility of the suggested sequence.

3.4.20 Oscillating Dual-Equilibrium Steady-State Angiography (ODESSA)

A novel technique of generating non-contrast angiograms was proposed [20]. This method utilized a modified steady-state free precession (SSFP) pulse sequence (see Fig. 3.32). The SSFP sequence was modified such that flowing material reaches a steady state that oscillates between two equilibrium values, while stationary material attains a single, non-oscillatory steady state. Subtraction of adjacent echoes results in large, uniform signal from all flowing spins and zero signal from stationary spins. Venous signal can be suppressed based on its reduced T2. ODESSA arterial signal was more than three times larger than that of traditional phase-contrast angiography (PCA) in the same scan time, and also compares favorably with other techniques of MR angiography (MRA). Pulse sequences are implemented in 2D, 3D, and volumetric-projection modes. Angiograms of the lower leg, generated in as few as 5 seconds, showed high arterial signal-to-noise ratio (SNR) and full suppression of other tissues.

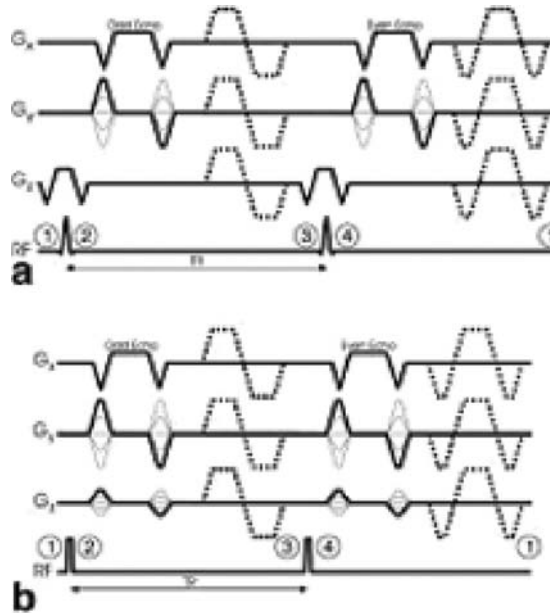


Figure 3.32: (a) Two-dimensional and (b) 3D ODESSA pulse sequences. Two TR intervals are shown. Each axis has zero net gradient area over the TR interval. During odd TRs (at left), a bipolar flow-encoding pulse follows readout on any axis. A triphasic pulse after even readouts, though not necessary, is included to mitigate imaging system nonidealities. The numbered locations correspond to spin states.

3.4.21 Fat-Suppressed 3D MRA

Appropriate rate of fat-suppression pulses (using spec IR spectral selective inversion recovery) were determined for fat-suppressed 3D magnetic resonance angiography (MRA) with an elliptical centric view order [21]. In abdominal 3D fast spoiled gradient echo (fast SPGR) with an elliptical centric view order, the spec IR pulse rate was changed from zero to one every 15 repetitions (in nine steps) in eight volunteers. In the equilibrium phase, abdominal contrast-enhanced 3D MRA was obtained by 3D fast SPGR using an elliptical centric order without fat-suppression and with two spec IR, and by fat-suppressed 3D fast SPGR with a sequential-centric view order (3D-EFGRE). Fat and vascular signals were estimated. Although 3D fast SPGR using an elliptical centric order with two spec IR placed every 15 TR and 3D-EFGRE effectively decreased fat signals, these sequences lengthened the breath-hold by 4–6 seconds compared

with non-fat suppressed sequence. 3D fat SPGR using an elliptical centric order and two spec IR reduced the fat signal by 30%. And provided good 3D MR angiography without substantial prolongation of breath-hold. Two spec IR can be used for generation of partially fat-suppressed abdominal 3D MRA without prolongation of the breath-hold when performing 3D fast SPGR using an elliptical centric view order.

3.4.22 Gadolinium Enhanced MRA with MR Cholangiography (MRC)

Simultaneously both methods were used in the preoperative evaluation of gallbladder carcinoma [22]. All MR images were analyzed in order to assess bile duct invasion, vascular invasion, hepatic invasion or metastasis, lymph node metastasis, and invasion into adjacent organs. The sensitivity and specificity of MR examination were distinctive 100% and 89% for bile duct invasion, 100% and 87% for vascular invasion, 67% and 89% for hepatic invasion, and 56% and 89% for lymph node metastasis supported by histopathologic findings. The “all-in-one” MR protocol, including MR imaging, MRC, and MRA, could be an effective diagnostic approach in the preoperative work-up for gallbladder carcinoma.

3.4.23 Ultrashort Contrast-Enhanced (CE) MRA

It was used for the morphologic evaluation of cerebral arteriovenous malformations (AVMs). The method was compared with conventional X-ray digital subtraction angiography (DSA) and time-of-flight (TOF) MRA to assess the angioarchitecture of the malformations that is essential for treatment planning and follow-up. Contrast-enhanced MRA was able to detect all AVMs seen on DSA, whereas the TOF MRA failed with a very small AVM [23]. However, there was no difference for the detection and delineation of feeding arteries and the AVM. The venous drainage patterns could always be clearly delineated in the CE MRA, whereas TOF MRA could demonstrate the exact venous drainage. Contrast-enhanced MRA was found to be superior to conventional TOF MRA in the assessment of the angioarchitecture of cerebral AVMs especially regarding the assessment of the venous drainage patterns. The superiority was

supported by the improved vessel-to-background contrast and contrast-to-noise ratios. The major limitation of this new technique was a low spatial resolution at the used time resolution that could be improved by further sequence modifications. Contrast-enhanced MRA is still an important additional imaging technique for treatment planning and follow-up of AVMs.

3.4.24 Quadruple Contrast Enhancement with MRA and MR Spectroscopic Imaging

This approach minimized the false tissue classifications by: (1) improving the lesion-to-tissue contrast on MR images by developing a fast imaging pulse sequence that incorporated both cerebrospinal fluid signal attenuation and magnetization transfer contrast (see Fig. 3.33); and (2) including information from MR flow images [24]. In pathologically defined abnormalities in the cortical gray matter (GM) and normal volunteers, high resolution MRI and short echo

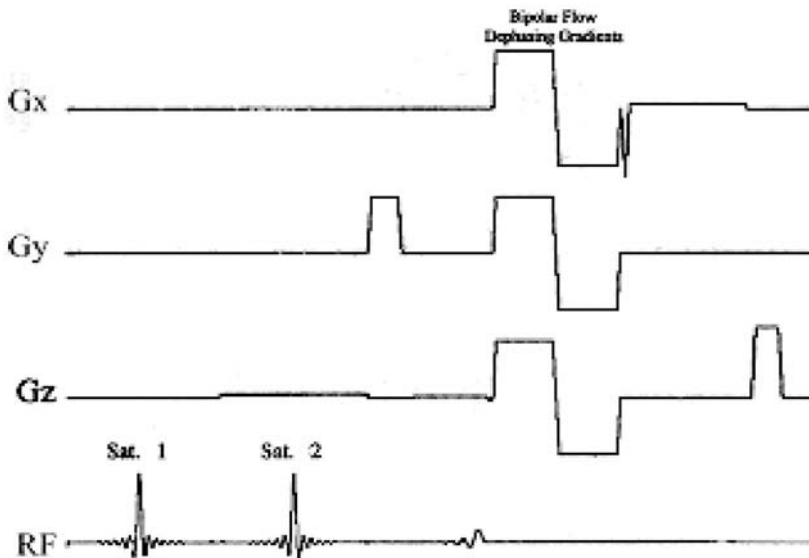


Figure 3.33: An approach is represented to improve the multiple sclerosis lesion-to-tissue contrast using MRA images by a fast imaging pulse sequence incorporating both cerebrospinal fluid signal and flow attenuation with magnetization transfer contrast. The technique suppressed gray matter or white matter and highlighted the lesion-to-tissue contrast.

proton magnetic resonance spectroscopic imaging (MRSI) estimated individual tissue contributions to the spectroscopic voxels in multiple sclerosis (MS).

3.4.25 Coronary MRA

For assessment of patients with atherosclerotic CAD, CMRA is reported useful for detection of patency of bypass grafts. Patients with suspected coronary artery anomalies and patients with Kawasaki disease and coronary aneurysms are among those for whom CMRA has demonstrated clinical usefulness. At centers with appropriate expertise and resources, CMRA also appears to be of value for exclusion of severe proximal multivessel CAD in selected patients. Data from multicenter trials defined the clinical role of CMRA, particularly as it relates to assessment of CAD. Future developments and enhancements of CMRA promise better lumen and coronary artery wall imaging. This may become the new target in noninvasive evaluation of CAD [25].

3.4.26 4D Phase Contrast (PC) Technique

4D PC technique was demonstrated for its feasibility that permits spatial and temporal coverage of an entire 3D volume [26]. It validated quantitatively the accuracy against an established time resolved 2D PC technique to explore advantages of the approach with regard to the 4D nature of the data. Time-resolved, 3D anatomical images were generated simultaneously with registered three-directional velocity vector fields. Improvements were compared to prior methods for gated and respiratory compensated image acquisition, interleaved flow encoding with freely selectable velocity encoding (VENC) along each spatial direction, and flexible trade-off between temporal resolution and total acquisition time. The implementation was validated against established 2D PC techniques using a well-defined phantom, and successfully applied in volunteer and patient examinations. Human studies were performed after contrast administration in order to compensate for loss of in-flow enhancement in the 4D approach. Advantages of the 4D approach included the complete spatial and temporal coverage of the cardiovascular region of interest and the ability to obtain high spatial resolution in all three dimensions with higher signal-to-noise ratio compared to 2D methods at the same resolution. In addition, the 4D nature of the data offered a variety of image processing options, such as magnitude and velocity

multi-planar reformation, 3D vector field plots, and velocity profiles mapped onto selected planes of interest.

3.4.27 Coronary MRA Receiver Operator Characteristics (ROC) Analysis

Magnetic resonance imaging (MRI) was evaluated as a potential noninvasive diagnostic tool to detect coronary artery bypass graft stenosis, in detecting stenotic grafts, including recipient vessels. Coronary angiography with baseline and stress flow mapping was performed. Marginal logistic regression was used to predict the probability for the presence of stenosis per graft type using multiple MRI variables. Receiver operator characteristics (ROC) analysis showed the diagnostic value of MRI. Sensitivity/specificity in detecting single vein grafts with stenosis $\geq 50\%$ and $\geq 70\%$ were significant. MRI with flow mapping was useful for identifying grafts and recipient vessels with flow-limiting stenosis. This proof-of-concept study suggested that noninvasive MRI detection of stenotic grafts in patients who present with recurrent chest pain after graft may be useful in selecting those in need of an invasive procedure [27].

3.4.28 MIP for Stenoses

MIP versus original source images (SI) were compared in respect to detection of coronary artery stenoses by means of magnetic resonance (MR) coronary angiography in the proximal and mid coronary segments and compared with x-ray angiography. The MIP reconstructions showed comparable accuracy to unprocessed SI. However, MIP post-processing was compromised by a higher number of images that were unable to be evaluated due to overlap of coronary arteries with adjacent cardiac structures [28].

3.4.29 3D Magnetization-Prepared True Fast Imaging with Steady-State Precession Sequence

Breath-hold coronary magnetic resonance (MR) angiography with and without preoxygenation was measured and the effect of preoxygenation on coronary artery imaging was evaluated. This sequence was employed for coronary MR

angiography. The extra imaging time allowed coronary artery imaging with increased spatial resolution [29].

3.4.30 3D Real-Time Navigator Magnetic Resonance (MR) Coronary Angiography

3D real-time navigator magnetic resonance (MR) coronary angiographic examination was reported for detection of significant coronary artery stenoses, with conventional coronary angiography as the standard of reference immediately before catheterization. It quantified coronary artery visualization, and evaluated the presence of significant narrowing or stenoses. Receiver operating characteristic (ROC) analysis signified that large portions of the coronary arteries could be visualized with MR coronary angiography. Imaging results were not consistently reliable, however, the examination was premature for routine clinical assessment of significant coronary artery stenosis owing to low sensitivity and large observer variability [30].

3.4.31 Free-breathing three-dimensional (3D) coronary magnetic resonance (MR) angiography

This method was reported to determine the anatomy of anomalous coronary arteries, in particular the relationship of the vessels to the aortic root. Multiple 3D volume slabs were acquired at the level of the sinuses of Valsalva by using diaphragmatic navigators for respiratory artifact suppression. The proximal anatomy of the coronary arteries was determined. Free-breathing 3D coronary MR angiography could be used to identify the proximal anatomy of anomalous coronary arteries [31].

3.4.32 BACSPIN (Breathing AutoCorrection with SPiral Interleaves) Coronary MRA Technique

Signal-to-noise ratio (SNR) of breath independent coronary magnetic resonance angiography (CMRA) was improved without increasing the number or duration of breath holds. In this BACSPIN technique, a single breath-held electrocardiogram (ECG)-gated multi-slice interleaved-spiral data set was acquired, followed

by repeated imaging of the same slices during free breathing. Each spiral interleaf from the breath-held data set was used as a standard for comparison with corresponding acquisitions at the same interleaf angle during free breathing. The most closely matched acquisitions are incorporated into a multi-slice, multi-average data set with increasing SNR over time. In-plane translations of the coronary artery could be measured and compensated for each accepted acquisition before combination with the other acquisitions. CMRA was performed with improved SNR and minimal motional blurring. BACSPIN provided a promising method for CMRA with improved SNR and limited breath-holding requirements [32].

3.4.33 Motion-Adapted Gating Window in Coronary MRA

An acquisition technique was reported that used subject-specific acquisition windows in the cardiac cycle and a motion-adapted gating window for respiratory navigator gating. Cardiac acquisition windows and trigger delays were determined individually from a coronary motion scan. Motion-adapted gating used a 2-mm acceptance window for the central 35% of k-space and a 6-mm window for the outer 65% of k-space. The adaptive technique was applied in patients who underwent coronary radiographic angiography. Scanning times with the adaptive technique were reduced for the right coronary artery and left coronary artery system compared with the conventional technique, due to the use of longer subject-specific acquisition windows in patients with low heart rates. Subjective and objective measurements of image quality showed no significant differences between the two techniques. Coronary MR angiography with subject-specific acquisition windows and motion-adapted respiratory gating reduced scanning times while maintaining image quality and provided high diagnostic accuracy for the detection of coronary artery stenosis [33].

3.4.34 Attenuated Coronary Blood—Myocardium In-Flow Contrast 3D Coronary Magnetic Resonance Angiography (CMRA)

The in-flow contrast between the coronary blood and the surrounding myocardium was attenuated as compared to thin-slab 2D techniques. The

application of a gadolinium (Gd)-based intravascular contrast agent provided an additional source of signal and contrast by reducing T1(blood) and supporting the visualization of more distal or branching segments of the coronary arterial tree. For imaging, an optimized free breathing, navigator-gated and -corrected 3D inversion recovery (IR) sequence was used. For comparison, state-of-the-art baseline 3D coronary MRA with T(2) preparation for non-exogenous contrast enhancement was acquired. The combination of IR 3D coronary MRA, sophisticated navigator technology, and B-22956 contrast agent allowed for an extensive visualization of the LCA system. Postcontrast showed a significant increase in both the signal-to-noise ratio (SNR) and contrast-to-noise ratio (CNR). Vessel sharpness of the left anterior descending (LAD) artery and the left coronary circumflex (LCx) were improved [34].

3.5 Limitations and Future Prospects

No specific MRA techniques have emerged so far that can provide sufficient sensitivity and specificity for quantification. MR angiography still remains a clinical choice of cardiovascular MR despite of cardiac and respiratory motion factors. Physical principles further highlight the intricacies and need of MRA technical improvements and modifications in coming years. From all techniques available, 2D/3D breath-hold coronary MRA(CMRA), black-blood FSE method, real-time navigator for respiratory gating with slice position correction and contrast enhanced CMRA have been evaluated clinically useful for coronary wall imaging. However, these high contrast angiography techniques suffer from limitations in temporal and spatial resolution and motion artifacts. These restrictions further limit its prediction value. Other hand, high contrast MR angiography techniques suffer from limitations in temporal and spatial resolution and motion artifacts. These advanced techniques have been described less sensitive <70% and specificity <75% while human artery risk in the wall is established >50% stenosis. These methods no doubt provide a quick way to image blood flow in a long segment of the artery for rapid burden measurements.

Other emerging MR techniques, such as water diffusion weighting, magnetization transfer weighting, steady-state free precession (SSFP) sequences, contrast enhancement methods may provide thin slices. Still measurements and plaque characterization methods are in infancy using thin slices. Some notable

improved blood suppression methods are promising for accurate imaging by dual-inversion 3D FSE imaging sequence with real-time navigator technology for high-resolution, free-breathing black-blood CMRA, delineation of coronary artery by echoplanar imaging. In general, in future, high-resolution MRA seems well suited to 3.0 T MR field strength since spatial resolution is often limited by S/N at 1.5 T. Initial feasibility of CMRA for intracranial and cervical studies is encouraging. 3.0 T and higher magnetic field scanners with superior field strength for 3DTOF and is extremely promising for 3DTOF and CMRA. The CMRA has advantages of shorter scan time and better depiction of slow flow hence it was the attention in last decade with combination of other modalities.

Questions

1. *What do you understand by term MRA?*
2. *How spatial encoding, spatial resolution show relationship?*
3. *What are MRA k-space trajectories and how do they are applied?*
4. *What are the unique properties of blood and MRA contrast agents?*
5. *How 'Black blood MRA' is unique and significant?*
6. *What are newer approaches commonly known as Bright blood MRA with t extragenous contrast?*
7. *How both Cine MRI and PC MRA are comparable?*
8. *How contrast enhanced bright blood MRA is unique and better clinical imaging modality?*
9. *What is present state-of-art in quantitative analysis of MRA images?*
10. *What are advanced approaches in vessel detection and artery-vessel separation in MRA image data sets?*

Bibliography

- [1] Maki, J. H., Wilson, G. J., Eubank, W. B., and Hoogeveen, R. M., Utilizing SENSE to achieve lower station sub-millimeter isotropic resolution and minimal venous enhancement in peripheral MR angiography, *J. Magn. Reson. Imaging*, Vol. 15, No. 4, pp. 484–491, 2002.
- [2] Hoffmann, U., Loewe, C., Bernhard, C., Weber, M., Cejna, M., Herold, C. J., and Schima, W., MRA of the lower extremities in patients with pulmonary embolism using a blood pool contrast agent: Initial experience, *J. Magn. Reson. Imaging*, Vol. 15, No. 4, pp. 429–437, 2002.
- [3] Carriero, A., Maggialelli, A., Pinto, D., Salcuni, M., Mansour, M., Petronelli, S., and Bonomo, L., Contrast-enhanced magnetic resonance angiography MoBI-trak in the study of peripheral vascular disease, *Cardiovasc. Intervent Radiol.* Vol. 25, No. 1, pp. 42–47, 2002.
- [4] Sutherland, G. R., Kaibara, T., Wallace, C., Tomanek, B., and Richter, M., Intraoperative assessment of aneurysm clipping using magnetic resonance angiography and diffusion-weighted imaging: Technical case report, *Neurosurgery* Vol. 50, No. 4, pp. 893–898, 2002.
- [5] Kalden, P., Mohrs, O., Kreitner, K. F., Thelen, M., and Schreiber, W. G., Preliminary results of coronary artery examination using a 3D-navigator sequence on a high performance MR system, *Rofo Fortschr Geb Rontgenstr Neuen Bildgeb Verfahr* Vol. 174, No. 2, pp. 183–186, 2002.
- [6] Sardanelli, F., Zandrino, F., Molinari, G., Iozzelli, A., Balbi, M., and Barsotti, A., MR evaluation of coronary stents with navigator echo and breath-hold cine gradient-echo techniques, *Eur. Radiol.*, Vol. 12, No. 1, pp. 193–200, 2002.
- [7] Ho, S. S., Chan, Y. L., Yeung, D. K., and Metreweli, C., Blood flow volume quantification of cerebral ischemia: comparison of three noninvasive imaging techniques of carotid and vertebral arteries, *AJR Am. J. Roentgenol.*, Vol. 178, No. 3, pp. 551–556, 2002.
- [8] Carr, J. C., Ma, J., Desphande, V., Pereles, S., Laub, G., and Finn, J. P., High-resolution breath-hold contrast-enhanced MR angiography of

the entire carotid circulation. *AJR Am. J. Roentgenol.*, Vol. 178, No. 3, pp. 543–549, 2002.

- [9] Huber, M. E., Oelhafen, M. E., Kozerke, S., Weber, O. M., and Boesiger, P., Single breath-hold extended free-breathing navigator-gated three-dimensional coronary MRA, *J. Magn. Reson. Imaging*, Vol. 15, No. 2, pp. 210–214, 2002.
- [10] Stuber, M., Bornert, P., Spuentrup, E., Botnar, R. M., and Manning, W. J., Selective three-dimensional visualization of the coronary arterial lumen using arterial spin tagging, *Magn. Reson. Med.*, Vol. 47, No. 2, pp. 322–329, 2002.
- [11] Holmqvist, C., Stahlberg, F., Hanseus, K., Hochbergs, P., Sandstrom, S., Larsson, E. M., and Laurin, S., Collateral flow in coarctation of the aorta with magnetic resonance velocity mapping: correlation to morphological imaging of collateral vessels, *J. Magn. Reson. Imaging*, Vol. 15, No. 1, pp. 39–46, 2002.
- [12] Greil, G. F., Powell, A. J., Gildein, H. P., and Geva, T., Gadolinium-enhanced three-dimensional magnetic resonance angiography of pulmonary and systemic venous anomalies, *J. American Coll. Cardiol.*, Vol. 39, No. 2, pp. 335–341, 2002.
- [13] Naganawa, S., Koshikawa, T., Fukatsu, H., Sakurai, Y., Ichinose, N., Ishiguchi, T., and Ishigaki, T., Contrast-enhanced MR angiography of the carotid artery using 3D time-resolved imaging of contrast kinetics: comparison with real-time fluoroscopic triggered 3D-elliptical centric view ordering, *Radiat. Med.*, Vol. 19, No. 4, pp. 185–192, 2001.
- [14] McGee, K. P., Felmlee, J. P., Jack, C. R. Jr., Manduca, A., Riederer, S. J., and Ehman, R. L., Autocorrection of three-dimensional time-of-flight MR angiography of the Circle of Willis, *AJR Am. J. Roentgenol.*, Vol. 176 No. 2, pp. 513–518, 2001.
- [15] Amann, M., Bock, M., Floemer, F., Schoenberg, S. O., Schad, L. R., Three-dimensional spiral MR imaging: Application to renal multiphase contrast-enhanced angiography, *Magnetic Resonance in Medicine*, Vol. 48, No. 2, pp. 290–296, 2002.

- [16] Jeong, E. K., Parker, D. L., Tsuruda, J. S., and Won, J. Y., Reduction of flow-related signal loss in flow-compensated 3D TOF MR angiography, using variable echo time (3D TOF-VTE), *Magn. Reson. Med.*, Vol. 48, No. 4, pp. 667–676, 2002.
- [17] Spuentrup, E., Manning, W. J., Bornert, P., Kissinger, K. V., Botnar, R. M., and Stuber, M., Renal arteries: navigator-gated balanced fast field-echo projection MR angiography with aortic spin labeling: initial experience, *Radiology*, Vol. 225, No. 2, pp. 589–596, 2002.
- [18] Bernstein, M. A., Huston, J., III, Lin, C., Gibbs, G. F., and Felmlee, J. P., High-resolution intracranial and cervical MRA at 3.0 T: Technical considerations and initial experience, *Magn. Reson. Med.*, Vol. 46, No. 5, pp. 955–962, 2001.
- [19] Azhari, H., McKenzie, C. A., and Edelman, R. R., MR angiography using spin-lock flow tagging, *Magn. Reson. Med.*, Vol. 46, No. 5, pp. 1041–1044, 2001.
- [20] Overall, W. R., Conolly, S. M., Nishimura, D. G., and Hu, B. S., Oscillating dual-equilibrium steady-state angiography, *Magn. Reson. Med.*, Vol. 47, No. 3, pp. 513–522, 2002.
- [21] Amano, Y., Amano, M., Matsuda, T., Tsuchihashi, T., Takahama, K., and Kumazaki, T., Fat-suppressed three-dimensional MR angiography technique with elliptical centric view order and no prolonged breath-holding time, *J. Magn. Reson. Imaging*, Vol. 16, No. 6, pp. 707–715, 2002.
- [22] Kim, J. H., Kim, T. K., Eun, H. W., Kim, B. S., Lee, M. G., Kim, P. N., and Ha, H. K., Preoperative evaluation of gall bladder carcinoma: Efficacy of combined use of MR imaging, MR cholangiography, and contrast-enhanced dual-phase three-dimensional MR angiography, *J. Magn. Reson. Imaging*, Vol. 16, No. 6, pp. 676–684, 2002.
- [23] Duran, M., Schoenberg, S. O., Yuh, W. T., Knopp, M. V., van Kaick, G., and Essig, M., Cerebral arteriovenous malformations: morphologic evaluation by ultrashort 3D gadolinium-enhanced MR angiography, *Euro. Radiol.*, Vol. 12, No. 12, pp. 2957–2964, 2002.

- [24] Sharma, R., Narayana, P. A., and Wolinsky, J. S., Grey matter abnormalities in multiple sclerosis: proton magnetic resonance spectroscopic imaging. *Multiple Sclerosis*, Vol. 7, No. 4, pp. 221–226, 2001.
- [25] Danias, P. G., Stuber, M., Botnar, R. M., Kissinger, K. V., Yeon, S. B., Rofsky, N. M., and Manning, W. J. Coronary MR angiography clinical applications and potential for imaging coronary artery disease. *Magn. Reson. Imaging Clin. North Am.*, Vol. 11, No. 1, pp. 81–99, 2003.
- [26] Markl, M., Chan, F. P., Alley, M. T., Wedding, K. L., Draney, M. T., Elkins, C. J., and Parker, D. W., Wicker, R., Taylor, C. A., Herfkens, R. J., Pelc, N. J. Time-resolved three-dimensional phase-contrast MRI. *J. Magn. Reson. Imaging*, Vol. 17, No. 4, pp. 499–506, 2003.
- [27] Langerak, S. E., Vliegen, H. W., Jukema, J. W., Kunz, P., Zwinderman, A. H., and Lamb, H. J., van der Wall, E. E., and de Roos, A. Value of magnetic resonance imaging for the noninvasive detection of stenosis in coronary artery bypass grafts and recipient coronary arteries. *Circulation.*, Vol. 107, No. 11, pp. 1502–1508, 2003.
- [28] Regenfus, M., Ropers, D., Achenbach, S., Schlundt, C., Kessler, W., Laub, G., Moshage, W., and Daniel, W. G. Diagnostic value of maximum intensity projections versus source images for assessment of contrast-enhanced three-dimensional breath-hold magnetic resonance coronary angiography. *Invest. Radiol.*, Vol. 38, No. 4, pp. 200–206, 2003.
- [29] McCarthy, R. M., Shea, S. M., Deshpande, V. S., Green, J. D., Pereles, F. S., Carr, J. C., Finn, J. P., and Li, D. Coronary MR angiography: true FISP imaging improved by prolonging breath holds with preoxygenation in healthy volunteers. *Radiology.*, Vol. 227, No. 1, pp. 283–288, 2003.
- [30] Bogaert, J., Kuzo, R., Dymarkowski, S., Beckers, R., Piessens, J., and Rademakers, F. E. Coronary artery imaging with real-time navigator three-dimensional turbo-field-echo MR coronary angiography: initial experience *Radiology.*, Vol. 226, No. 3, pp. 707–16, 2003.
- [31] Bunce, N. H., Lorenz, C. H., Keegan, J., Lesser, J., Reyes, E. M., Firmin, D. N., and Pennell, D. J. Coronary artery anomalies: assessment with

- free-breathing three-dimensional coronary MR angiography. *Radiology*, Vol. 227, No. 1, pp. 201–208, 2003.
- [32] Hardy, C. J., Zhao, L., Zong, X., Saranathan, M., and Yucel, E. K. Coronary MR angiography: respiratory motion correction with BACSPIN. *J. Magn. Reson. Imaging*, Vol. 17, No. 2, pp. 170–176, 2003.
- [33] Plein, S., Jones, T. R., Ridgway, J. P., and Sivananthan, M. U. Three-dimensional coronary MR angiography performed with subject-specific cardiac acquisition windows and motion-adapted respiratory gating. *AJR Am. J. Roentgen.*, Vol. 180, No. 2, pp. 505–512, 2003.
- [34] Huber, M. E., Paetsch, I., Schnackenburg, B., Bornstedt, A., Nagel, E., Fleck, E., Boesiger, P., Maggioni, F., Cavagna, F. M., and Stuber, M. Performance of a new gadolinium-based intravascular contrast agent in free-breathing inversion-recovery 3D coronary MRA. *Magn. Reson. Med.*, Vol. 49, No. 1, pp. 115–121, 2003.

Chapter 4

Recent Advances in the Level Set Method

*David Chopp*¹

4.1 Introduction

The level set method was introduced in the groundbreaking paper by Osher and Sethian in 1988 [85]. The ground work for this paper began with a paper by Sethian on moving fronts [102]. The original application for this work was to solve problems in flame propagation, where the flame speed was given as a function of the local mean curvature of the propagating flame front. The work in [85] combined two fundamental ideas together in a unique way, and formed the basis for the level set method in wide use today.

The first fundamental idea was the choice of an implicit representation for the moving interface. At first glance, this appears to be a completely unnatural choice; it is more difficult to specifically locate the interface at any given time, and, in its simplest form, requires an order of magnitude greater computational cost. However, this approach also offers powerful geometric properties which no other method can as easily provide, and can be extended to higher dimensions with vastly greater ease. Specifically, the implicit representation allows for changes in the topology of the interface to happen naturally without requiring collision detection and interface reconstruction as required by Lagrangian-type methods. Also, the evolution equation they derived for propagating the interface can be written entirely in terms of the embedding function, so that (at least for their application) the actual location of the interface at any given time need not be determined.

¹Engineering Sciences and Applied Mathematics Department, Northwestern University.

The second fundamental idea was the adaptation of numerical methods developed for hyperbolic conservation laws. The field of numerical hyperbolic conservation laws is a mature field with a substantial body of research devoted to both the theory and practice of these methods. Much of this field is concerned with the construction of numerical flux functions, which approximate the physical flux function in a way which respects the propagating characteristics of the problem. The resulting numerical methods more accurately compute the speeds for propagating shocks, and find the unique entropy condition satisfying rarefaction fans. In [102], Sethian observed that the theory of hyperbolic conservation laws could be applied to the problem of propagating interfaces. This naturally led to [85], where the equation for propagating the interface using the implicit representation was formulated as the integral of a hyperbolic conservation law. In the context of moving interfaces, the shocks became corners in the interface, and the rarefaction fans became regions of interface expansion.

The coupling of the numerical hyperbolic conservation laws with the implicit representation led to the first level set method, which was demonstrated to be a powerful, robust method for solving the flame propagation problem.

Though the level set method, in its original form, was successful for the original application, it was soon observed by Chopp [19] that a fundamental problem in the method still existed. At that point, nearly all of the applications of the level set method involved interface speed functions which depended solely upon mean curvature. This class of problems is very special, as indicated in [39–42], because the embedding function maintained bounded gradients almost everywhere, giving the method additional stability properties. This property does not hold for a general interface speed function, and so for the level set method to be generalizable, one important modification was required in order to maintain a stable level set method.

The key modification to the level set method, proposed in [19], was to observe that forcing the embedding function to maintain bounded gradients was possible, without changing the underlying motion of the interface. This process was called reinitialization, and it essentially forced the embedding function to be the signed distance function, even if the level set evolution equation would not do it on its own. Once this piece was added to the level set method toolbox, the level set method exploded in popularity, being used in a wide array of interface motion applications.

In the remainder of this chapter, we will begin by giving a more detailed description of the basic level set method. Next, some of the recent modifications to the method will be explored, particularly those relevant to the medical imaging community. The chapter will conclude with a brief review of the myriad applications of the level set and fast marching methods that have been published over the last few years.

4.2 Basic Level Set Method

In this section, the necessary pieces for implementing the general level set method are presented. These include the implicit representation of the interface, the equation which describes interface motion, and the gradient control process. There are now two methods for gradient control: reinitialization and velocity extensions. Both of these methods will require some background information on the fast marching method for implementation. The fast marching method is an interesting method in its own right, and a description of this method will also be presented.

4.2.1 The Level Set Representation

At the heart of the level set method is the implicit representation of the interface. If the interface is given by Γ , Γ can then be represented by a function ϕ , called the level set function, defined by the signed distance function

$$\phi(\mathbf{x}) = \pm d_{\Gamma}(\mathbf{x}). \quad (4.1)$$

Here $d_{\Gamma}(\mathbf{x})$ is the distance from the point $\mathbf{x}$ to the interface Γ , and the sign is determined so that it is negative on the inside and positive on the outside. At any time, the interface can be recovered by locating the set

$$\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\} \equiv \phi^{-1}(0). \quad (4.2)$$

For example, a circle interface and the corresponding level set function representation are shown in Fig. 4.1.

For most applications, this representation works well, but there are interfaces which cannot use it. For example, interfaces with triple junctions or any interface which does not have a clearly defined inside and outside cannot easily

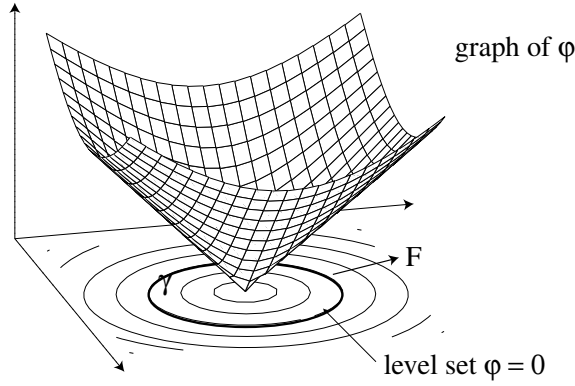


Figure 4.1: Example of a level set representation of a circle.

be represented using a level set function. However, the level set method, with some modifications, can even be applied to these cases as well. These variations will be discussed in Section 4.3.

Once the level set function, ϕ , is constructed, the evolution equation for the interface must be rewritten in terms of ϕ . Given the interface Γ , let $F(\mathbf{x})$ be the speed of the interface in the direction of the normal (see Fig. 4.2). Let $\mathbf{x}(t)$ be a point on the interface which evolves with the interface, then $\phi(\mathbf{x}(t), t) \equiv 0$ for all t . Differentiating with respect to t gives

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \frac{d\mathbf{x}}{dt} = 0. \quad (4.3)$$

Now, the evolution of $\mathbf{x}(t)$ can be described by

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}(t), t)\mathbf{n}, \quad (4.4)$$

where $\mathbf{n}$ is the unit normal to the interface. Use the fact that the unit normal can also be computed to be $\mathbf{n} = \nabla \phi / \|\nabla \phi\|$, and substituting this with Eq. 4.4 into

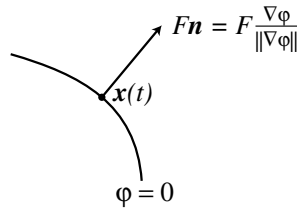


Figure 4.2: Illustration of the relationship between $\phi(\mathbf{x}, t)$, $\mathbf{x}$, and F .

Eq. 4.3 gives the level set evolution equation,

$$\frac{\partial \phi}{\partial t} + F \|\nabla \phi\| = 0. \quad (4.5)$$

This is the key evolution equation that was introduced in [85]. Through this equation, the motion of the interface $\Gamma(t)$ is captured through Eq. 4.5 so that at any time t ,

$$\Gamma(t) = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}. \quad (4.6)$$

One key observation about Eq. 4.5 is that we have implicitly assumed that the function F is known over the entire domain of ϕ . Very often, this is not the case, and F is only defined on the interface itself. However, this problem can be solved by using velocity extensions, which will be discussed in Section 4.2.5.

4.2.2 Numerical Implementation of the Level Set Method

As noted in the introduction, the second critical part of the paper by Osher and Sethian was the use of methods borrowed from hyperbolic conservation laws for discretizing the level set equation Eq. 4.5. This concept was generalized in [103], where numerical flux functions designed for hyperbolic conservation laws were used to solve Hamilton–Jacobi equations of the form

$$\frac{\partial \phi}{\partial t} + H(\nabla \phi) = 0. \quad (4.7)$$

Here, the function $H(\nabla \phi)$ is called the Hamiltonian, and it is a function of the gradient of ϕ . There is a rich history of numerical methods for hyperbolic conservation laws. An excellent review of numerical methods for hyperbolic conservation laws can be found in [75].

In the case of the level set method, the Hamiltonian is given by

$$H(\nabla \phi) = F \|\nabla \phi\|. \quad (4.8)$$

A first-order numerical Hamiltonian for solving Eq. 4.7 is given by Godunov’s method, where

$$\begin{aligned} \phi_{ij}^{n+1} = & \phi_{ij}^n - \Delta t (\max(\text{sign}(F_{ij})D_{-x}\phi_{ij}^n, -\text{sign}(F_{ij})D_{+x}\phi_{ij}^n, 0)^2 \\ & + \max(\text{sign}(F_{ij})D_{-y}\phi_{ij}^n, -\text{sign}(F_{ij})D_{+y}\phi_{ij}^n, 0)^2)^{1/2}. \end{aligned} \quad (4.9)$$

Here, the finite difference operators $D_{\pm x}$ are defined by

$$D_{+x}\phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \quad D_{-x}\phi_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}. \quad (4.10)$$

The operators $D_{\pm y}$ are defined in a similar manner for the j th index. Note that the numerical flux function, the term multiplied by Δt in Eq. 4.9, senses the direction in which the interface is moving, then chooses the finite difference approximation which looks in the correct direction, also known as the upwind direction.

A second-order method based upon the ENO method [55] is given by

$$\begin{aligned} \phi_{ij}^{n+1} = & \phi_{ij}^n - \Delta t(\max(\text{sign}(F_{ij})A, -\text{sign}(F_{ij})B, 0)^2 \\ & + \max(\text{sign}(F_{ij})C, -\text{sign}(F_{ij})D, 0)^2)^{1/2}, \end{aligned} \quad (4.11)$$

where

$$A = D_{-x}\phi_{ij}^n + \frac{\Delta x}{2} \min\text{mod}(D_{-x}D_{-x}\phi_{ij}^n, D_{-x}D_{+x}\phi_{ij}^n), \quad (4.12)$$

$$B = D_{+x}\phi_{ij}^n + \frac{\Delta x}{2} \min\text{mod}(D_{+x}D_{-x}\phi_{ij}^n, D_{+x}D_{+x}\phi_{ij}^n), \quad (4.13)$$

$$C = D_{-y}\phi_{ij}^n + \frac{\Delta y}{2} \min\text{mod}(D_{-y}D_{-y}\phi_{ij}^n, D_{-y}D_{+y}\phi_{ij}^n), \quad (4.14)$$

$$D = D_{+y}\phi_{ij}^n + \frac{\Delta y}{2} \min\text{mod}(D_{+y}D_{-y}\phi_{ij}^n, D_{+y}D_{+y}\phi_{ij}^n), \quad (4.15)$$

and where

$$\min\text{mod}(a, b) = \frac{1}{2}(\text{sign}(a) + \text{sign}(b)) \min(|a|, |b|). \quad (4.16)$$

In general, the speed function, F , in Eq. 4.5 is split into $F = F_{\text{adv}} + F_{\text{diff}}$, where F_{adv} is the advective part and F_{diff} is the diffusive part. When constructing the numerical method for solving Eq. 4.5, the numerical flux function is used for the advective part, and the diffusive part is discretized using standard central differences.

To illustrate this, we take an example used in [85], where $F = 1 - \epsilon\kappa$, $0 < \epsilon \ll 1$, and κ is the mean curvature given by

$$\kappa = \frac{\phi_{xx}\phi_y^2 + \phi_{yy}\phi_x^2 - 2\phi_{xy}\phi_x\phi_y}{(\phi_x^2 + \phi_y^2)^{3/2}}. \quad (4.17)$$

In this example, F is broken down so that $F_{\text{adv}} = 1$ and $F_{\text{diff}} = -\epsilon\kappa$. Using Godunov's method for the advective term and central differences for the diffusive

term gives

$$\begin{aligned}\phi_{ij}^{n+1} = & \phi_{ij}^n - \Delta t(\max(\text{sign}(F_{ij})D_{-x}\phi_{ij}^n, -\text{sign}(F_{ij})D_{+x}\phi_{ij}^n, 0)^2 \\ & + \max(\text{sign}(F_{ij})D_{-y}\phi_{ij}^n, -\text{sign}(F_{ij})D_{+y}\phi_{ij}^n, 0)^2)^{1/2} \\ & + \epsilon \frac{D_{+x}D_{-x}\phi_{ij}D_{0y}\phi_{ij} + D_{+y}D_{-y}\phi_{ij}D_{0x}\phi_{ij} - 2D_{0x}D_{0y}\phi_{ij}D_{0x}\phi_{ij}D_{0y}\phi_{ij}}{(D_{0x}\phi_{ij})^2 + (D_{0y}\phi_{ij})^2}.\end{aligned}\quad (4.18)$$

Here, the difference operators, D_{0x} , D_{0y} , are the central finite difference operators defined by

$$D_{0x}\phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}, \quad D_{0y}\phi_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y}. \quad (4.19)$$

4.2.3 The Fast Marching Method

An interesting method related to the level set method is the fast marching method, which was introduced by Sethian [105, 106]. While the fast marching method is used for some subsidiary algorithms within the general level set method, this method is interesting in its own right. The fast marching method solves a subclass of the problems normally solved with the level set method, but it does so much more quickly.

Like the level set method, the fast marching method also uses an implicit representation for an evolving interface, but for the fast marching method, the embedding function carries much more information. For the fast marching method, the entire evolution of the interface is encoded in the embedding function, not just a single time slice. In other words, the location of the interface at time t is given by the set

$$\Gamma(t) = \{\mathbf{x} : \phi(\mathbf{x}) = t\}. \quad (4.20)$$

As a result, in the fast marching method, the embedding function, ϕ , has no time dependency.

The embedding function, ϕ , is constructed by solving a static Hamilton–Jacobi equation of the form

$$F\|\nabla\phi\| = 1, \quad (4.21)$$

where F is again the speed of the interface. What makes the fast marching method fast is the fact that Eq. 4.21 can be solved with one pass over the mesh.

This contrasts with the level set method, where each time step requires an additional pass over the mesh to evolve the level set function in time.

The implementation of the fast marching method also uses the numerical flux functions discussed in Section 4.2.2; however, in this case, only one-sided differences such as Godunov's method may be used. For example, suppose the values of $\phi_{i-1,j}$, $\phi_{i,j+1}$ are already determined, and we wish to compute ϕ_{ij} . Then Eq. 4.21 is discretized using one-sided differences to obtain

$$F_{ij}^2((D_{-x}\phi_{ij})^2 + (D_{+y}\phi_{ij})^2) = 1. \quad (4.22)$$

This equation can be rewritten as a quadratic in terms of the unknown ϕ_{ij} :

$$\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)\phi_{i,j}^2 - 2\left(\frac{\phi_{i-1,j}}{\Delta x^2} + \frac{\phi_{i,j+1}}{\Delta y^2}\right)\phi_{i,j} + \frac{\phi_{i-1,j}^2}{\Delta x^2} + \frac{\phi_{i,j+1}^2}{\Delta y^2} - \frac{1}{F^2} = 0. \quad (4.23)$$

In most cases, solving Eq. 4.23 will produce two solutions, one which is less than the values of $\phi_{i-1,j}$, $\phi_{i,j+1}$, and one which is greater. The larger of the two values is always chosen because of the causality assumption made by this method; values that are unknown are always greater than the known values.

Occasionally, Eq. 4.23 will not have any real roots. In that case, each of the coordinate directions is considered separately. For example, if we consider the x -direction, we assume that $\partial\phi/\partial y = 0$, and then discretize Eq. 4.21 to get

$$F_{ij}D_{-x}\phi_{ij} = 1. \quad (4.24)$$

This equation is linear and is easily solved for ϕ_{ij} . Similarly, the y -direction is considered, and the smaller of the two solutions is taken as the new estimate for ϕ_{ij} .

The key to solving Eq. 4.21 in one pass is to traverse the mesh in the proper order. The grid points must be evaluated in the order of increasing t . This is accomplished by using a sorted heap which always keeps track of which grid point is to be evaluated next. To begin, the set of grid points is divided into three disjoint sets, the *accepted* points A , the *tentative* points T , and the *distant* points D . The accepted points in A are the points $\mathbf{x}_{ij}$ for which the computed value of ϕ_{ij} is already determined. The tentative points in T are the points $\mathbf{x}_{ij}$ for which a tentative value for ϕ_{ij} is computed. The remainder of the points are in the set D . One by one, points in T are taken, in order of increasing value of ϕ_{ij} , from the set T into A . Each time, points ϕ_{ij} in D which become adjacent to points in the set A are moved into the set T and a tentative value

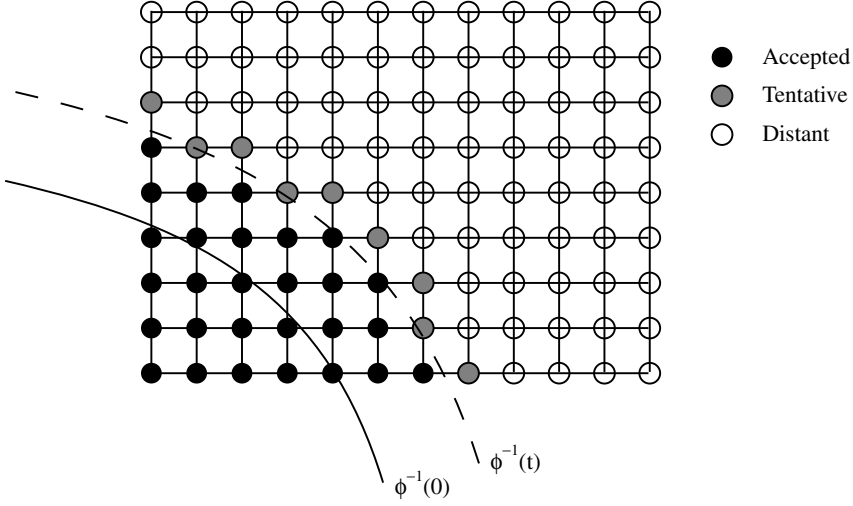


Figure 4.3: Illustration of the sets A , T , and D associated with the fast marching method. This figure reprinted from [22].

for ϕ_{ij} is computed using Eq. 4.21. The algorithm terminates when all points have migrated into the set A . See Fig. 4.3 for an illustration of the sets A , T , and D .

The full algorithm for the fast marching method becomes:

1. Initialize all the points adjacent to the initial interface with an initial value, put those points in A . A discussion about initialization follows in Section 4.2.3. All points $\mathbf{x}_{i,j} \notin A$, adjacent to a point in A , are given initial estimates for $\phi_{i,j}$ by solving Eq. 4.21. These points are tentative points and put in the set T . All remaining points unaccounted for are placed in D and given initial value of $\phi_{i,j} = +\infty$.
2. Choose the point $\mathbf{x}_{i,j} \in T$ which has the smallest value of $\phi_{i,j}$ and move it into A .
3. Any point which is adjacent to $\mathbf{x}_{i,j}$ (i.e. the points $\mathbf{x}_{i-1,j}$, $\mathbf{x}_{i,j-1}$, $\mathbf{x}_{i+1,j}$, and $\mathbf{x}_{i,j+1}$) which is in T has its value $\phi_{i,j}$ recalculated using Eq. 4.21. Any point adjacent to $\mathbf{x}_{i,j}$ and in D has its value $\phi_{i,j}$ computed using Eq. 4.21 and is moved into the set T .
4. If $T \neq \emptyset$, go to step 2.

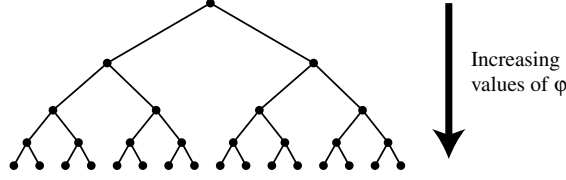


Figure 4.4: Example of a binary tree for the heap sort algorithm.

A higher order version of the fast marching method can be obtained by replacing Eq. 4.23 with

$$\begin{aligned}
 & \max(D_{-x}\phi_{i,j} + s_{x,-1} \frac{\Delta x}{2} D_{-x} D_{-x} \phi_{i,j} + s_{x,-1} s_{x,-2} \frac{\Delta x^2}{6} D_{-x} D_{-x} D_{-x} \phi_{i,j}, \\
 & -D_{+x}\phi_{i,j} - s_{x,1} \frac{\Delta x}{2} D_{+x} D_{+x} \phi_{i,j} - s_{x,1} s_{x,2} \frac{\Delta x^2}{6} D_{+x} D_{+x} D_{+x} \phi_{i,j}, 0)^2 \\
 & + \max(D_{-y}\phi_{i,j} + s_{y,-1} \frac{\Delta y}{2} D_{-y} D_{-y} \phi_{i,j} + s_{y,-1} s_{y,-2} \frac{\Delta y^2}{6} D_{-y} D_{-y} D_{-y} \phi_{i,j}, \\
 & -D_{+y}\phi_{i,j} - s_{y,1} \frac{\Delta y}{2} D_{+y} D_{+y} \phi_{i,j} - s_{y,1} s_{y,2} \frac{\Delta y^2}{6} D_{+y} D_{+y} D_{+y} \phi_{i,j}, 0)^2 \\
 & = \frac{1}{F_{i,j}^2}. \tag{4.25}
 \end{aligned}$$

The fast marching method algorithm presented in [105], is first-order accurate and can be recovered from Eq. 4.25 by taking all the switches $s_{*,*} = 0$. The second-order accurate method presented in [106] can also be recovered from Eq. 4.25 by taking all the switches $s_{*,\pm 2} = 0$.

The Heap-Sort Algorithm

The heap sort algorithm employed in the fast marching method is a balanced binary-tree structure which always maintains the smallest value of ϕ at the top. For purposes of illustration, see Fig. 4.4. The top of the tree is indicated by the single node at the top in Fig. 4.4. Each of the nodes connected to the top is called the child of that node, and the top node is the parent of its children. Except for the top node, each node has one parent, and may have zero, one, or two children depending upon where it is in the tree.

The operations on the tree that are required for the fast marching method are:

1. Resort the tree from one element.

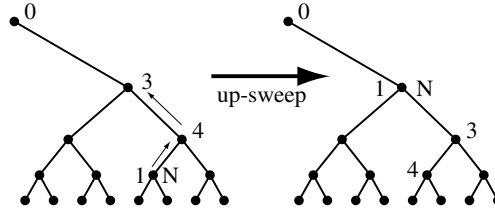


Figure 4.5: Example of the up-sweep for re-sorting a tree.

It is important that any operation on the tree ensures that after the operation, the tree preserves its property that any parent node has a smaller value of ϕ than either of its children. Occasionally, an operation on a particular node may mean that it is no longer correctly placed. This requires the tree to be re-sorted to accommodate this modified node. Either an up-sweep or a down-sweep process is required to restore the tree structure. Suppose there is a single misplaced node, N . First, compare N with its parent. If N is smaller than its parent, than an up-sweep is required. Otherwise, N is compared with its children, and if N is larger than either child, a down-sweep is used.

In the up-sweep, since N is smaller than its parent, N and its parent are exchanged. This process continues, with N comparing with its parent, until the parent is smaller or N has reached the top of the tree; see Fig. 4.5 for an illustration.

In the down-sweep, the node N is compared against its children. If N is smaller than either child, it is exchanged with the smaller of its two children. Like the up-sweep, this process is repeated until N is smaller than both of its children, or reaches the end of the tree. The down-sweep is illustrated in Fig. 4.6.

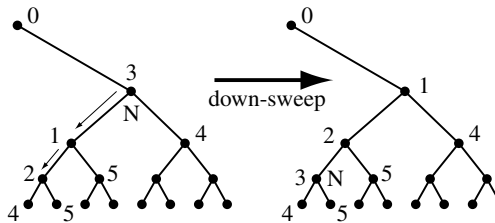


Figure 4.6: Example of the down-sweep for re-sorting a tree.

2. Remove the smallest (top) node of the tree.

When the top node of the tree is removed, the child of the top node, whose value for ϕ is smallest, is chosen to be the new top node. This process of promoting the smallest child up the tree is then propagated down until a node with less than two children is detected. This process preserves the property of the tree that parent nodes always have a smaller value of ϕ than the children.

3. Add a new node to the tree.

When a grid point is moved from the set D to T , it is also added to the tree. Since the initial estimate for ϕ at this point is likely to be larger than any of those already in the tree, it is best to add the node to an outer branch. For purposes of efficiency, care should be taken to keep the tree as balanced as possible, hence the new node should be added to the sparsest part of the tree. Once the node is appended, an up-sweep is performed to ensure proper placement.

4. Change the key value of an element in the tree.

When a grid point value is changed, it may require the tree to be re-sorted. If the value of the node is increased, then a down-sweep is done, and if the value is decreased, an up-sweep is done.

Initialization of the Fast Marching Method

The best form of initialization is where the exact solution is assigned to all the points in the original set A . These are all the nodes which are immediately adjacent to the initial interface. Most often, the exact solution is not known, and the initial values for the set A must be approximated from the initial data.

The method for initializing the set A given in [105, 106] is only first-order accurate, and can be prone to errors which will propagate through the remainder of the calculation. It was shown in [22] that a more accurate method is available, which can drive higher order fast marching method solutions.

The underpinning of this higher degree of accuracy around the initial front is the use of a bicubic interpolation function p which is a second-order accurate local representation of a level set function ϕ , i.e. $p(\mathbf{x}) \approx \phi(\mathbf{x})$. The interpolation function $p(\mathbf{x})$ can serve many purposes, including second-order

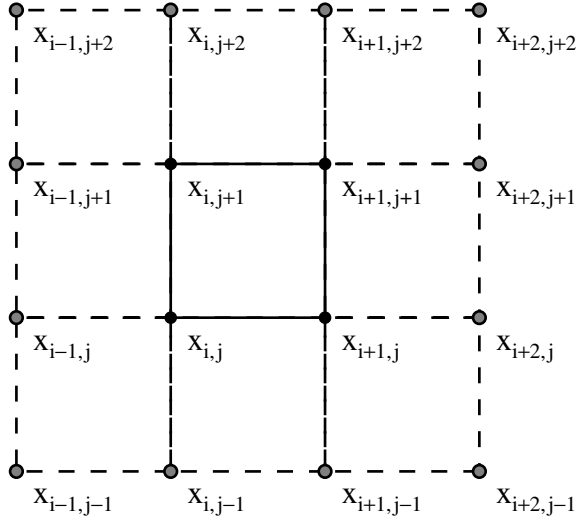


Figure 4.7: Sample portion of the mesh where a bicubic interpolation is used. This figure reprinted from [22].

accuracy for the distance to the zero level set, subgrid resolution of the shape of the interface, as well as subgrid resolution of the level set function $\phi(\mathbf{x})$ itself.

We begin with a description of the bicubic interpolation for a level set function given on a rectangular mesh. The approximation is done locally in a box of the mesh bounded by grid points, call them $\mathbf{x}_{i,j}$, $\mathbf{x}_{i+1,j}$, $\mathbf{x}_{i,j+1}$, and $\mathbf{x}_{i+1,j+1}$, as in Fig. 4.7.

A bicubic interpolation $p(\mathbf{x})$ of a function $\phi(\mathbf{x})$ is a function

$$p(\mathbf{x}) = p(x, y) = \sum_{m=0}^3 \sum_{n=0}^3 a_{m,n} x^m y^n, \quad (4.26)$$

which solves the following set of equations:

$$\begin{aligned} p(\mathbf{x}_{k,\ell}) &= \phi(\mathbf{x}_{k,\ell}) \\ \frac{\partial p}{\partial x}(\mathbf{x}_{k,\ell}) &= \frac{\partial \phi}{\partial x}(\mathbf{x}_{k,\ell}) \\ \frac{\partial p}{\partial y}(\mathbf{x}_{k,\ell}) &= \frac{\partial \phi}{\partial y}(\mathbf{x}_{k,\ell}) \\ \frac{\partial^2 p}{\partial x \partial y}(\mathbf{x}_{k,\ell}) &= \frac{\partial^2 \phi}{\partial x \partial y}(\mathbf{x}_{k,\ell}) \end{aligned}$$

for $k = i, i + 1, \ell = j, j + 1$. This gives 16 equations for the 16 unknown coefficients $a_{m,n}$. Solving for $a_{m,n}$ makes $p(x, y)$ a bicubic interpolating function of $\phi(x, y)$ on the rectangle bounded by the corners $\mathbf{x}_{i,j}$, $\mathbf{x}_{i+1,j}$, $\mathbf{x}_{i,j+1}$, and $\mathbf{x}_{i+1,j+1}$.

Since ϕ is only known on the mesh points, the values for the derivatives of ϕ must be approximated. We use second-order finite difference approximations for the derivatives of ϕ :

$$\begin{aligned}\frac{\partial \phi}{\partial x}(\mathbf{x}_{m,n}) &\approx \frac{1}{2\Delta x}(\phi(\mathbf{x}_{m+1,n}) - \phi(\mathbf{x}_{m-1,n})) \\ \frac{\partial \phi}{\partial y}(\mathbf{x}_{m,n}) &\approx \frac{1}{2\Delta y}(\phi(\mathbf{x}_{m,n+1}) - \phi(\mathbf{x}_{m,n-1})) \\ \frac{\partial^2 \phi}{\partial x \partial y}(\mathbf{x}_{m,n}) &\approx \frac{1}{4\Delta x \Delta y}(\phi(\mathbf{x}_{m+1,n+1}) - \phi(\mathbf{x}_{m-1,n+1}) \\ &\quad - \phi(\mathbf{x}_{m+1,n-1}) + \phi(\mathbf{x}_{m-1,n-1}))\end{aligned}$$

for $m = i, i + 1$ and $n = j, j + 1$. Thus, construction of the interpolant p requires all the points shown in Fig. 4.7. Higher order local approximations can be made using higher order finite difference approximations and using a larger set of grid points around the box where the interpolant is used.

Now, given the interpolating function $p(x, y)$ in the domain $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, and given a point (x_0, y_0) in that domain, we compute the distance between (x_0, y_0) and the zero level curve of $p(x, y)$. The point (x_1, y_1) on the zero level curve closest to (x_0, y_0) must satisfy two conditions:

$$p(x_1, y_1) = 0, \quad (4.27)$$

$$\nabla p(x_1, y_1) \times ((x_0, y_0) - (x_1, y_1)) = 0. \quad (4.28)$$

Equation 4.27 is a requirement that (x_1, y_1) must be on the interface. Equation 4.28 is a requirement that the interface normal, given by $\nabla p(x_1, y_1)$, must be aligned with the line through the points (x_0, y_0) and (x_1, y_1) . Equations 4.27 and 4.28 are solved simultaneously using Newton's method. Typically, less than five iterations are necessary in order to achieve sufficient accuracy.

Given the front speed $F(x_1, y_1)$ and the initial distance to the front, $d = \|(x_1, y_1) - (x_0, y_0)\|$, the initial value for a point adjacent to the initial front for the general fast marching method solving Eq. 4.21 is d/F .

4.2.4 Reinitialization

As noted in the introduction to this chapter, there are two means by which the level set method can be kept stable for arbitrary speed functions. For nearly all applications of the level set method, one of these techniques must be used. One method involves using velocity extensions, and the other uses reinitialization. Both methods are frequently used, and there is disagreement as to which method is preferred. Recent advances in the level set method have resulted in either method producing good results. For balance, both methods are presented, with reinitialization treated here and velocity extensions to follow in Section 4.2.5.

Reinitialization was first introduced in [19], where it was observed that the only part of the level set function which is of interest is the portion immediately around the zero level set. While initially, the level set function can be constructed to be the signed distance function to the interface, most speed functions, F , will not preserve this property over time. This can lead to instability, and ultimately failure of the method. Reinitialization is, therefore, a process where the level set function is reconstructed to be the signed distance function.

Let ϕ be the level set function, and let $\tilde{\phi}$ be the desired reconstructed level set function, then $\tilde{\phi}$ solves

$$\phi^{-1}(0) = \tilde{\phi}^{-1}(0), \quad (4.29)$$

$$\|\nabla \tilde{\phi}\| = 1. \quad (4.30)$$

This pair of equations is precisely the type of problem the fast marching method is designed to solve, with $F \equiv 1$ in Eq. 4.21. Furthermore, the function ϕ can be used to initialize the fast marching method, as described in Section 4.2.3. The solution $\tilde{\phi}$ of Eqs. 4.29 and 4.30 is now called reinitialized.

Early implementations of reinitialization suffered from accuracy, particularly in regions of high curvature. When the interface was reinitialized, there was significant error in the computed solution in Eq. 4.29. This was primarily due to the low-order accurate methods used for interpolating ϕ . More recent methods, such as the one presented in Section 4.2.3, significantly reduced this error, as illustrated in Fig. 4.8.

It has been observed recently [100] that for the specific application of reinitialization, it is not necessary to use the heap sort method. In fact, the same results can be achieved by simply taking a first-in-first-out strategy for the order of the grid points. In other words, instead of maintaining the binary tree and

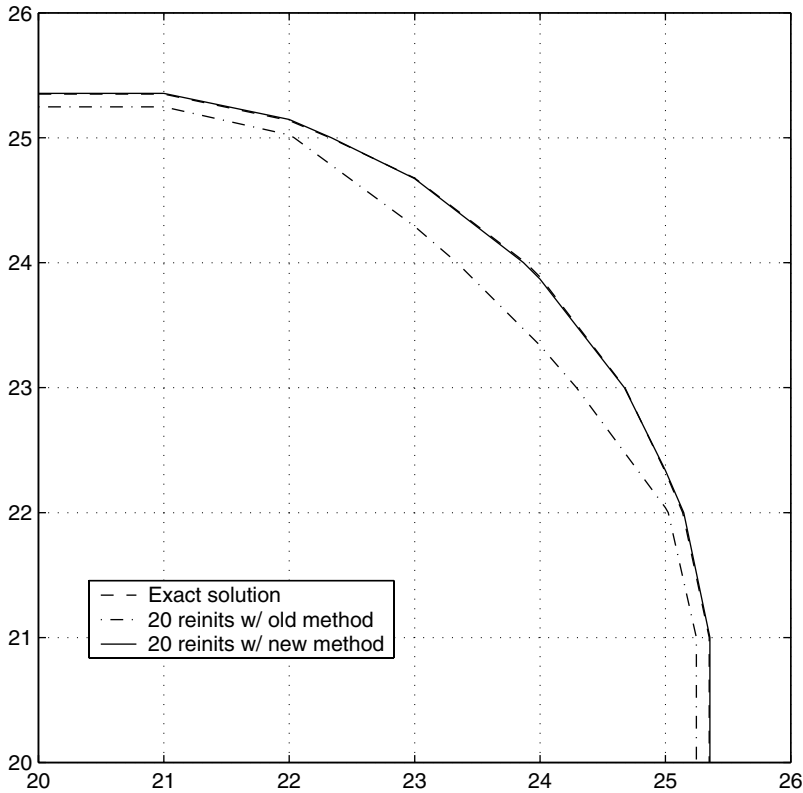


Figure 4.8: Comparison of modern and original reinitialization results for a coarsely meshed circle. The exact solution and the modern reinitialization method are nearly overlapping. This figure reprinted from [22].

continually sorting the nodes, it is sufficient to simply take points out of the set T in the same order in which they entered. The only exception is that the initial set of grid points in the set T should still start out sorted. This observation is of interest because it reduces the computational complexity of the fast marching method from $O(N \log N)$ to simply $O(N)$ where N is the total number of grid points.

4.2.5 Velocity Extensions

As noted in the previous section, reinitialization fixes the level set function when the velocity field does not preserve the level set function as a signed distance function. An alternative way to preserve the signed distance function is to adjust

the velocity field in the first place. This approach was introduced by Adalsteinsson and Sethian [3] as an alternative to the use of reinitialization. As noted in the introduction, velocity extensions also serve the purpose of constructing a velocity field for the entire domain of ϕ even when the speed, F , is defined only on the interface itself.

For velocity extensions, the objective is to force the velocity field, F , to be such that the signed distance function is preserved, i.e.

$$\nabla \phi \cdot \nabla \phi \equiv 1. \quad (4.31)$$

Differentiating Eq. 4.31 with respect to t , and using Eqs. 4.5 and 4.31, gives

$$\nabla \phi \cdot \nabla F = 0. \quad (4.32)$$

Viewed geometrically, Eq. 4.32 makes sense because it requires the speed function normal to the interface to be constant along that normal. This effectively keeps the level sets of ϕ evenly spaced.

To solve Eq. 4.32, assume the function F is given on the zero level set of ϕ . The goal is to construct an extension velocity F^{ext} , such that

$$F^{\text{ext}} \Big|_{\phi=0} = F \Big|_{\phi=0} \quad \text{and} \quad \nabla F^{\text{ext}} \cdot \nabla \phi = 0. \quad (4.33)$$

The solution of Eq. 4.33 is done in a manner very similar to the fast marching method. The discretization of Eq. 4.33 is given by

$$\begin{aligned} & \min(D_{+x}\phi_{i,j}, 0)D_{+x}F_{i,j}^{\text{ext}} + \max(D_{-x}\phi_{i,j}, 0)D_{-x}F_{i,j}^{\text{ext}} \\ & + \min(D_{+y}\phi_{i,j}, 0)D_{+y}F_{i,j}^{\text{ext}} + \max(D_{-y}\phi_{i,j}, 0)D_{-y}F_{i,j}^{\text{ext}} = 0. \end{aligned} \quad (4.34)$$

This is a linear equation in terms of the unknown $F_{i,j}^{\text{ext}}$ and is easily solved. Note that Eq. 4.34 must be solved at the grid points $\mathbf{x}_{i,j}$ in the order of increasing magnitude of $\phi_{i,j}$ similar to the fast marching method. This is easily accomplished using the same heap-sort strategy described in Section 4.2.3.

The initialization of $F_{i,j}^{\text{ext}}$ on the grid points near the interface $\phi^{-1}(0)$ is done using the bicubic interpolation method discussed in Section 4.2.3. Given a grid point $\mathbf{x}_{i,j}$, the point, $\mathbf{y}$, on the interface $\phi^{-1}(0)$ nearest to $\mathbf{x}_{i,j}$ is computed using the bicubic interpolant. The value of $F_{i,j}^{\text{ext}}$ must be the same as $F(\mathbf{y})$, because the vector $\mathbf{x}_{i,j} - \mathbf{y}$ is orthogonal to the interface, and hence parallel to $\nabla \phi$, so F^{ext} must be constant along that vector. This populates the grid points adjacent to the interface, and the velocity extension algorithm can then proceed.

The algorithm for velocity extensions is therefore given by

1. Initialize the value of F^{ext} at all the grid points adjacent to the interface $\phi^{-1}(0)$ using the bicubic interpolation algorithm discussed in Section 4.2.3. Place all those grid points in the accepted set A . Add all grid points adjacent to a point in the set A into the set T , and the remainder of the grid points are placed in the set D .
2. Take the grid point $\mathbf{x}_{i,j}$ with the smallest value of $|\phi_{i,j}|$ from the set T and place it in set A . Solve Eq. 4.34 for $F_{i,j}^{\text{ext}}$ at this $\mathbf{x}_{i,j}$. Take all neighbors of $\mathbf{x}_{i,j}$ which are in D , and place them in T .
3. Repeat step 2 while $T \neq \emptyset$.

Similar to the comments made in the previous section, the velocity extension method also can avoid the cost of the heap sort by taking the first-in-first-out strategy. Therefore, the computational cost for the velocity extension is the same as for reinitialization, $O(N)$.

4.2.6 Narrow Band Methods

There is another technique frequently used in level set methods that deserves attention. While it is not an essential part of the level set method it is useful in that it can significantly reduce the computational cost.

As noted earlier, switching from a parametric representation to the implicit representation used in the level set method also increased the computational cost. For example, if an evolving curve in the plane can be modeled with $O(N)$ points, then the corresponding level set representation would require $O(N^2)$ points, due to the higher dimension of the level set function. However, most of that increased computational cost is spent computing the evolution of ϕ in regions far from the $\phi = 0$ interface of interest.

It was observed in [19] that it is not necessary to compute the evolution of ϕ everywhere, but only in the neighborhood of the $\phi = 0$ interface. This observation effectively reduces the computation back to $O(N)$. This technique is called a narrow-band level set method, and was significantly refined in [2]. Basically, the evolution equation of ϕ is computed on a dynamically determined set of grid points where ϕ is small.

Not all applications will benefit from a narrow band implementation; it depends heavily on the cost of computing F , which can easily overwhelm the cost

of the rest of the level set method. However, if that is not the case, the reader should see [2] for a detailed description of the narrow band method.

4.2.7 Assembling the Basic Algorithm

To summarize this section, we conclude with an algorithmic description of the full level set method. The strategies of using reinitialization or velocity extensions are similar, so the algorithms will be presented together.

1. Initialize ϕ to be the signed distance function to the initial interface Γ . If necessary, reinitialization can be used to enforce this condition.
2. Compute the speed function F . This is, obviously, application dependent.
3. Determine F^{ext} from F (if using velocity extensions).
4. Advance ϕ in time using Eq. 4.5, with speed F (or F^{ext} if using velocity extensions).
5. Reinitialize (if using reinitialization).
6. Return to step 2.

Note that reinitialization does not have to be done every time step, but can be done as needed, so step 5 can be skipped most of the time. The frequency of reinitialization is application dependent. Automated detectors for determining when reinitialization is required have been proposed, but the computational cost of these detectors is of the same order of magnitude as doing reinitialization in the first place. Consequently, it is preferable to simply choose an appropriate number of time steps between each reinitialization.

On the other hand, velocity extensions must be done every time step. The common belief that the velocity extension method is more accurate was certainly true when it was first introduced. However, improvements in the reinitialization process have closed the gap. Nonetheless, for the majority of applications, the speed function F is defined only on the front, so velocity extensions will be required every step anyway, just to be able to use Eq. 4.5. In that case, it makes more sense to use velocity extensions and never do reinitialization. For the remainder of the applications where stability is a concern, it is essentially a toss-up between the two.

4.2.8 Example Calculations

There are numerous examples of applications for the level set and fast marching methods. A comprehensive list of the most recent applications will be given in Section 4.4. Two simple examples which illustrate the basic level set method are presented here.

Minimal Surfaces with Voids

As noted in the introduction, the original paper on the level set function involved a speed function which depended on mean curvature. Flow by mean curvature was also used to compute examples of minimal surfaces [19]. In the present example, flow by mean curvature is again used, but this time there are void regions where the surface area contained in the void is not counted in the total surface area.

It is shown in [123] that the minimal surface in this case will meet the voids orthogonally. The orthogonality boundary condition can be rewritten in a way that is familiar. Suppose ϕ is the evolving surface moving by mean curvature, and assume ψ is a level set function representing the voids with the surface of the voids identified by $\psi = 0$. The orthogonality boundary condition is equivalent to the surface normals being orthogonal; in other words, we must have

$$\nabla\phi \cdot \nabla\psi = 0. \quad (4.35)$$

This equation is reminiscent of the equation for velocity extensions, Eq. 4.32. In fact, the velocity extension algorithm is used to determine ϕ inside the voids.

In Fig. 4.9, a surface which passes through five spherical voids is illustrated. Initially, the surface passes over the central void. As the surface relaxes, it strikes the center sphere and finally reaches equilibrium on the lower side of the sphere. The voids are semitransparent so that the results of applying the velocity extension code to ϕ can be seen. Also, the shading on the surface indicates the magnitude of the mean curvature.

Curvature Flow in Hyperbolic Space

In [25], mean curvature flow in hyperbolic space mapped onto the upper half-space is investigated. In particular, foliations of the space are computed using a

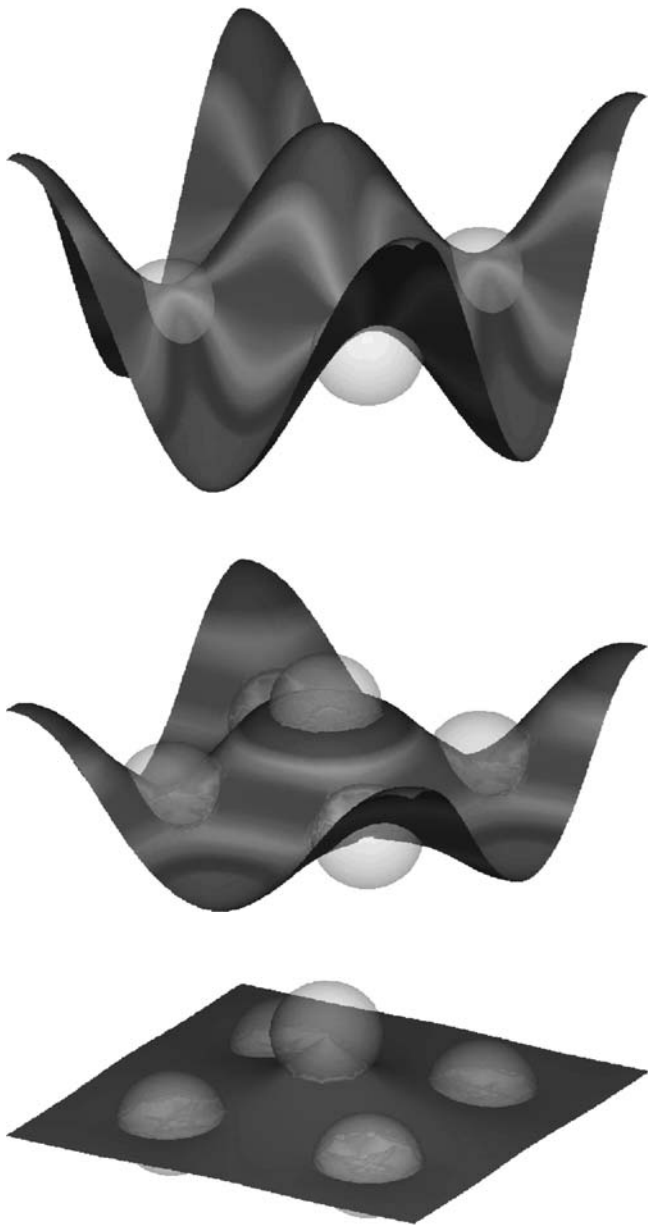


Figure 4.9: Example of curvature flow with voids.

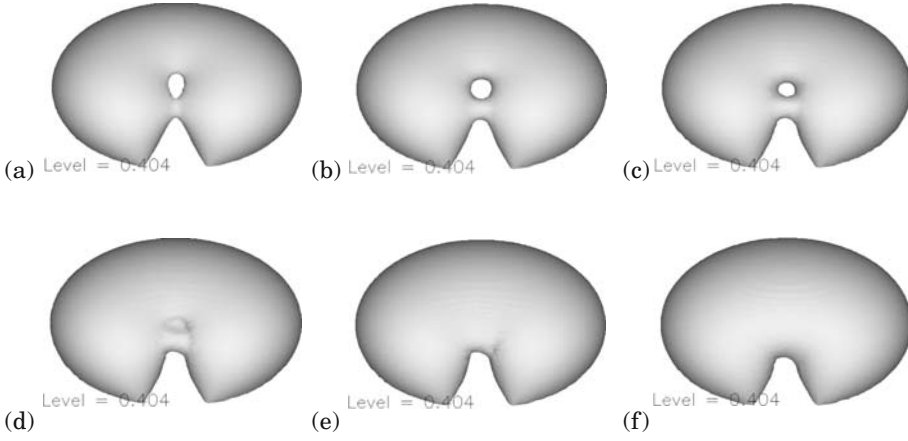


Figure 4.10: Change in topology of prescribed level for a notched annulus. Time steps are (a) $t = 0$, (b) $t = 0.05$, (c) $t = 0.1$, (d) $t = 0.15$, (e) $t = 0.2$, and (f) $t = 0.5$. Reprinted with permission from [25].

flow of the form

$$\frac{\partial \phi}{\partial t} = (\kappa - \phi) \|\nabla \phi\|, \quad (4.36)$$

with the boundary condition

$$\phi(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \text{ inside } \Gamma \\ -1 & \mathbf{x} \text{ outside } \Gamma \end{cases}, \quad \text{on the plane } z = 0, \quad (4.37)$$

where Γ is a closed curve in the $z = 0$ plane.

In Fig. 4.10, a sample evolution of one of the level curves is shown. One of the questions addressed in [25] is whether all disks in a foliation are topologically disks. In Fig. 4.10, the resolving of the topology for a particular leaf in a foliation is illustrated as it evolves in time. The numerical experiments conducted in [25] suggest that the answer is that the foliation is of disks, even for very complicated boundary curves.

4.3 Recent Developments

There have been numerous modifications and additions to the level set method since its inception. Sometimes multiple level set functions are used in tandem to produce other effects. For example, multiple level set functions are used to

represent the step edges of a stratified material [21]. Lower dimensional manifolds, such as one-dimensional curves in $\mathbb{R}^3$, can be captured by the intersection of two level surfaces [14]. Multiple distinct regions with interfaces that result in triple junctions can also be captured using multiple level set functions [17, 108]. Interfaces with boundaries, such as a finite-length crack in a plate, can also be represented using multiple level sets [107, 115, 117].

In other examples, multiple level surfaces of a single level set function are used. In [95], the level surfaces for each of the integers represent a different interface. In other applications, the entire continuous spectrum of level surfaces are used. For example, in [25], each level surface evolves to a surface of constant curvature, while in [20], the spectrum of evolving level surfaces is shifted in order to locate an unstable equilibrium surface.

The fast marching method has also been used in a variety of applications, resulting in dramatically increased speed in some computationally intensive calculations. For example, see the work on computing multiple travel-time arrivals in [46].

The range of applications for the level set and fast marching methods is now very wide, and still growing. Many times, variations of the method are required to make it fit the problem. In this section, some recent improvements and variations, which will be of general interest are presented.

4.3.1 Ordered Upwind Methods

In [101, 129], Sethian and Vladimirsky developed a novel extension of the fast marching method, making it applicable to a significantly wider class of problems.

Recall the fast marching method equation

$$F\|\nabla\phi\| = 1. \quad (4.38)$$

It is important to recognize that this equation assumes that from any point $\mathbf{x}$, the speed, F , is the same, independent of the direction the interface is traveling. In other words, the speed function is isotropic. Sethian and Vladimirsky have generalized the fast marching method so that the speed function can vary with direction, i.e. the speed function, F , may depend on $\nabla\phi$. In this case, the speed function is called anisotropic. The generalized method is called the ordered upwind method, of which the fast marching method is a special case.

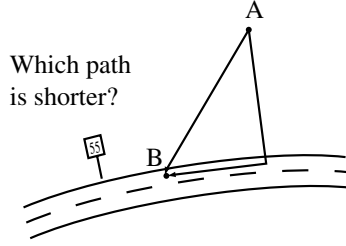


Figure 4.11: Illustration of the difficulty with anisotropic speed functions. Which path is optimal depends on the speed *and* direction.

To illustrate the difference, consider the problem of finding the fastest route between two cities. If the problem were isotropic, then it would mean that you will travel at the same speed regardless of the direction you are traveling. The solution is therefore simple: a straight line path between the two cities. However, in reality, there are roads, bridges, rivers, mountains, and other assorted terrain features that can influence the choice of the path. When on a road, the speed function depends heavily on the direction to be traveled, with the highest speed along the road and the slowest speed off the road.

The example of the road highlights one of the key technical issues that had to be addressed in this paper. In the isotropic case, when computing an estimate for the value of $\phi(\mathbf{x})$, it is sufficient to only check immediately neighboring points. In the anisotropic case, this is not the case. When standing at a point on the road, one must check not only the immediate neighborhood, but must also check far down the road to see if a shorter path along the road would be possible. This comparison is illustrated in Fig. 4.11, where the shortest path arriving at point B may not be directly from nearby points, but may come from far away points along directions which are faster. The key observation in [129] was the identification of how far away one must check to assure locating the shortest path.

More specifically, the ordered upwind method solves equations of the form

$$F(\nabla\phi, \mathbf{x})\|\nabla\phi\| = 1, \quad (4.39)$$

with the additional assumption that $F(\nabla\phi, \mathbf{x}) > 0$ is convex. The case where F is non-convex is significantly more challenging and remains an open problem.

The algorithm for the ordered upwind method is similar to the fast marching method described in Section 4.2.3, with only step 3 requiring modification. In the fast marching method, when a point $\mathbf{x}$ is moved from the tentative set, T ,

to the accepted set, A , only the immediately adjacent grid points require the approximate value of ϕ to be updated. Let $F_{\min}$ and $F_{\max}$ be the minimum and maximum values of the speed function F . For the more general ordered upwind method, all the tentative points in a radius of $\Delta x F_{\max}/F_{\min}$ around $\mathbf{x}$ must be updated. If the new approximate value for ϕ is smaller, this new value is used. This is to account for the possible highest speed direction which could allow the point $\mathbf{x}$ to influence grid points within this radius *before* the immediately adjacent grid points. The formulation for computing the approximation for ϕ at these tentative points uses the same type of one-sided discretization as used in the fast marching method to follow the characteristics from $\mathbf{x}$.

As an example of the use of the ordered upwind method, the geodesic distance from the origin on the manifold $z = \frac{3}{4} \sin(3\pi x) \sin(3\pi y)$ is computed on the square $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$ in the x - y plane. The resulting distance isocontours are shown in Fig. 4.12

4.3.2 Improved Velocity Extensions

The velocity extension method currently in common usage was described in Section 4.2.5, and can be attributed to [3]. However, as noted in [23], the velocity extension characteristics are not supposed to be the straight line extensions that are currently constructed. While it is true that $\nabla F \cdot \nabla \phi = 0$ should hold at the initial interface, it does not necessarily hold off the interface.

As an example of what can happen with the current velocity extension method, consider the example of an interface consisting of two circles, with the left circle having speed 1, and the right circle having speed 2 (see Fig. 4.13). The current velocity extension method is such that the left half-plane will have $F = 1$, and the right half-plane will have $F = 2$, with the break indicated by the dashed line in Fig. 4.13. The evolution makes a clear error when the right circle expands to the dividing line. Once the circle crosses that line, the velocity extension incorrectly changes the speed from 2 to 1. By noting the gap between successive contours, it is clear that the right-hand circle has slowed down on the left side. The reason that the velocity extension in Fig. 4.13 failed is because the characteristics of the problem were not respected. Once the interface crossed the center line, the velocity came from the left circle, while the characteristics came from the right circle. Ultimately, this happened because the velocity extension was done independent of, and prior to, the actual evolution.

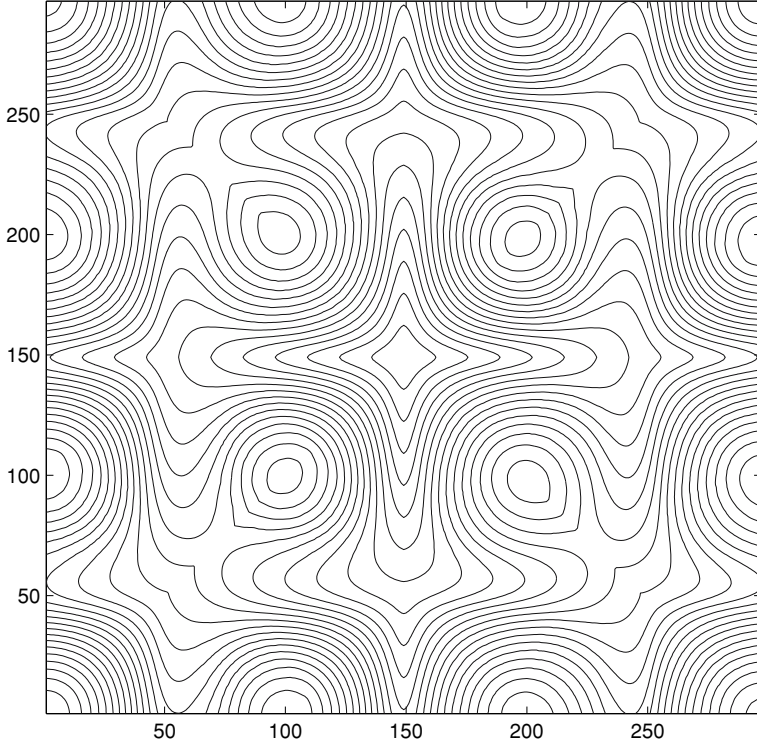


Figure 4.12: A contour map of the distance from the origin on the manifold $z = \frac{3}{4} \sin(3\pi x) \sin(3\pi y)$, computed using the ordered upwind method. Reprinted with permission from [101].

The solution is to do both the fast marching method with the velocity extension at the same time:

$$F \|\nabla \phi\| = 1, \quad (4.40)$$

$$\nabla F \cdot \nabla \phi = 0. \quad (4.41)$$

The discretization of these two equations is the same as before, but the solution method requires some explanation. Again, suppose the values of ϕ and F are already determined at $\mathbf{x}_{i-1,j}$ and $\mathbf{x}_{i,j+1}$. Then Eqs. 4.40 and 4.41 become

$$F_{i,j}^2 ((D_x^- \phi_{i,j})^2 + (D_y^+ \phi_{i,j})^2) = 1, \quad (4.42)$$

$$(D_x^- F_{i,j})(D_x^- \phi_{i,j}) + (D_y^+ F_{i,j})(D_y^+ \phi_{i,j}) = 0. \quad (4.43)$$

These equations correspond to Eqs. 4.22 and 4.34 respectively, where the unknowns are $F_{i,j}$ and $\phi_{i,j}$, and the remainder of the terms are known. This pair

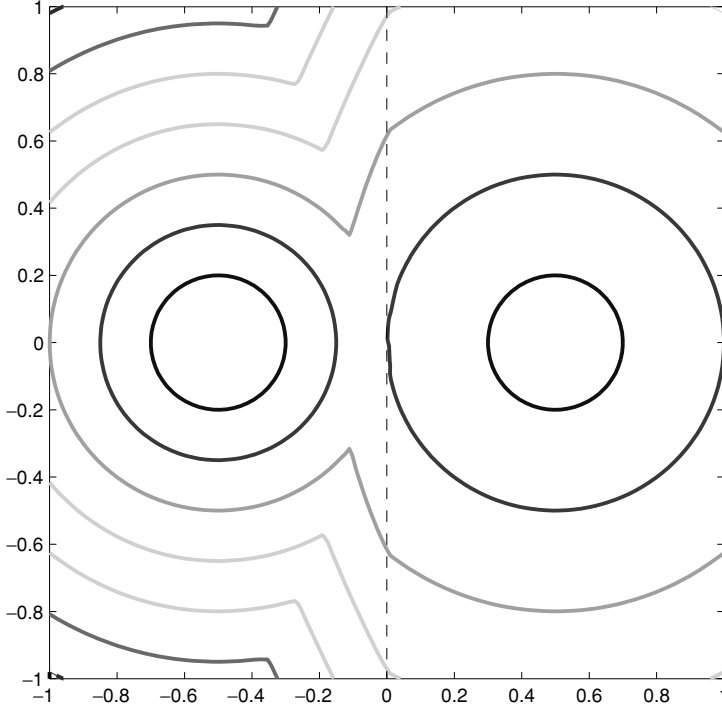


Figure 4.13: Example of two circles expanding using the current velocity extension method.

of equations results in a quartic polynomial in $F_{i,j}$ which can be solved using a Newton solver or by a direct quartic polynomial solver. Once $F_{i,j}$ is computed, the value of $\phi_{i,j}$ is easily computed from Eq. 4.43.

The initialization of this method uses a similar bicubic representation as was discussed in Section 4.2.3. The initialization process is based upon the following theorem from [23], and illustrated in Fig. 4.14:

Theorem 1. Suppose $\Gamma = \{(x, y) : ax + by = c\}$ and $F_0(x, y) = dx + ey + f$

for $(x, y) \in \Gamma$ with F_0 not identically zero on Γ , then the equations

$$F \|\nabla \phi\| = 1, \quad (4.44)$$

$$\nabla F \cdot \nabla \phi = 0, \quad (4.45)$$

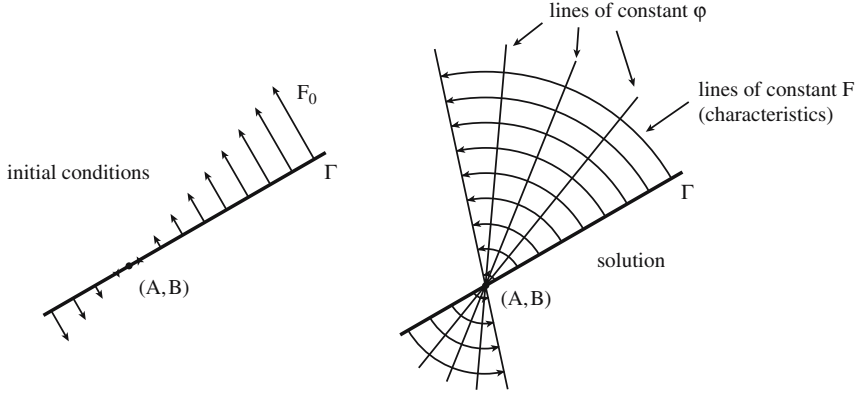


Figure 4.14: Illustration of a sample initial condition and the corresponding solution.

with $\phi(x, y) = 0$, $F(x, y) \Big|_{\Gamma} = F_0(x, y)$, have a solution of the form

$$F(x, y) = \frac{db - ea}{\sqrt{a^2 + b^2}} \sqrt{X(x, y)^2 + Y(x, y)^2}, \quad (4.46)$$

$$\phi(x, y) = \frac{\sqrt{a^2 + b^2}}{db - ea} \tan^{-1} \left(\frac{Y(x, y)}{X(x, y)} \right). \quad (4.47)$$

If $db - ea \neq 0$, where

$$X(x, y) = \frac{b}{\sqrt{a^2 + b^2}}(x - A) - \frac{a}{\sqrt{a^2 + b^2}}(y - B), \quad (4.48)$$

$$Y(x, y) = \frac{a}{\sqrt{a^2 + b^2}}(x - A) + \frac{b}{\sqrt{a^2 + b^2}}(y - B) \quad (4.49)$$

and where $A = \frac{ec+fb}{ae-bd}$, $B = \frac{af+cd}{bd-ae}$. The solution is valid in the set $\mathbb{R}^2 \setminus L$, where L is an arbitrary line passing through the point (A, B) .

If $db - ea = 0$, then $F_0(x, y) = F_0$ is constant on Γ , and the solution becomes

$$F(x, y) = F_0, \quad (4.50)$$

$$\phi(x, y) = \frac{\pm 1}{F_0 \sqrt{a^2 + b^2}} (ax + by - c), \quad (4.51)$$

valid on all $\mathbb{R}^2$.

Given an initial piece of the interface, the interface is approximated using a linear function, and also the speed, F , along the interface uses a linear

approximation. The linear approximation allows the solutions in the theorem to apply, where it is observed that the characteristics travel in circles with varying speed F , and the linear approximation of F designates a center of rotation, (A, B) , depending on where F crosses zero. This leads to a generalized form of Eq. 4.28:

$$\nabla\phi(\mathbf{y}) \times (\mathbf{x} - \mathbf{y}) = \frac{\|\mathbf{x} - \mathbf{y}\|^2 (\mathbf{k} \cdot (\nabla F(\mathbf{y}) \times \nabla\phi(\mathbf{y})))}{2F(\mathbf{y})}. \quad (4.52)$$

Note that Eq. 4.28 is recovered if $\nabla F = 0$ is assumed. Equations 4.27 and 4.52 are solved in the same manner as described in Section 4.2.3.

Using the modified velocity extension method on the earlier two-circle example produces the correct results as shown in Fig. 4.15.

Another example that illustrates the difference between the two velocity extension methods is given by an initial circle, with F varying linearly with respect to x , and near zero on the left side. The largest difference between the two methods can be seen on the side where F is small. In the old method,

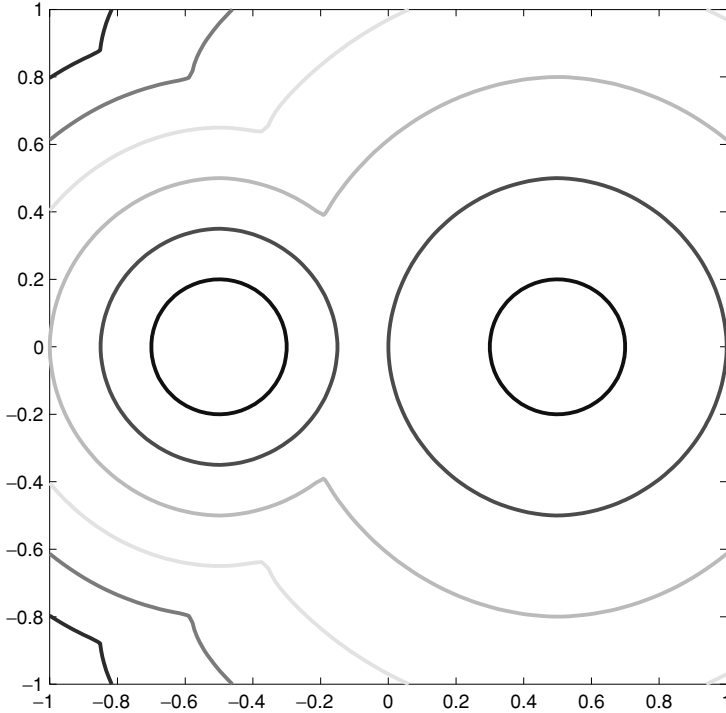


Figure 4.15: Two-circle example with the modified velocity extension method.

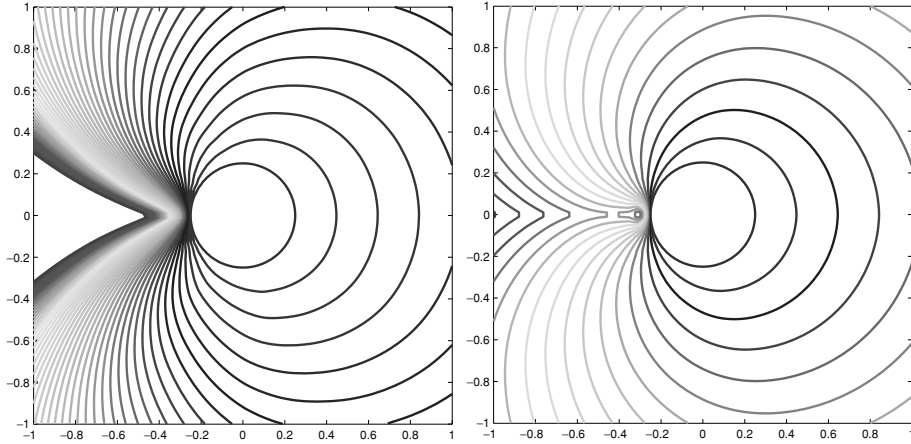


Figure 4.16: Comparison of the old (left) and new (right) velocity extension methods.

the interface slows down when it approaches the left side, while with the new method the interface wraps around and merges. The two solutions are shown side by side in Fig. 4.16. The characteristics for this example, represented by the lines of constant F , are shown in Fig. 4.17, illustrating the analogous solution as computed in the theorem. Note how the lines of constant F are orthogonal to the lines of constant ϕ , as a result of solving Eq. 4.41.

4.3.3 Coupling to Elliptic Solvers

Very often, the speed of the interface is determined by solving an associated elliptic equation, e.g. the pressure equation for incompressible fluid flow. This leads to an elliptic equation which must be solved on an irregularly shaped domain or where there is an internal boundary with jump conditions across the boundary. There are several strategies to handle this problem. When using finite elements to solve this elliptic equation, a mesh is dynamically generated so that it conforms to this irregular boundary. When using finite differences, special delta functions can be added at nodes near the interface to enforce the jump conditions, see e.g. [88].

In the context of the level set method, there are three strategies for setting up and solving the associated elliptic equation. They vary in generality,

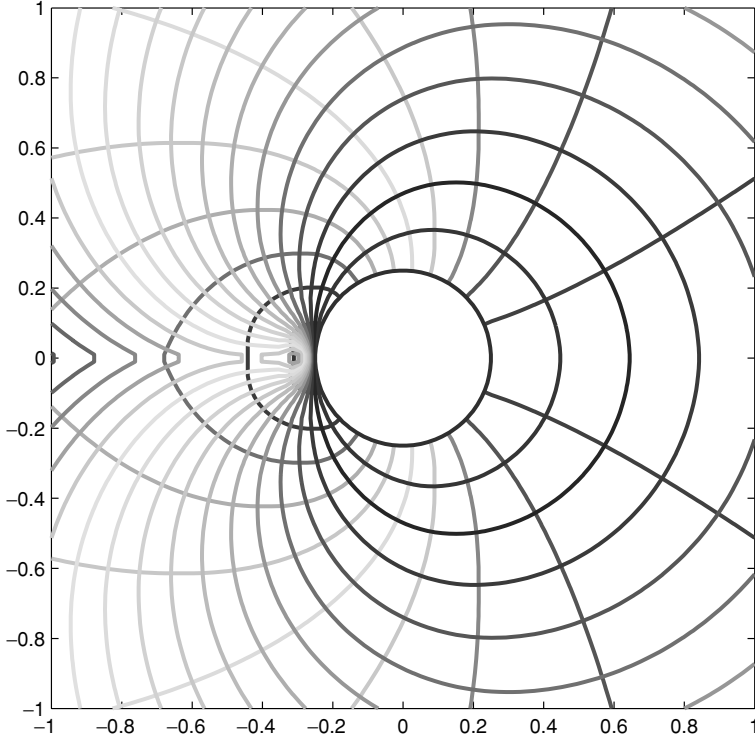


Figure 4.17: Plot of the characteristic curves along which F is constant.

complexity, and accuracy, and provide different advantages. All three strategies are presented here.

The Extended Finite Element Method

The extended finite element method (X-FEM) [29, 81, 121] is a numerical method to model internal (or external) boundaries without the need for the mesh to conform to these boundaries. The X-FEM is based on a standard Galerkin procedure and uses the concept of partition of unity [80] to accommodate the internal boundaries in the discrete model. The partition of unity method [80] generalized finite element approximations by presenting a means to embed local solutions of boundary-value problems into the finite element approximation.

For a standard finite element approximation, consider a point $\mathbf{x}$ of $\mathbb{R}^d$ that lies inside a finite element e . Denote the nodal set $N = \{n_1, n_2, \dots, n_m\}$, where m is the number of nodes of element e . The approximation for a vector-valued

function $\mathbf{u}(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ assumes the form

$$\mathbf{u}^h(\mathbf{x}) = \sum_{\substack{I \\ n_I \in \mathbf{N}}} \phi_I(\mathbf{x}) \mathbf{u}_I, \quad (\mathbf{u}_I \in \mathbb{R}^d), \quad (4.53)$$

where the functions $\phi_I(\mathbf{x})$ are the finite element basis functions and $\mathbf{u}_I$ are the weights.

The extended finite element method uses enrichment functions, extra basis functions which are sensitive to prescribed boundaries, to capture the boundary conditions and improve the solution in the neighborhood of regions which would otherwise require greater spatial resolution. Consider again a point $\mathbf{x}$ that lies inside a finite element e . The enriched approximation for the function $\mathbf{u}(\mathbf{x})$ becomes

$$\mathbf{u}^h(\mathbf{x}) = \underbrace{\sum_{\substack{I \\ n_I \in \mathbf{N}}} \phi_I(\mathbf{x}) \mathbf{u}_I}_{\text{classical}} + \underbrace{\sum_{\substack{J \\ n_J \in \mathbf{N}^g}} \phi_J(\mathbf{x}) \psi(\mathbf{x}) a_J}_{\text{enriched}}, \quad (4.54)$$

where the nodal set $\mathbf{N}^g$ consists of nodes which are on elements cut by the boundary, for example, see Fig. 4.18. In general, the choice of the enrichment function $\psi(\mathbf{x})$ that appears in Eq. 4.54 depends on the geometry, the boundary condition, and the elliptic equation being solved.

To illustrate the effectiveness of this approach, consider the following simple example. Suppose we wish to solve the radial heat equation on an annulus given

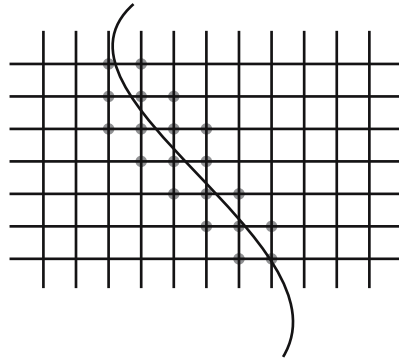


Figure 4.18: Example of choosing enriched nodes. Enriched nodes are indicated by gray dots.

by

$$u_{rr} + \frac{1}{r}u_r = 0, \quad 0 < \epsilon \leq r < L, \quad (4.55)$$

$$u_r(\epsilon) = -10, \quad u(L) = 0. \quad (4.56)$$

The exact solution is given by

$$u(r) = -10\epsilon \ln(r) + 10\epsilon \ln(L). \quad (4.57)$$

If we solve this equation for $\epsilon = 0.01$, $L = 9$ using a standard finite element method with linear elements and with nodes at $r = 0, \dots, 9$, the solution for $\epsilon \leq r < 1$ is very unsatisfactory, as shown in Fig. 4.19. However, by using a simple enrichment function $\psi_1(r) = \ln(r)$, and using this enrichment function on the first two nodes (located at $r = 0, 1$), dramatically better results are achieved (Fig. 4.19). Of course, refining the finite element mesh would also improve the results, but this requires remeshing as the interface (in this example the left boundary) moves. The X-FEM achieves this accuracy without remeshing.

The merits of coupling level sets to the extended finite element method were first explored in [118], and subsequently its advantages further realized in [53, 61, 82, 115, 117, 120]. The two methods make a natural pair of methods where:

1. Level sets provide greater ease and simplification in the representation of geometric interfaces.
2. The X-FEM, given the right enrichment functions, can accurately compute solutions of elliptic equations which are often required for computing the interface velocity.
3. Geometric computations required for evaluating the enrichment functions (such as the normal or the distance to the interface) are readily computed from the level set function [120].
4. The nodes to be enriched are easily identified using the signed distance construction of the level set function [115, 117, 118, 120].

Compared to the other methods to follow, this algorithm is more complex, but it is also much more general. Through the use of enrichment functions, this method provides a much better solution near the interface, providing subgrid resolution in that region without requiring additional mesh refinement. This is

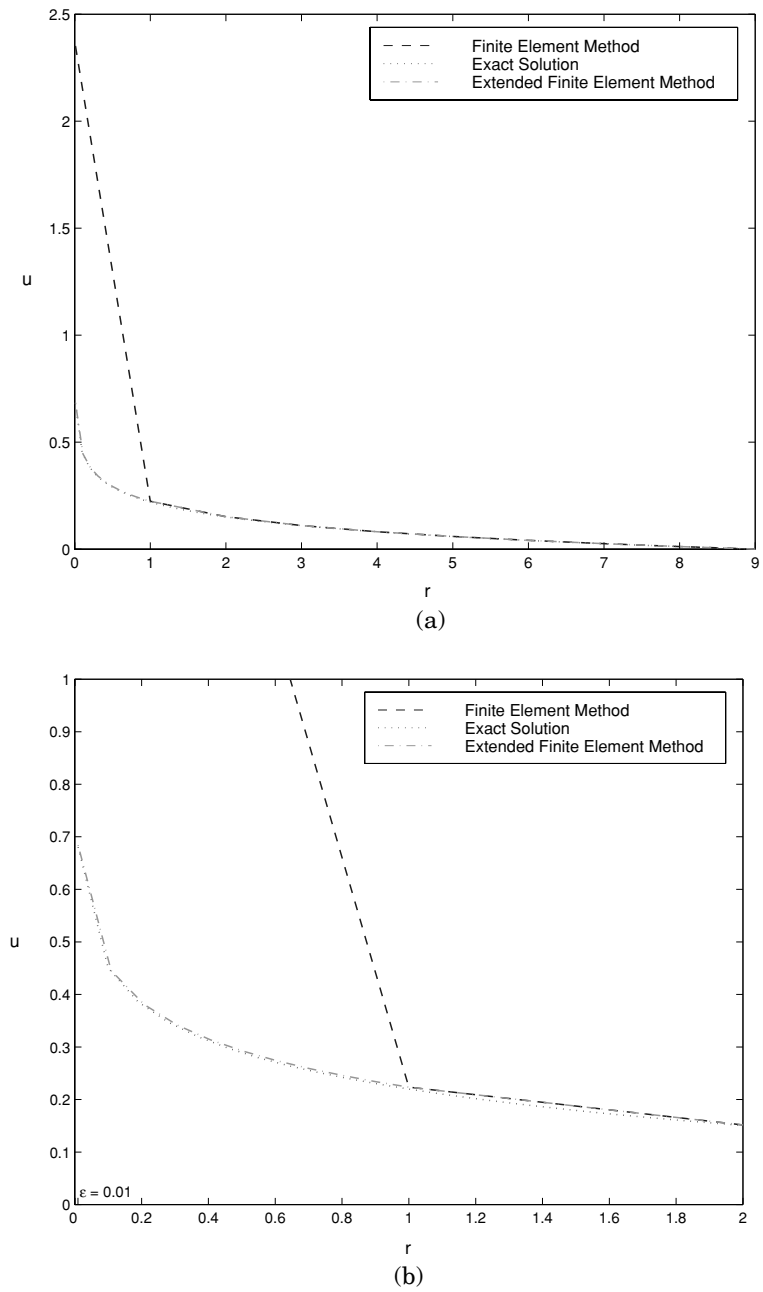


Figure 4.19: Solutions of the radial heat equation: (a) whole domain $\epsilon \leq r \leq L$ and (b) across first three nodes.

important when having to interpolate the data to determine the front speed on the boundary contour. Of the three methods, this is the only one that has this capability.

The Immersed Interface Method

The immersed interface method, introduced by LeVeque and Li [74], has also been coupled to the level set method [76, 78]. Like the X-FEM described above, the immersed interface method is designed to solve elliptic equations which arise in a variety of physical applications. The advantage of the immersed interface method is that it is second-order accurate, even near the interface where jump conditions may appear.

The immersed interface method is designed to solve equations of the form

$$\nabla \cdot (\beta(\mathbf{x})\nabla u(\mathbf{x})) + \kappa(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}), \quad (4.58)$$

where the coefficient functions β , κ , and f may have discontinuities across an interface Γ . The function f may also have a delta function singularity, which often arises, for example, from surface tension in multiphase flow.

The key idea in the immersed interface method is to modify the discretization of Eq. 4.58 in such a way that the jump discontinuities and singularities are accounted for, leading to a fully second-order method. At points away from the interface, where the coefficient functions and the solution are smooth, the standard central difference approximation is used. However, for grid points which are near the interface, an additional grid point is added to the usual central difference stencil to account for a second-order Taylor approximation around a point on the interface.

To illustrate how this method works, consider the one-dimensional problem

$$(\beta u_x)_x + \kappa u = f, \quad x \in [0, 1] \setminus \alpha, \quad (4.59)$$

$$u^+ - u^- = a, \quad \text{at } x = \alpha, \quad (4.60)$$

$$u_x^+ - u_x^- = b, \quad \text{at } x = \alpha, \quad (4.61)$$

where u^- is the value of u on the interval $[0, \alpha]$, and u^+ is the value of u on the interval $[\alpha, 1]$. Suppose that the point α is located between the uniformly spaced grid points x_i and x_{i+1} . The idea is to calculate coefficients γ_{i-1} , γ_i , γ_{i+1} , and an

additional constant, C_i , so that the approximation

$$\gamma_{i-1}u_{i-1} + \gamma_i u_i + \gamma_{i+1}u_{i+1} + \kappa_i u_i = f_i + C_i \quad (4.62)$$

is second-order accurate, with jump conditions Eqs. 4.60 and 4.61.

To determine the γ_i 's, Taylor expansions are taken about the point $x = \alpha$ to get

$$u(x_{i-1}) = u^- + (x_{i-1} - \alpha)u_x^- + \frac{1}{2}(x_{i-1} - \alpha)^2 u_{xx}^- + O(\Delta x^3), \quad (4.63)$$

$$u(x_i) = u^- + (x_i - \alpha)u_x^- + \frac{1}{2}(x_i - \alpha)^2 u_{xx}^- + O(\Delta x^3), \quad (4.64)$$

$$u(x_{i+1}) = u^+ + (x_{i+1} - \alpha)u_x^+ + \frac{1}{2}(x_{i+1} - \alpha)^2 u_{xx}^+ + O(\Delta x^3). \quad (4.65)$$

These expansions are inserted into Eq. 4.62, and the u^+ terms are eliminated from the equation by using the jump conditions Eqs. 4.60 and 4.61, combined with the equation

$$(\beta u_x^+)_x + \kappa u^+ = (\beta u_x^-)_x + \kappa u^-, \quad (4.66)$$

which comes from the continuity of f in Eq. 4.59. The function f on the right side of Eq. 4.62 is replaced with the approximation from the left side, $f = (\beta u_x^-)_x + \kappa u^-$. This results in the following equation:

$$\begin{aligned} & \gamma_{i-1} \left(u^- + (x_{i-1} - \alpha)u_x^- + \frac{1}{2}(x_{i-1} - \alpha)^2 u_{xx}^- \right) \\ & + \gamma_i \left(u^- + (x_i - \alpha)u_x^- + \frac{1}{2}(x_i - \alpha)^2 u_{xx}^- \right) \\ & + \gamma_{i+1} \left(u^- + a + (x_{i+1} - \alpha)(u_x^- + b) + \frac{1}{2}(x_{i+1} - \alpha)^2 \left(u_{xx}^- - \frac{b\beta_x - \kappa a}{\beta} \right) \right) \\ & + \kappa \left(u^- + (x_i - \alpha)u_x^- + \frac{1}{2}(x_i - \alpha)^2 u_{xx}^- \right) \\ & = \beta_x u_x^- + \beta u_{xx}^- + \kappa u^- + C_i + O(\Delta x^3) \end{aligned} \quad (4.67)$$

The coefficients γ_{i-1} , γ_i , γ_{i+1} , and C_i are now chosen so that Eq. 4.67 holds up to second order. This leads to the following equations:

$$\gamma_{i-1} + \gamma_i + \gamma_{i+1} = 0, \quad (4.68)$$

$$\gamma_{i-1}(x_{i-1} - \alpha) + \gamma_i(x_i - \alpha) + \gamma_{i+1}(x_{i+1} - \alpha) + \kappa(x_i - \alpha) = \beta_x, \quad (4.69)$$

$$\gamma_{i-1}(x_{i-1} - \alpha)^2 + \gamma_i(x_i - \alpha)^2 + \gamma_{i+1}(x_{i+1} - \alpha)^2 + \kappa(x_i - \alpha)^2 = 2\beta, \quad (4.70)$$

$$\gamma_{i+1} \left(a + b(x_{i+1} - \alpha) - \frac{1}{2} \frac{(b\beta_x - \kappa a)(x_{i+1} - \alpha)^2}{\beta} \right) = C_i. \quad (4.71)$$

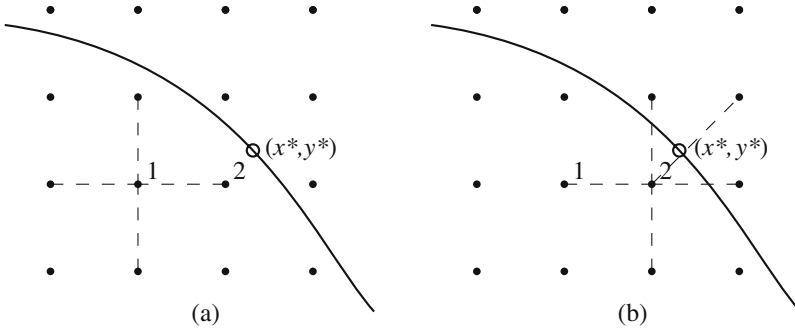


Figure 4.20: Choice of stencil for (a) points not crossed by the interface and (b) points where the interface crosses the stencil. Dashed lines indicate the points used in the stencil.

These equations are solved for γ_{i-1} , γ_i , γ_{i+1} , and C_i , thus determining the numerical approximation corresponding to the point x_i using Eq. 4.62. A similar process is followed for the approximation centered at x_{i+1} . This results in a specialized discretization at these two points and standard central difference approximations everywhere else.

For higher dimensional problems, a similar approach is taken. At grid points not crossed by the interface, the standard central difference stencil is used (see Fig. 4.20(a)) to approximate Eq. 4.58. At grid points where the interface crosses through the stencil, an additional grid point is chosen across the interface from the center of the stencil (see Fig. 4.20(b)).

When building the specialized discretization for the stencil at grid points as in Fig. 4.20(b), a point (x^*, y^*) is chosen for the point around which the approximation will be computed, and around which all Taylor expansions will be taken. Usually, the point (x^*, y^*) is the point on the interface closest to the center of the stencil (in this example, point 2). Once (x^*, y^*) is chosen, a coordinate transformation is taken so that the interface normal maps onto the x -axis. Once this coordinate transformation is completed, the computation of the stencil is similar to the one-dimensional case described above.

As noted earlier, the advantage of this method is that it is truly second-order accurate, even in the neighborhood of the interface. However, the stencil that is produced is irregular, and it sometimes can be difficult to solve the resulting linear system. Also, the choice of the points (x^*, y^*) is somewhat arbitrary, and

it is not clear what the best choices should be. Nonetheless, the method has been used successfully in a number of applications, e.g. see the review in [76].

The Ghost Point Method

The ghost point method [50] is another method designed to solve elliptic equations with irregular and moving boundaries represented by the level set method. The idea behind this method is similar to the use of what are often called *ghost points* for discretizing boundary conditions in finite difference methods. In this context, ghost points are grid points located outside the computational domain, and are used to enforce boundary conditions.

The method presented in [50] is designed to solve equations of the form

$$\nabla \cdot (\beta \nabla u) = f, \quad u|_{\partial\Omega} = g, \quad (4.72)$$

in an irregularly shaped domain Ω , where β and f are smooth functions defined on Ω , and g is defined on $\partial\Omega$, the boundary of Ω . This is a more restrictive class of problems than can be handled by the previous two methods described, but it is a class of problems that often arises. By focusing on this simpler class, a second-order method with a simple discretization can be employed, which uses a stencil that has properties which make it easier to solve numerically than the system created by the previous methods.

To illustrate this method, consider first the one-dimensional problem

$$(\beta u_x)_x = f, \quad (4.73)$$

with $\partial\Omega = x_I$, and $u(x_I) = u_I$. Assume x_I lies between the two grid points x_i and x_{i+1} . For points x_j in the interior of the domain, the central difference discretization, similar to the one used in the immersed interface method, is used:

$$\frac{1}{\Delta x} \left(\beta_{j+\frac{1}{2}} \left(\frac{u_{j+1} - u_j}{\Delta x} \right) - \beta_{j-\frac{1}{2}} \left(\frac{u_j - u_{j-1}}{\Delta x} \right) \right) = f_j. \quad (4.74)$$

At the boundary, the discretization Eq. 4.74 is again employed, but the value of u_{i+1} is not defined because x_{i+1} is outside of Ω . Instead, a ghost value for u_{i+1} is computed from the boundary condition using a linear extrapolation:

$$u_{i+1} = \frac{u_I + (\theta - 1)u_i}{\theta}, \quad \text{where } \theta = \frac{x_I - x_i}{\Delta x}. \quad (4.75)$$

For stability reasons, if $\theta < \Delta x$, then Eq. 4.75 is replaced with $u_{i+1} = u_I$. Using Eq. 4.75 in Eq. 4.74 produces the following discretization for the point near the boundary:

$$\frac{1}{\Delta x} \left(\beta_{i+\frac{1}{2}} \left(\frac{u_I - u_i}{\theta \Delta x} \right) - \beta_{i-\frac{1}{2}} \left(\frac{u_i - u_{i-1}}{\Delta x} \right) \right) = f_i. \quad (4.76)$$

In multiple dimensions, this same extrapolation technique is carried out along each coordinate direction.

The resulting discretization is only first-order accurate near the boundary, but is second-order accurate overall. This is due to the confinement of the first-order error to the nodes adjacent to the boundary. On the other hand, the linear system that comes from this discretization can be solved using faster conjugate gradient-type algorithms. Increasing the order of the extrapolation to compute u_{i+1} can result in a linear system that is more difficult to solve numerically, because of the non-symmetric stencil, and hence is not preferred.

This method is used primarily for its simplicity, while still yielding second-order convergence overall. For problems where the accuracy at the boundary is critical, this is probably not the preferred method, especially if the solution is difficult to resolve near the boundary. The method has been used in a handful of applications, for example, see [124].

Comparison of the Elliptic Equation Solvers

The algorithms presented here, for solving elliptic equations in conjunction with the level set method, vary significantly in sophistication, complexity, and capability. The X-FEM approach is by far the most difficult to construct, but is also the most general, and has the greatest potential to solve challenging problems. In particular, the X-FEM approach provides a much more accurate representation of the solution near the boundary, a property that is of critical importance when the velocity of the interface depends on this very value.

The immersed interface method and ghost point method, on the other hand, are built much more easily, and still produce accurate solutions. The immersed interface method handles a larger range of equations than does the ghost point method, which is the most restrictive in this regard. Between these two methods, the immersed interface method is more accurate at the boundary, but at the expense of a more difficult system of equations to solve numerically.

The ghost point method is probably the fastest, due to its use of faster linear solvers, but an actual direct comparison has not been done. Both the immersed interface method and ghost point method will be faster than the X-FEM approach on the same mesh. However, to obtain the same accuracy near the interface, the X-FEM will not require as fine a mesh as the others, and hence can make up the difference in time by using a coarser mesh to obtain comparable results. A direct comparison of these three methods is the subject of current research.

4.3.4 Particle Level Set Method

Another modification of the level set method, called the particle level set method, was proposed by Enright *et al.* in [38]. In the particle level set method, the level set function is compared with the motion of particles which move along the characteristics of the same velocity field. For an interface which is passively advected using the same velocity field, the particles, in theory, should not cross the interface. By comparing the motion of the particles with the moving interface, problems with the location of the interface can be identified and corrected.

Suppose the interface velocity is determined by a velocity field $\mathbf{v}(\mathbf{x}, t)$. Given this velocity, the interface speed function, F , in Eq. 4.5 is given by

$$F = \mathbf{v} \cdot \mathbf{n} = \mathbf{v} \cdot \frac{\nabla \phi}{\|\nabla \phi\|}. \quad (4.77)$$

Substituting this expression for F into Eq. 4.77 gives the passive interface advection equation

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0. \quad (4.78)$$

At the same time, the particles themselves are moving with this same velocity, $\mathbf{v}$. These two evolutions are coupled together when the particles are checked to see if any has crossed the interface, which in this case indicates that a particle has moved from a point where $\phi > 0$ to a point where $\phi < 0$, or vice versa. At that point, the level set function is “corrected.”

In [38], a large number of particles are randomly distributed uniformly in the neighborhood of the interface $\phi = 0$. Each particle, p , is assigned a sign, s_p , to indicate whether it is starting where $\phi > 0$ or $\phi < 0$, and is also assigned its distance, r_p , to the interface. As the evolution of the interface and the particles

proceeds, the particle locations are periodically checked to determine whether they have strayed across the level set function interface.

When a particle is determined to have strayed sufficiently far across the level set interface, the interface is reconstructed using the particle information. To do this, each particle, p , located at the point $\mathbf{x}_p$, is assigned a local signed distance function

$$d_p(\mathbf{x}) = s_p(r_p - \|\mathbf{x} - \mathbf{x}_p\|). \quad (4.79)$$

The level set function is now reconstructed in two steps. First, the functions ϕ^+ and ϕ^- are computed where

$$\phi^+(\mathbf{x}) = \max_{p \in P^+} d_p(\mathbf{x}), \quad (4.80)$$

$$\phi^-(\mathbf{x}) = \min_{p \in P^-} d_p(\mathbf{x}), \quad (4.81)$$

and where P^+ and P^- are the sets of points which were assigned positive and negative s_p respectively. The final ϕ function is now recovered from ϕ^+ and ϕ^- by the equation

$$\phi(\mathbf{x}) = \text{absmin}(\phi^+(\mathbf{x}), \phi^-(\mathbf{x})), \quad (4.82)$$

where

$$\text{absmin}(a, b) = \begin{cases} a, & |a| < |b| \\ b, & |b| \leq |a| \end{cases}. \quad (4.83)$$

There is no guarantee that the resulting reconstructed level set function will be a signed distance function, so if this is desired, a reinitialization step will be applied to reform ϕ into a signed distance function.

What is novel about this approach is the use of the Lagrangian and Eulerian methods to play against each other to ensure proper interface motion. However, one must carefully determine when the particle solution is correct, versus the level set evolution. This is determined by checking the local characteristics to see if they are colliding or expanding. The level set evolution tends to be better when characteristics are colliding, whereas the particle method will be more reliable when the interface is moving tangentially or stretching. Nonetheless, this combination tries to extract the positive capabilities of both the Lagrangian and Eulerian types of approaches to interface motion, while discounting the negatives.

4.4 Conclusion

The level set method has been used for a wide variety of applications and continues to be a very popular tool. Since 2001, the method has been applied to multiphase flow [7–9, 11, 16, 26, 34, 48, 49, 58, 61, 64, 72, 92, 94, 108–113, 135–138], combustion [98], granular flow [36], surfactants [1], solid mechanics [90, 119], crack propagation [53, 116, 117, 127], welding [65, 66], superconductor manufacturing [91], sintering [77], crystal growth [70, 71], Ostwald ripening and epitaxial growth [18, 37, 51, 89, 95], etching and deposition [59, 62, 63, 73, 96, 97, 130, 132], inverse scattering and shape reconstruction [15, 31, 43–45], image processing [10, 13, 27, 54, 79, 93, 99, 125, 126, 128, 134], medical imaging [30, 87, 122], shape optimization and tomography [5, 60, 86, 131], grid generation [57], bacterial biofilms [33], tissue engineering [83], and string theory [56]. The breadth of the applications is a tribute to the level set method and its creators.

In addition, the fast marching method on its own has made a contribution to a number of areas including crack propagation [24, 120], shape reconstruction [35], image processing [4, 28, 47, 52, 67, 114], medical imaging [6, 12, 32, 133], computer graphics and visualization [139], and robotic navigation [68, 69].

Despite its tremendous popularity, the level set method is not suitable for every interface propagation problem. The implicit representation of the interface can be cumbersome at times, and if the more powerful features of the level set method are not required for a given problem, then simpler methods may be more appropriate. This is especially true if the alternative methods are also faster, which can often be the case. For this reason, it is important to remember the following key distinguishing features of the level set method:

1. topological changes are handled smoothly with no user intervention required,
2. corners and cusps in the interface are handled properly by using methods borrowed from hyperbolic conservation laws,
3. the method is easily extended to higher dimensions.

Any one of these reasons may be sufficient to employ the level set method, but not every problem requires these advantages. In that case, it would serve the

practitioner to consider alternative numerical methods. It may or may not be the case that the level set method is still the best choice.

For a more comprehensive discussion on the level set method, the interested reader is directed to the books by Sethian [104] (which also includes the fast marching method) and Osher and Fedkiw [84].

Questions

1. *What are the main advantages of the level set method?*
2. *What is the importance of the connection between the level set method and hyperbolic conservation laws?*
3. *What is the difference between the level set method and the fast marching method?*
4. *Why are triple junctions a problem for the level set method?*
5. *What is the primary purpose of reinitialization, and why is it important to do it as accurately as possible?*
6. *What is the alternative to using repeated reinitializations?*
7. *What kinds of problems can be solved by the general ordered upwind method that could not be solved by the fast marching method?*
8. *What is the difference between the original velocity extension and the new velocity extension methods?*
9. *Can the level set method be implemented using the finite element method?*
10. *What is the advantage of using the X-FEM over a standard finite element formulation?*
11. *Is the level set method appropriate for all interface propagation problems?*

Bibliography

- [1] Adalsteinsson, D. and Sethian, J. A., Transport and diffusion of material quantities on propagating interfaces via level set methods, *J. Comput. Phys.*, Vol. 185, pp. 271–288, 2003.
- [2] Adalsteinsson, D. and Sethian, J. A., A fast level set method for propagating interfaces, *J. Comput. Phys.*, Vol. 118, No. 2, pp. 269–277, 1995.
- [3] Adalsteinsson, D. and Sethian, J. A., The fast construction of extension velocities in level set methods, *J. Comput. Phys.*, Vol. 48, No. 1, pp. 2–22, 1999.
- [4] Alkhalifah, T., Traveltime computation with the linearized eikonal equation for anisotropic media, *Geophys. Prospecting*, Vol. 50, pp. 373–382, 2002.
- [5] Allaire, G., Jouve, F., and Toader, A. M., A level-set method for shape optimization, *C. R. Math.*, Vol. 334, No. 1125–1130, 2002.
- [6] Antiga, L., Ene-Iordache, B., and Remuzzi, A., Computational geometry for patient-specific reconstruction and meshing of blood vessels from mr and ct angiography, *IEEE Trans. Med. Imaging*, Vol. 22, pp. 674–684, 2003.
- [7] Balabel, A., Binninger, B., Herrmann, M., and Peters, N., Calculation of droplet deformation by surface tension effects using the level set method, *Combust. Sci. Technol.*, Vol. 174, pp. 257–278, 2002.
- [8] Bassano, E., Numerical simulation of thermo-solutal-capillary migration of a dissolving drop in a cavity, *Int. J. Numer. Methods Fluids*, Vol. 41, pp. 765–788, 2003.
- [9] Bazdidi-Tehrani, F., and Zaman, S., Two-phase heat transfer on an isothermal vertical surface: a numerical simulation, *Int. J. Heat Fluid Flow*, Vol. 23, pp. 308–316, 2002.
- [10] Bertalmio, M., Cheng, L. T., Osher, S., and Sapiro, G., Variational problems and partial differential equations on implicit surfaces, *J. Comput. Phys.*, Vol. 174, pp. 759–780, 2001.

- [11] Biben, T., Misbah, C., Leyrat, A., and Verdier, C., An advected-field approach to the dynamics of fluid interfaces, *Europhys. Lett.*, Vol. 63, pp. 623–629, 2003.
- [12] Bottigli, U., and Golosio, B., Feature extraction from mammographic images using fast marching methods, *Nucl. Instrum. Methods Phys. Res. A*, Vol. 487, pp. 209–215, 2002.
- [13] Breen, D. E., and Whitaker, R. T., A level-set approach for the metamorphosis of solid models, *IEEE Trans. Visualization Comput. Graphics*, Vol. 7, pp. 173–192, 2001.
- [14] Burchard, P., Cheng, L.-T., Merriman, B., and Osher, S., Motion of curves in three spatial dimensions using a level set approach, *J. Comput. Phys.*, Vol. 170, pp. 720–741, 2001.
- [15] Burger, M., A level set method for inverse problems, *Inverse Problems*, Vol. 17, pp. 1327–1355, 2001.
- [16] Caiden, R., Fedkiw, R. P., and Anderson, C., A numerical method for two-phase flow consisting of separate compressible and incompressible regions, *J. Comput. Phys.*, Vol. 166, pp. 1–27, 2001.
- [17] Chan, T., and Vese, L., A level set algorithm for minimizing the Mumford–Shah functional in image processing. In: *IEEE Computing Society Proceedings of the 1st IEEE Workshop on “Variational and Level Set Methods in Computer Vision”*, pp. 161–168, 2001.
- [18] Chen, S., Merriman, B., Kang, M., Caffisch, R. E., Ratsch, C., Cheng, L. T., Gyure, M., Fedkiw, R. P., Anderson, C., and Osher, S., A level set method for thin film epitaxial growth, *J. Comput. Phys.*, Vol. 167, pp. 475–500, 2001.
- [19] Chopp, D. L., Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.*, Vol. 106, No. 1, pp. 77–91, 1993.
- [20] Chopp, D. L., Numerical computation of self-similar solutions for mean curvature flow, *J. Exp. Math.*, Vol. 3, No. 1, pp. 1–15, 1994.
- [21] Chopp, D. L., A level-set method for simulating island coarsening, *J. Comput. Phys.*, Vol. 162, pp. 104–122, 2000.

- [22] Chopp, D. L., Some improvements of the fast marching method, *SIAM J. Sci. Comput.*, Vol. 23, No. 1, pp. 230–244, 2001.
- [23] Chopp, D. L., The bidirectional fast marching method, Preprint, 2003.
- [24] Chopp, D. L., and Sukumar, N., Fatigue crack propagation of multiple coplanar cracks with the coupled extended finite element/fast marching method, *Int. J. Eng. Sci.*, Vol. 41, No. 8, pp. 845–869, 2003.
- [25] Chopp, D. L., and Velling, J. A., Foliations of hyperbolic space by constant mean curvature surfaces sharing ideal boundary, *J. Exp. Math.*, Vol. 12, No. 3, pp. 339–350, 2003.
- [26] Chung, M. H., A level set approach for computing solutions to inviscid compressible flow with moving solid boundary using fixed cartesian grids, *Int. J. Numer. Methods Fluids*, Vol. 36, 373–389, 2001.
- [27] Combettes, P. L. and Luo, J., An adaptive level set method for nondifferentiable constrained image recovery, *IEEE Trans. Image Processing*, Vol. 11, pp. 1295–1304, 2002.
- [28] Danielsson P. E., and Lin, Q. F., A modified fast marching method, *Image Anal., Proc.*, Vol. 2749, pp. 1154–1161, 2003.
- [29] Daux, C., Moës, N., Dolbow, J., Sukumar, N., and Belytschko, T., Arbitrary cracks and holes with the extended finite element method, *Int. J. Numer. Methods Eng.*, Vol. 48, No. 12, pp. 1741–1760, 2000.
- [30] Deng, J. W. and Tsui, H. T., A fast level set method for segmentation of low contrast noisy biomedical images, *Pattern Recognit. Lett.*, Vol. 23, pp. 161–169, 2002.
- [31] Deng, S. Z., Ito, K., and Li, Z. L., Three-dimensional elliptic solvers for interface problems and applications, *J. Comput. Phys.*, Vol. 184, pp. 215–243, 2003.
- [32] Deschamps, T. and Cohen, L. D., Fast extraction of minimal paths in 3d images and applications to virtual endoscopy, *Med. Image Anal.*, Vol. 5, pp. 281–299, 2001.

- [33] Dockery, J. and Klapper, I., Finger formation in biofilm layers, *SIAM J. Appl. Math.*, Vol. 62, pp. 853–869, 2002.
- [34] Du, Q., Li, D. Z., Li, Y. Y., Li, R., and Zhang, P. W., Simulating a double casting technique using level set method, *Comput. Mater. Sci.*, Vol. 22, pp. 200–212, 2001.
- [35] Elad, A. and Kimmel, R., On bending invariant signatures for surfaces, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 25, pp. 1285–1295, 2003.
- [36] Elperin, T. and Vikhansky, A., Variational model of granular flow in a three-dimensional rotating container, *Physica A*, Vol. 303, pp. 48–56, 2002.
- [37] Emmerich, H., Modeling elastic effects in epitaxial growth—stress induced instabilities of epitaxially grown surfaces, *Contin. Mech. Thermodyn.*, Vol. 15, pp. 197–215, 2003.
- [38] Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I., A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.*, Vol. 183, No. 1, pp. 83–116, 2002.
- [39] Evans, L.C. and Spruck, J., Motion of level sets by mean curvature i, *J. Differ. Geom.*, Vol. 33, p. 635, 1991.
- [40] Evans, L. C. and Spruck, J., Motion of level sets by mean curvature ii, *Trans. Am. Math. Soc.*, Vol. 330, No. 1, pp. 321–332, 1992.
- [41] Evans, L. C. and Spruck, J., Motion of level sets by mean curvature iii, *J. Geom. Anal.*, Vol. 2, pp. 121–150, 1992.
- [42] Evans, L. C. and Spruck, J., Motion of level sets by mean curvature iv, *J. Geom. Anal.*, Vol. 5, No. 1, pp. 77–114, 1995.
- [43] Ferraye, R., Dauvignac, J. Y., and Pichot, C., A boundary-oriented inverse scattering method based on contour deformations by means of level sets for radar imaging, *Int. J. Appl. Electromag. Mech.*, Vol. 15, pp. 213–218, 2001.
- [44] Ferraye, R., Dauvignac, J. Y., and Pichot, C., An inverse scattering method based on contour deformations by means of a level set method

- using frequency hopping technique, *IEEE Trans. Antennas Propag.*, Vol. 51, pp. 1100–1113, 2003.
- [45] Ferraye, R., Dauvignac, J. Y., and Pichot, C., Reconstruction of complex and multiple shape object contours using a level set method, *J. Electromagn. Waves Appl.*, Vol. 17, pp. 153–181, 2003.
- [46] Fomel, S. and Sethian, J. A., Fast-phase space computation of multiple arrivals, *Proc. Nat. Acad. Sci.*, Vol. 99, No. 11, pp. 7329–7334, 2002.
- [47] Frenkel, M. and Basri, R., Curve matching using the fast marching method, *Energy Minimization Methods Comput. Vision Pattern Recognition, Proc.*, Vol. 2683, pp. 35–51, 2003.
- [48] Fukano, T. and Inatomi, T., Analysis of liquid film formation in a horizontal annular flow by dns, *Int. J. Multiphase Flow*, Vol. 29, pp. 1413–1430, 2003.
- [49] Geurts, B. J., Mixing efficiency in turbulent shear layers, *J. Turbulence*, Vol. 2, pp. 1–24, 2001.
- [50] Gibou, F., Fedkiw, R., Cheng, L.-T., and Kang, M., A second order accurate symmetric discretization of the Poisson equation on irregular domains, *J. Comput. Phys.*, Vol. 176, pp. 1–23, 2002.
- [51] Gibou, F., Ratsch, C., and Caffisch, R., Capture numbers in rate equations and scaling laws for epitaxial growth, *Phys. Rev. B*, Vol. 67, pp. 155403–155406, 2003.
- [52] Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M., Fast geodesic active contours, *IEEE Trans. Image Process.*, Vol. 10, pp. 1467–1475, 2001.
- [53] Gravouil, A., Moes, N., and Belytschko, T., Non-planar 3d crack growth by the extended finite element and level sets. Part ii Level set update, *Int. J. Numer. Methods Eng.*, Vol. 53, No. 11, pp. 2569–2586, 2002.
- [54] Han, X., Xu, C. Y., and Prince, J. L., A topology preserving level set method for geometric deformable models, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 25, pp. 755–768, 2003.

- [55] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., Uniformly high order accurate essentially non-oscillatory schemes. iii, *J. Comput. Phys.*, Vol. 71, No. 2, pp. 231–303, 1987.
- [56] Hindmarsh, M., Level set method for the evolution of defect and brane networks, *Phys. Rev. D*, Vol. 68, pp. 043510–043529, 2003.
- [57] Hoch, P. and Rascle, M., Hamilton-Jacobi equations on a manifold and applications to grid generation or refinement, *SIAM J. Sci. Comput.*, Vol. 23, pp. 2055–2073, 2002.
- [58] Hunter, J. K., Li, Z. L., and Zhao, H. K., Reactive autophobic spreading of drops, *J. Comput. Phys.*, Vol. 183, pp. 335–366, 2002.
- [59] Hwang, H. H., Meyyappan, M., Mathad, G. S., and Ranade, R., Simulations and experiments of etching of silicon in hbr plasmas for high aspect ratio features, *J. Vacuum Sci. Technol. B*, Vol. 20, pp. 2199–2205, 2002.
- [60] Ito, K., Kunisch, K., and Li, Z. L., Level-set function approach to an inverse interface problem, *Inverse Problems*, Vol. 17, pp. 1225–1242, 2001.
- [61] Ji, H., Chopp, D., and Dolbow, J. E., A hybrid extended finite element/level set method for modeling phase transformations, *Int. J. Numer. Methods Eng.*, Vol. 54, No. 8, pp. 1209–1233, 2002.
- [62] Khenner, M., Averbuch, A., Israeli, M., and Nathan, M., Numerical simulation of grain-boundary grooving by level set method, *J. Comput. Phys.*, Vol. 170, pp. 764–784, 2001.
- [63] Khenner, M., Averbuch, A., Israeli, M., Nathan, M., and Glickman, E., Level set modeling of transient electromigration grooving, *Comput. Mater. Sci.*, Vol. 20, pp. 235–250, 2001.
- [64] Ki, H., Mohanty, P. S., and Mazumder, J., Modelling of high-density laser-material interaction using fast level set method, *J. Phys. D: Appl. Phys.*, Vol. 34, pp. 364–372, 2001.
- [65] Ki, H., Mohanty, P. S., and Mazumder, J., Modeling of laser keyhole welding: Part ii Simulation of keyhole evolution, velocity, temperature

- profile, and experimental verification, *Metall. Mater. Trans., A*, Vol. 33, pp. 1831–1842, 2002.
- [66] Ki, H., Mohanty, P. S., and Mazumder, J., Multiple reflection and its influence on keyhole evolution, *J. Laser Appl.*, Vol. 14, pp. 39–45, 2002.
- [67] Kimmel, R., and Sethian, J. A., Optimal algorithm for shape from shading and path planning, *J. Math. Imaging Vision*, Vol. 14, pp. 237–244, 2001.
- [68] Kobayashi, K., and Sugihara, K., Approximation of multiplicatively weighted crystal growth Voronoi diagram and its application, *Electron. Commun. Japan* 3, Vol. 85, pp. 21–31, 2002.
- [69] Kobayashi, K., and Sugihara, K., Crystal voronoi diagram and its applications, *Future Gener. Comput. Syst.*, Vol. 18, pp. 681–692, 2002.
- [70] Kohno, H., and Tanahashi, T., Finite element simulation of single crystal growth process using gsmac method, *J. Comput. Appl. Math.*, Vol. 149, pp. 359–371, 2002.
- [71] Kohno, H. and Tanahashi, T., Three-dimensional gsmac-fem simulations of the deformation process and the flow structure in the floating zone method, *J. Cryst. Growth*, Vol. 237, pp. 1870–1875, 2002.
- [72] Koren, B., Lewis, M. R., van Brummelen, E. H., and van Leer, B., Riemann-problem and level-set approaches for homentropic two-fluid flow computations, *J. Comput. Phys.*, Vol. 181, pp. 654–674, 2002.
- [73] La Magna, A., D'Arrigo, G., Garozzo, G., and Spinella, C., Computational analysis of etched profile evolution for the derivation of 2d dopant density maps in silicon, *Mater. Sci. Eng. B*, Vol. 102, pp. 43–48, 2003.
- [74] LeVeque, R., and Li, Z., The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.*, Vol. 31, pp. 1019–1044, 1994.
- [75] LeVeque, R. J., *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, Basel, 1990.

- [76] Li, Z. L., An overview of the immersed interface method and its applications, *Taiwanese J. Math.*, Vol. 7, No. 1, pp. 1–49, 2003.
- [77] Li, Z. L. and Cai, W., A level set-boundary element method for simulation of dynamic powder consolidation of metals, *Numer. Anal. Appl.*, Vol. 1988, pp. 527–534, 2001.
- [78] Li, Z. L., Zhao, H. K., and Gao, H. J., A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid, *J. Comput. Phys.*, Vol. 152, No. 1, pp. 281–304, 1999.
- [79] Liu, F., Luo, Y. P., and Hu, D. C., Adaptive level set image segmentation using the Mumford and Shah functional, *Opt. Eng.*, Vol. 41, pp. 3002–3003, 2002.
- [80] Melenk, J. M., and Babuška, I., The partition of unity finite element method: Basic theory and applications, *Comput. Meth. Appl. Mech. Eng.*, Vol. 139, pp. 289–314, 1996.
- [81] Moës, N., Dolbow, J., and Belytschko, T., A finite element method for crack growth without remeshing, *Int. J. Numer. Methods Eng.*, Vol. 46, No. 1, pp. 131–150, 1999.
- [82] Moës, N., Gravouil, A., and Belytschko, T., Non-planar 3d crack growth by the extended finite element and the level sets. Part I: Mechanical model, *Int. J. Numer. Methods Eng.*, Vol. 53, No. 11, pp. 2549–2568, 2002.
- [83] Nishimura, I., Garrell, R. L., Hedrick, M., Iida, K., Osher, S., and Wu, B., Precursor tissue analogs as a tissue-engineering strategy, *Tissue Eng.*, Vol. 9, pp. S77–S89, 2003.
- [84] Osher S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer Verlag, Heidelberg, 2002.
- [85] Osher S. and Sethian, J. A., Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.*, Vol. 79, No. 1, pp. 12–49, 1988.
- [86] Osher, S. J. and Santosa, F., Level set methods for optimization problems involving geometry and constraints. i: Frequencies of a

- two-density inhomogeneous drum, *J. Comput. Phys.*, Vol. 171, pp. 272–288, 2001.
- [87] Paragios, N., A variational approach for the segmentation of the left ventricle in cardiac image analysis, *Int. J. Comput. Vision*, Vol. 50, pp. 345–362, 2002.
- [88] Peskin, C. S., Numerical analysis of blood flow in the heart, *J. Comput. Phys.*, Vol. 25, pp. 220–252, 1977.
- [89] Petersen, M., Zangwill, A., and Ratsch, C., Homoepitaxial Ostwald ripening, *Surf. Sci.*, Vol. 536, pp. 55–60, 2003.
- [90] Phan, A. V., Kaplan, T., Gray, L. J., Adalsteinsson, D., Sethian, J. A., Barvosa-Carter, W., and Aziz, M. J., Modelling a growth instability in a stressed solid, *Modelling Simul. Mater. Sci. Eng.*, Vol. 9, pp. 309–325, 2001.
- [91] Picaud, V., Hiebel, P., and Kauffmann, J. M., Superconducting coils quench simulation, the Wilson’s method revisited, *IEEE Trans. Magnetics*, Vol. 38, pp. 1253–1256, 2002.
- [92] Pillapakam, S. B. and Singh, P., A level-set method for computing solutions to viscoelastic two-phase flow, *J. Comput. Phys.*, Vol. 174, pp. 552–578, 2001.
- [93] Preusser, T., and Rumpf, M., A level set method for anisotropic geometric diffusion in 3d image processing, *SIAM J. Appl. Math.*, Vol. 62, pp. 1772–1793, 2002.
- [94] Quecedo, M., and Pastor, M., Application of the level set method to the finite element solution of two-phase flows, *Int. J. Numer. Methods Eng.*, Vol. 50, pp. 645–663, 2001.
- [95] Ratsch, C., Gyure, M. F., Caffisch, R. E., Gibou, F., Petersen, M., Kang, M., Garcia, J., and Vvedensky, D. D., Level-set method for island dynamics in epitaxial growth, *Phys. Rev. B*, Vol. 65, pp. 195403–195415, 2002.
- [96] Ratsch, C., Kang, M., and Caffisch, R. E., Atomic size effects in continuum modeling, *Phys. Rev. E*, Vol. 6402, pp. 020601–020604, 2001.

- [97] Richards, D. F., Bloomfield, M. O., Sen, S., and Cale, T. S., Extension velocities for level set based surface profile evolution, *J. Vac. Sci. Technol. A*, Vol. 19, pp. 1630–1635, 2001.
- [98] Schmidt, H. and Klein, R., A generalized level-set/in-cell-reconstruction approach for accelerating turbulent premixed flames, *Combust. Theory Modelling*, Vol. 7, pp. 243–267, 2003.
- [99] Schupp, S., Elmoataz, A., Fadili, M. J., and Bloyet, D., Fast statistical level sets image segmentation for biomedical applications, *Scale-Space Morphology Computer Vision, Proc.*, Vol. 2106, pp. 380–388, 2001.
- [100] Sethian, J. A., Personal communication, 2002,
- [101] Sethian, J. A. and Vladimirovsky, A., Ordered upwind methods for static Hamilton–Jacobi equations, *Proc. Nat. Acad. Sci.*, Vol. 98, No. 20, pp. 11069–11074, 2001.
- [102] Sethian, J.A., Curvature and the evolution of fronts, *Commun. Math. Phys.*, Vol. 101, pp. 487–499, 1985.
- [103] Sethian, J. A., Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws, *J. Differ. Geom.*, Vol. 31, pp. 131–161, 1990.
- [104] Sethian, J.A., *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Science*, Cambridge University Press, Cambridge, 1996.
- [105] Sethian, J.A., A marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci.*, Vol., 93, No. 4, pp. 1591–1595, 1996.
- [106] Sethian, J.A., Fast marching methods, *SIAM Rev.*, Vol. 41, No. 2, pp. 199–235, 1999.
- [107] Smereka, P., Spiral crystal growth, *Physica D*, Vol. 138, pp. 282–301, 2000.
- [108] Smith, K. A., Solis, F. J., and Chopp, D. L., A projection method for motion of triple junctions by level sets, *Interfaces Free Bounda.*, Vol. 4, No. 3, pp. 263–276, 2002.

- [109] Son, G., Efficient implementation of a coupled level-set and volume-of-fluid method for three-dimensional incompressible two-phase flows, *Numer. Heat Transfer*, Vol. 43, pp. 549–565, 2003.
- [110] Son, G. and Hur, N., A coupled level set and volume-of-fluid method for the buoyancy-driven motion of fluid particles, *Numer. Heat Transfer B*, Vol., 42, pp. 523–542, 2002.
- [111] Son, G., Ramanujapu, N., and Dhir, V. K., Numerical simulation of bubble merger process on a single nucleation site during pool nucleate boiling, *Trans. ASME, J. Heat Transfer*, Vol. 124 pp. 51–62, 2002.
- [112] Son, G. H., A numerical method for bubble motion with phase change, *Numer. Heat Transfer, B*, Vol. 39, pp. 509–523, 2001.
- [113] Son, G. H., Numerical study on a sliding bubble during nucleate boiling, *Ksme Int. J.*, Vol. 15, pp. 931–940, 2001.
- [114] Spira, A., Kimmel, R., and Sochen, N., Efficient Beltrami flow using a short time kernel, *Scale Space Methods Comput. Vision, Proc.*, Vol. 2695, pp. 511–522, 2003.
- [115] Stolarska, M., and Chopp, D. L., Modeling spiral cracking due to thermal cycling in integrated circuits, *Int. J. of Eng. Sci.*, Vol. 41, No. 20, pp. 2381–2410, 2003.
- [116] Stolarska, M., and Chopp, D. L., Modeling thermal fatigue cracking in integrated circuits by level sets and the extended finite element method, *Int. J. Eng. Sci.*, Vol. 41, pp. 2381–2410, 2003.
- [117] Stolarska, M., Chopp, D. L., Moës N., and Belytschko, T., Modelling crack growth by level sets in the extended finite element method, *Int. J. Numer. Methods Eng.*, Vol. 51, No. 8, pp. 943–960, 2001.
- [118] Sukumar, N., Chopp, D. L., Moës N., and Belytschko, T., Modeling holes and inclusions by level sets in the extended finite element method, *Comput. Methods Appl. Mech. and Eng.*, Vol. 190, No. 46–47, pp. 6183–6200, 2001.
- [119] Sukumar, N., Chopp, D. L., Moes, N., and Belytschko, T., Modeling holes and inclusions by level sets in the extended finite-element

- method, *Comput. Methods Appl. Mech. Eng.*, Vol. 190, pp. 6183–6200, 2001.
- [120] Sukumar, N., Chopp, D. L., and Moran, B., Extended finite element method and fast marching method for three-dimensional fatigue crack propagation, *Eng. Fracture Mech.*, Vol. 70, No. 1, pp. 29–48, 2003.
- [121] Sukumar, N., Moës, N., Moran, B., and Belytschko, T., Extended finite element method for three-dimensional crack modeling, *Int. J. Numer. Methods Eng.*, Vol. 48, No. 11, pp. 1549–1570, 2000.
- [122] Suri, J. S., Two-dimensional fast magnetic resonance brain segmentation, *IEEE Eng. Med. Biol. Mag.*, Vol. 20, pp. 84–95, 2001.
- [123] Torres, M., Chopp, D. L., and Walsh, T., Level set methods to compute minimal surfaces in a medium with exclusions (voids), *Interfaces and Free Boundaries*, 2004, to appear.
- [124] Udaykumar, H. S. and Mao, L., Sharp-interface simulation of dendritic solidification of solutions, *Int. J. Heat Mass Transfer*, Vol. 45, No. 24, pp. 4793–4808, 2002.
- [125] Vemuri, B. C., Guo, Y. L., and Wang, Z. Z., Deformable pedal curves and surfaces: Hybrid geometric active models for shape recovery, *Int. J. Comput. Vision*, Vol. 44, pp. 137–155, 2001.
- [126] Vemuri, B. C., Ye, J., Chen, Y., and Leonard, C. M., Image registration via level-set motion: Applications to atlas-based segmentation, *Med. Image Anal.*, Vol. 7, pp. 1–20, 2003.
- [127] Ventura, G., Xu, J. X., and Belytschko, T., A vector level set method and new discontinuity approximations for crack growth by EFG, *Int. J. Numer. Methods Eng.*, Vol. 54, pp. 923–944, 2002.
- [128] Vese, L. A. and Chan, T. F., A multiphase level set framework for image segmentation using the Mumford and Shah model, *Int. J. Comput. Vision*, Vol. 50, pp. 271–293, 2002.
- [129] Vladimirovsky, A., Fast Methods for Static Hamilton–Jacobi Partial Differential Equations, Ph.D. Thesis, Univ. of California, Berkeley, 2001.

- [130] Vogl, P., Hansen, U., and Fiorentini, V., Multiscale approaches for metal thin film growth, *Comput. Mater. Sci.*, Vol. 24, pp. 58–65, 2002.
- [131] Wang, M. Y., Wang, X. M., and Guo, D. M., A level set method for structural topology optimization, *Comput. Methods Appl. Mech. Eng.*, Vol. 192, pp. 227–246, 2003.
- [132] Wheeler, D., Josell, D., and Moffat, T. P., Modeling superconformal electrodeposition using the level set method, *J. Electrochem. Soc.*, Vol. 150, pp. C302–C310, 2003.
- [133] Yan, J. Y. and Zhuang, T. G., Applying improved fast marching method to endocardial boundary detection in echocardiographic images, *Pattern Recognit. Lett.*, Vol. 24, pp. 2777–2784, 2003.
- [134] Ye, J. C., A self-referencing level-set method for image reconstruction from sparse Fourier samples, *Int. J. Comput. Vision*, Vol. 50, pp. 253–270, 2002.
- [135] Yokoi, K., Numerical method for complex moving boundary problems in a Cartesian fixed grid, *Phys. Rev. E*, Vol. 65, pp. 055701–055705, 2002.
- [136] Yokoi, K., Numerical method for a moving solid object in flows, *Phys. Rev. E*, Vol. 67, pp. 045701–045704, 2003.
- [137] Yokoi, K., and Xiao, F., Mechanism of structure formation in circular hydraulic jumps: Numerical studies of strongly deformed free-surface shallow flows, *Physica. D*, Vol. 161, pp. 202–219, 2002.
- [138] Yue, W. S., Lin, C. L., and Patel, V. C., Numerical simulation of unsteady multidimensional free surface motions by level set method, *Int. J. Numer. Methods Fluids*, Vol. 42, pp. 853–884, 2003.
- [139] Zigelman, G., Kimmel, R., and Kiryati, N., Texture mapping using surface flattening via multidimensional scaling, *IEEE Trans. Vis. Comput. Graphics*, Vol. 8, pp. 198–207, 2002.

Chapter 5

Shape From Shading Models

Xiaoping Shen¹ and Lin Yang²

5.1 Introduction

In many applications, for instance, visual inspection in robot vision and autonomous land vehicle navigation to name a few, a procedure of recovering three-dimensional surfaces of unknown objects is of considerable practical interest. In this chapter, we consider one of the reconstruction models: the shape from shading (SFS) model. The SFS models are not only important for applications in engineering but also of great intrinsic mathematical interest. We begin with a portrait of the model.

5.1.1 The Shape from Shading Model

The problem of SFS is to determine the shape of a surface, given a *single* gray level image of the surface. Mathematically speaking, if we denote the surface of the object by

$$Z = Z(x, y), \quad (x, y) \in \Omega$$

with the unit normal to the surface

$$\vec{N} = \frac{1}{\sqrt{1 + p^2 + q^2}}(-p, -q, 1)^T,$$

¹Department of Mathematics, Ohio University, Athens, OH 45701, USA

²Department of EECS, Ohio University, Athens, OH 45701, USA

where

$$(p, q) = \left(\frac{\partial Z}{\partial x}, \frac{\partial Z}{\partial y} \right) \quad (5.1)$$

is the gradient field, the image irradiance (intensity function) of the surface $I(x, y)$ and the reflectance map $R(p, q)$ are related by the following image irradiance equation [29] (p. 218):

$$I(x, y) = R(p, q). \quad (5.2)$$

The reflectance map $R(p, q)$ depends on the reflectance properties of the surface and the distribution of the light sources. It could be linear or nonlinear. An SFS problem is classified as a linear shape from shading problem if the reflectance map is linear or otherwise it is a nonlinear shape from shading problem. For instance, the one commonly used to model the lunar surface—Maria of the moon—is linear:

$$R(p, q) = \rho \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2}}, \quad (5.3)$$

where ρ , the surface albedo, and

$$\vec{s}_0 = \frac{1}{\sqrt{1 + p_0^2 + q_0^2}}(p_0, q_0, -1)^T,$$

the light source direction, are given. Solving the surface Z from (5.3) is a linear shape from shading problem.

Equation (5.2) is sometimes called the Horn image irradiance equation since it was first derived by Horn in 1970 in his thesis [26]. We would like to point out that since Eq. (5.2) depends only on the partial derivatives (p, q) of the surface $Z(x, y)$, therefore without additional conditions, the uniqueness of the solution is obviously not possible. These additional conditions are usually given by the boundary conditions. Boundary conditions can be given in many different ways; as an example, we consider the system

$$\rho \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2}} = I(x, y), \quad (x, y) \in \Omega,$$

where $\Omega = [0, 1] \times [0, 1]$ with boundary conditions:

$$\begin{aligned} (i) \quad & Z(0, y) = g_1(y), \quad 0 \leq y \leq 1, \\ (ii) \quad & Z(1, y) = g_2(y), \quad 0 \leq y \leq 1, \\ (iii) \quad & Z(x, 0) = g_3(x), \quad 0 \leq x \leq 1. \end{aligned} \quad (5.4)$$

Here g_i , $i = 1, 2, 3$, are smooth functions.

An ideal Lambertian surface is one that appears equally bright from all viewing directions and reflects all incident light, absorbing none ([29], p. 212). One of the most interesting properties of a Lambertian surface is that the maximum point of reflectance map is unique if it exists [51]. Assuming that the object has a Lambertian surface and is illuminated by a planar wave of light, the Lambertian reflectance map becomes

$$R(p, q) = \vec{N} \cdot \vec{s},$$

where $\vec{s}$ is the unit vector pointing to the light source, which is given.

A nonlinear shape from shading model is given by an ideal Lambertian surface. In this case, the reflectance map has the well-known form:

$$R(p, q)(x, y) = \rho \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2} \sqrt{1 + p^2 + q^2}}. \quad (5.5)$$

In a stereographic coordinate system, the stereographic coordinate (f, g) is related to the Cartesian coordinate by

$$f = \frac{2p}{1 + \sqrt{1 + p^2 + q^2}} \quad \text{and} \quad g = \frac{2q}{1 + \sqrt{1 + p^2 + q^2}},$$

or conversely

$$p = \frac{4f}{4 - f^2 - g^2} \quad \text{and} \quad q = \frac{4g}{4 - f^2 - g^2}.$$

In such a coordinate system, instead of using (p, q) , the reflectance map becomes

$$R(f, g) = \frac{4 - (f^2 + g^2)}{4 + (f^2 + g^2)} \left(-\frac{4f}{4 - (f^2 + g^2)}, -\frac{4g}{4 - (f^2 + g^2)}, 1 \right) \cdot \vec{s}. \quad (5.6)$$

In summary, the shape from shading problems can be formulated by using either $\vec{N}$ or (p, q) or (f, g) . Together with adequate boundary conditions, the shape from shading problem is to solve a linear or nonlinear partial differential equation (PDE) of first order. In this chapter, we have limited our attention to

recovering a non-self-shadowing Lambertian surface with constant albedo. We further assume that the object is illuminated by a *single distant* light source.

The earliest mathematical method to solve this problem, posed by Horn [28], is based on the characteristic strip expansion (see next section). Like the idea of dealing with any other nonlinear problems, linearization is the most common and easiest approach to obtain an approximation to the exact solution. Taylor expansion can be used to derive a linear equation associated with the original equation. After the equation is linearized, some criteria are chosen to discretize the linear PDE to get an algebraic equation. Such methods include, for example, numerical differentiation and integral transform (see [13, 15]). Then a numerical method is selected to find an approximation of the solution to the algebraic problem numerically. Since there is no guarantee to the existence of the solution, another approach is to search for optimization solution. This procedure includes introduction of a satisfactory energy function and finding the solution of the posed optimization problem numerically.

5.1.2 About this Chapter

This chapter is written for the purpose of introducing students and practitioners to the necessary elements, including numerical methods and algorithms, in order to understand the current methods and use them in dealing with some practical problems. With a limited set of mathematical jargons and symbols, the emphasis is given to kindle interest for the problem. This has been done by selecting those methods which are easily understood and best demonstrate the idea of SFS models. Of course, our selection of the techniques and numerical examples is limited by the usual constraints: author prejudice and author limitation. Our goal is to draw an outline or describe the framework for solving this problem and leave the details to the readers for further study.

We conclude this section by giving an outline of the chapter. In this chapter, we consider one of the reconstruction methods: shape from shading. The chapter is organized as follows: the first section serves as a brief review of the SFS models, their history, and recent developments. Section 5.2 provides certain mathematical background related to SFS. It discusses some selected numerical methods for solving discretized SFS problems. The emphasis is given to the well-developed method—Finite difference method (FDM). Section 5.4 is devoted to the illustration of numerical techniques for solving SFS problems. It concerns

related algorithms and their implementations. The section ends with a discussion about the advantages and disadvantages of the algorithms introduced in this section. The last section attempts to introduce the recently developed wavelet-based methods by using an example. A part of the section, however, is devoted to a brief introduction of the basic facts of wavelet theory. In the hope that readers will be able to extrapolate the elements presented here to initiate the understanding of the subject on their own, the chapter concludes with some remarks on other advanced methods. Finally, we include an intensive set of references to make up whatever important spirits which the authors have indeed hardly to touch in this short chapter.

5.2 Mathematical Background of SFS Models

Many problems of mathematical physics lead to PDEs. In general, PDEs are classified in many different ways. However, in most mathematics literature, PDEs are classified on the basis of their characteristics, or curves of information propagation (see, for example, [60] and [19]). The irradiance equation (5.2) is a first-order nonlinear equation. The general format of such an equation in the two-dimensional space is given by

$$f\left(\frac{\partial Z}{\partial x}, \frac{\partial Z}{\partial y}, z, x, y\right) = 0, \quad (x, y) \in \Omega. \quad (5.7)$$

Theoretically, a compactible boundary condition should be given as

$$Z(x, y) = g(x, y), \quad (x, y) \in \Gamma,$$

where Γ is the boundary curve of the domain Ω .

In general, nonlinear PDEs are much more difficult than the linear equations, while the more the nonlinearity affects the higher derivatives, the more difficult the PDE is. The irradiance equation (5.2) with a nonlinear reflectance map (5.5) is a hyperbolic PDE of first order with severe nonlinearity. Although the nonlinearity prevents the possibility of deriving any simple method to solve the equation, there are still some techniques developed to obtain local information of the solution to a certain extent. In this section, we briefly review some basics about the irradiance equation, namely, the existence and uniqueness of

the solution. We also describe a technique, characteristic strip method, which leads to the solution of the equation.

5.2.1 The Uniqueness and Existence

It has been shown that surfaces with continuously varying surface orientation give rise to shaded images. The problem of shape from shading is to reconstruct the three-dimensional shape of a surface from the brightness or intensity variation in a *single* black-and-white photographic image of the surface. For a long time in history, the SFS model was believed ill-posed. However, it has been shown that the problem in its *idealized* form is actually well posed or “partially” well posed under a wide range of conditions ([32, 42]).

The standard assumptions for the idealized surface are:

- “Lambertian” reflectance—the surface is matte, rather than mirror-like and reflects light evenly in all directions,
- “Orthographic” projection—the illuminating light is from a single known direction and that the surface is distant from the camera, and
- “Nonocclusion”—all portions of the surface are visible.

If only one source of illumination is available, uniqueness can be proved. Further Saxberg [51, 52] discussed conditions for existence of the solution. Oliensis [41, 42] has shown the following:

Proposition 1. *For an image of a light region contained in a black background, if the reflectance map is known, as given in (5.2), then there is a unique solution for a generic surface which is smooth and non-self-occluding.*

Despite various existence and uniqueness theorems for smooth solutions (see [14, 30, 34, 41, 42, 51, 52, 64]), in practice the problem is unstable, which is catastrophic for general numerical algorithms [4, 18]. This is because the reflectance map is, in general, given by its sampled data rather than an analytic expression. This data may be sparse and contaminated by noise. We will not go into the detailed discussion about the uniqueness and existence issue here; the readers who are interested in this issue are referred to the excellent review paper by Hurt [32] and references [14, 30, 34].

5.2.2 The Characteristic Strip Method

Horn [29] established a method to find the solution of (5.2), the characteristic strip method ([29], p. 244). This method is to generate the characteristic strip expansion for the nonlinear PDE (5.2) along a curve on the surface by solving a group of five ordinary differential equations called characteristic equations:

$$\begin{aligned}\dot{x} &= R_p, & \dot{y} &= R_q, & \dot{Z} &= pR_p + qR_q, \\ \dot{p} &= E_x, & \dot{q} &= E_y,\end{aligned}$$

where the dot denotes differentiation along a solution curve. The characteristic equation can be organized in a matrix format:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ Z \\ p \\ q \end{bmatrix} = \begin{bmatrix} R_p \\ R_q \\ pR_p + qR_q \\ E_x \\ E_y \end{bmatrix}. \quad (5.8)$$

The solution, $(x, y, Z, p, q)^T$, to (5.8) forms a characteristic strip along the curve. The curves traced out by the solutions of the five ordinary differential equations are called characteristic curves, and their projections in the image are called base characteristics. If an initial curve (with known derivative along this curve) is given by a parametric equation:

$$\vec{r}(\eta) = \{x(\eta), y(\eta), Z(\eta)\}^T,$$

then we can derive the surface by integrating the equation

$$\frac{\partial Z}{\partial \eta} = p \frac{\partial x}{\partial \eta} + q \frac{\partial y}{\partial \eta}. \quad (5.9)$$

Example 2. Consider an ideal Lambertian surface illuminated by a light source close to the viewer at $(p_0, q_0, 1) = (0, 0, 1)$. (p_0, q_0) is the direction toward the light source. In this case, the image irradiance equation is

$$I(x, y) = \frac{1}{\sqrt{1 + p^2 + q^2}},$$

where we have set $\rho = 1$ for simplicity.

The characteristic equation is then given by

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ Z \\ p \\ q \end{bmatrix} = \begin{bmatrix} -p(1+p^2+q^2)^{-\frac{3}{2}} \\ -q(1+p^2+q^2)^{-\frac{3}{2}} \\ -(p^2+q^2)(1+p^2+q^2)^{-\frac{3}{2}} \\ I_x \\ I_y \end{bmatrix} = \begin{bmatrix} -pI^3(x, y) \\ -qI^3(x, y) \\ -(p^2+q^2)I^3(x, y) \\ I_x \\ I_y \end{bmatrix}. \quad (5.10)$$

In general, (5.10) has to be solved numerically to get characteristic curves. In practice, since the intensity function is only available as a discrete set of data, analytic solution is simply impossible. An alternative method, which is also the most common method in solving any nonlinear problem, is the calculus of variations.

5.2.3 The Idea of Calculus of Variations

We denote the nonlinear partial differential operator associated with (5.7) by

$$\mathbf{D}[Z](x, y) = 0. \quad (5.11)$$

If there exists an “energy” function $\mathbf{E}[z](x, y)$ such that

$$\mathbf{D}[\cdot] = \mathbf{E}'[\cdot],$$

then the problem (5.11) is equivalent to solving:

$$\mathbf{E}'[Z] = 0. \quad (5.12)$$

The solution of the irradiance equation is the critical point of $\mathbf{E}$. In many cases, finding the minimum (or maximum) is easier than solving (5.11) directly. In addition, many of the laws of physics and other scientific disciplines arise directly as variational principle [11, 19, 60].

5.2.3.1 Euler Equation and Lagrange Multipliers

Calculus of variations seeks to find the path, curve, surface, etc. for which a given function has a stationary value (which, in physical problems, is usually a minimum or maximum). In 2D space, this involves finding stationary values of

integrals of the form

$$E = \int_{\Omega} F(x, y, Z, p, q) dx dy.$$

E has an extremum only if the Euler differential equation

$$F_z - \frac{\partial}{\partial x} F_p - \frac{\partial}{\partial y} F_q = 0$$

is satisfied. If the solution is subject to the constraints

$$g_j(x, y, Z) = 0, \quad j = 1, \dots, k,$$

then we have

$$G = F + \sum_{j=1}^k \lambda_j g_j(x, y, Z).$$

Now the Euler equation is

$$F_z - \frac{\partial}{\partial x} F_p - \frac{\partial}{\partial y} F_q + \sum_{j=1}^k \lambda_j \frac{\partial g_j}{\partial Z} = 0. \quad (5.13)$$

The λ_j 's are called Lagrange multipliers. An example is provided in Section 5.3.2.1.

5.2.3.2 The Constraint Functions Used in SFS Models

When iterative algorithms are used for solving the SFS problem, constraints will be proposed to secure a weak solution. The following constraints are examples:

- (1) total squared brightness error [27]:

$$G_0 = \int_{\Omega} (I(x, y) - R(p, q))^2 dx dy. \quad (5.14)$$

- (2) weak smoothness: After the tangent planes are available, the surface Z is reconstructed by minimizing the following functional:

$$G_1 = \int_{\Omega} (p_x^2 + p_y^2 + q_x^2 + q_y^2) dx dy. \quad (5.15)$$

- (3) integrability: Since p and q are considered independent variables, (p, q) may not correspond to the orientation of the underlying surface Z , that is, the surface Z cannot be derived by integrating $Z_x = p$, $Z_y = q$. An

integrability constraint is then posed as

$$G_2 = \int_{\Omega} (p_y - q_x)^2 dx dy, \quad (5.16)$$

or

$$G_3 = \int_{\Omega} (Z_x - p)^2 + (Z_y - q)^2 dx dy. \quad (5.17)$$

(4) depth [58]:

$$G_4 = \int_{\Omega} (Z(x, y) - d(x, y))^2 dx dy. \quad (5.18)$$

(5) minimal curvature:

$$G_5 = \int_{\Omega} (Z_{xx}^2 + 2Z_{xy}^2 + Z_{yy}^2) dx dy. \quad (5.19)$$

(6) strong smoothness [31]: Introduced in [31], this constraint is used to enforce a stronger integrability and smoothness:

$$G_6 = \int_{\Omega} ((Z_{xx} - p)^2 + (Z_{yy} - q)^2) dx dy. \quad (5.20)$$

A combination of the first three of the above constrains (5.14), (5.15), and (5.16), that is,

$$Eng(p, q) = \sum_{k=1}^3 \lambda_k G_k, \quad (5.21)$$

is commonly used to control the stability of iteration algorithms. Here λ_k , $k = 1, 2, 3$, are the Lagrange multipliers. The last three of the above constraints are introduced to enforce the smoothness and convergence (of the depth constraint) of the approximation solution. We will demonstrate some examples in Section 5.3.

An iterative scheme for solving the shape from shading problem has been proposed by Horn *et al.* [27]. The method consists the following two steps.

Step 1. A preliminary phase recovers information about orientation of the planes tangent to the surface at each point by minimizing a functional containing the image irradiance equation and an integrability constraint:

$$Eng(p, q) = \int_{\Omega} [(E(x, y) - R(p, q))^2 + \lambda(p_y - q_x)^2] dx dy, \quad (5.22)$$

Step 2. After the tangent planes are available, the surface Z is reconstructed by minimizing the functional (5.17).

Remark 1. *The variational approach introduced in [27] does not necessarily guarantee the existence of a solution of the problem. In fact, [10] has addressed this crucial question and shown that the variational approach does not lead to an exact solution of the SFS problem in general. For the discretization of the Euler differential equation and some numerical methods used to solve it, see Sections 5.2.5 and 5.3.*

5.2.4 Numerical Methods for Linear and Nonlinear SFS Models

Unfortunately, in practice, even with greatly simplified initial and boundary conditions, the analytic solution for a nonlinear PDE is too difficult to obtain in a closed form. A numerical technique is then employed to find a reasonable approximate solution. In this sense, it is more useful to know of such numerical methods which provide us a technique to be actually used in everyday life.

When dealing with the shape from shading model, it becomes clear that the analytic solutions to the irradiance equation (5.2) or the system of ordinary equations (5.8) are practically impossible.

To obtain a numerical approximation for the solution, the first step is to simplify the irradiance equation. The basic approaches for this purpose include:

- direct method: discretizing the irradiance equation directly using Taylor series or difference formula.
- integral transform: using linear transforms, such as Fourier transform and wavelet transform [13, 15].
- projection method: approximating the solution by a finite set of basis functions.

The second step is to choose a criterion to discretize the simplified irradiance equation to get an algebraic equation. Then a numerical method is chosen to solve the algebraic equation. Some of these steps can be done simultaneously.

5.2.4.1 Finite Difference Method

The FDM consists of two steps: (1) replacing the (partial) derivatives by some numerical differentiation formulas to get a difference equation, that is,

derivatives are discretized by using “difference” and (2) solving the derived difference equation—an algebraic equation—by using either an iterative or a direct method.

To begin with, we first partition the domain Ω by a mesh grid. For example, we use a uniform mesh grid with grid lines:

$$\begin{aligned}x_j &= x_0 + jh, & j &= 0, 1, \dots, J, \\y_l &= y_0 + lk, & l &= 0, 1, \dots, L,\end{aligned}$$

where $h = x_{i+1} - x_i$ and $k = y_{i+1} - y_i$ are the mesh sizes in the x and y directions, respectively. For simplicity, we write $f_{j,l} = f(x_j, y_l)$, the function values on the nodes of the mesh.

Using Taylor expansion and intermediate value theorem, we can derive the following numerical differentiation formulas:

- forward difference formula:

$$\begin{aligned}u_x(x, y) &\approx \frac{1}{h}(u_{i+1,j} - u_{i,j}), \\u_y(x, y) &\approx \frac{1}{k}(u_{i,j+1} - u_{i,j});\end{aligned}\tag{5.23}$$

- backward difference formula:

$$\begin{aligned}u_x(x, y) &\approx \frac{1}{h}(u_{i,j} - u_{i-1,j}), \\u_y(x, y) &\approx \frac{1}{k}(u_{i,j} - u_{i,j+1});\end{aligned}\tag{5.24}$$

- centered difference formula:

$$\begin{aligned}u_x(x, y) &\approx \frac{1}{2h}(u_{i+1,j} - u_{i-1,j}), \\u_y(x, y) &\approx \frac{1}{2k}(u_{i,j+1} - u_{i,j-1}).\end{aligned}\tag{5.25}$$

Similarly, the three second-order partial derivatives are given by

$$\begin{aligned}u_{xx}(x, y) &\approx \frac{1}{hk}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}), \\u_{yy}(x, y) &\approx \frac{1}{hk}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}), \\u_{xy}(x, y) &\approx \frac{1}{hk}(u_{i+1,j+1} - 2u_{i,j} + u_{i-1,j-1}),\end{aligned}\tag{5.26}$$

Formulas (5.23–5.26) will be used in Section 5.3 to discretize (or linearize) the irradiance equation (5.2).

We now demonstrate the idea of FDM by the following examples.

Example 3. As an example, we consider using FDM to solve the linear shape from shading problem (5.3) on a square domain:

$$\Omega = \{(x, y), 0 < x < 1, 0 < y < 1\}$$

with the boundary condition given by (5.4). Using forward difference formula (5.23), we have

$$p \approx \frac{1}{h}(Z_{i+1,j} - Z_{i,j}) \text{ and } q \approx \frac{1}{k}(Z_{i,j+1} - Z_{i,j}). \quad (5.27)$$

We rewrite Eq. (5.3) as

$$\Lambda I(x, y) = p_0 p + q_0 q, \quad (5.28)$$

where $\Lambda = \sqrt{1 + p_0^2 + q_0^2} - \rho$. Substituting (5.27) and (5.28), we have

$$\Lambda I_{i,j} = \frac{p_0}{h}(Z_{i+1,j} - Z_{i,j}) + \frac{q_0}{k}(Z_{i,j+1} - Z_{i,j}).$$

Solving for $Z_{i,j+1}$, we have

$$Z_{i,j+1} = -\alpha Z_{i+1,j} + (\alpha + 1)Z_{i,j} + \beta I_{i,j},$$

where $\alpha = \frac{p_0 k}{q_0 h}$, $\beta = \frac{k \Lambda}{q_0}$, $i = 0, \dots, n-2$, $Z_{i,0} = g_1(x_i)$ and $Z_{n-1,j} = g_2(y_j)$, $j = 0, \dots, n-2$. Written in matrix format, we have

$$\begin{bmatrix} Z_{0,j+1} \\ Z_{1,j+1} \\ \dots \\ Z_{n-2,j+1} \end{bmatrix} = \begin{bmatrix} \alpha + 1 & -\alpha & 0 & \dots & 0 \\ & \alpha + 1 & \alpha & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \alpha + 1 \end{bmatrix} \begin{bmatrix} Z_{0,j} \\ Z_{1,j} \\ \dots \\ Z_{n-2,j} \end{bmatrix} + \beta \begin{bmatrix} I_{0,j} \\ I_{1,j} \\ \dots \\ I_{n-2,j} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dots \\ -\alpha Z_{n-1,j} \end{bmatrix}, \quad j = 0, 1, \dots, n-2. \quad (5.29)$$

Figure 5.1 shows the discretization we are using.

The finite difference scheme (5.29) is called the explicit method since it is given by an iterative formula. If instead, the central (5.25) and forward difference formulas (5.23) are used to approximate the partial derivatives, an implicit finite difference scheme can be derived. The approximate solution is then derived iteratively by using the iteration formula (5.29). Numerical

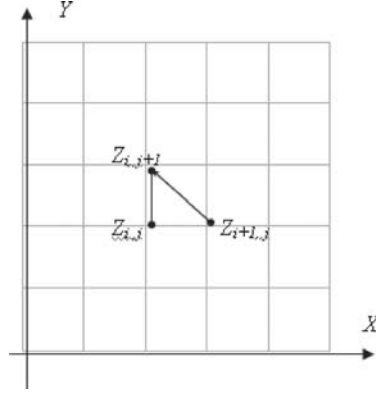


Figure 5.1: The grid mesh for the discretization in Example 3.

methods used to solve these matrix equations, for example, the Jacobi method, can be found in the standard numerical analysis textbooks [24].

For a nonlinear shape from shading model (5.5), we have to linearize the reflectance map by using Taylor expansion to get a linear equation and then apply the FDM in a similar way as in the above example. To linearize the equation, we only need to replace the nonlinear part in Eq. (5.2) by its linear approximation. We first rewrite the equation to separate the linear and nonlinear parts:

$$I(x, y)\sqrt{1 + p_0^2 + q_0^2}\sqrt{1 + p^2 + q^2} = \rho(1 + p_0 p + q_0 q). \quad (5.30)$$

Denoting the nonlinear part by

$$F(p, q) := \sqrt{1 + p^2 + q^2},$$

the Taylor expansion of $F(p, q)$ at $(\bar{p}, \bar{q})$ is

$$\begin{aligned} F(p, q) &= F(\bar{p}, \bar{q}) + (p - \bar{p})F_p(\bar{p}, \bar{q}) + (q - \bar{q})F_q(\bar{p}, \bar{q}) \\ &\quad + O(|(p - \bar{p})^2 + (q - \bar{q})^2|) \\ &\approx \sqrt{1 + \bar{p}^2 + \bar{q}^2} + (p - \bar{p})\frac{\bar{p}}{\sqrt{1 + \bar{p}_0^2 + \bar{q}_0^2}} + (q - \bar{q})\frac{\bar{q}}{\sqrt{1 + \bar{p}_0^2 + \bar{q}_0^2}}, \end{aligned} \quad (5.31)$$

where the error term $O(|(p - \bar{p})^2 + (q - \bar{q})^2|)$ depends on the value of $(\bar{p}, \bar{q})$ and the smoothness of the solution function Z . If we assume that $Z \in C^2(\Omega)$, then this error term can be ignored locally. Now we substitute (5.30) into (5.31) to have the linearized irradiance equation:

$$P(x, y)p + Q(x, y)q = \tilde{I}(x, y), \quad (5.32)$$

where

$$\begin{aligned} P(x, y) &= -\bar{p}I(x, y) + \rho p_0, \\ Q(x, y) &= -\bar{q}I(x, y) + \rho q_0, \\ \tilde{I}(x, y) &= \sqrt{1 + p_0^2 + q_0^2} \sqrt{1 + \bar{p}^2 + \bar{q}^2 - (\bar{p}^2 + \bar{q}^2)} - \rho. \end{aligned}$$

Note that (5.32) is a first-order partial differential equation with nonconstant coefficients P and Q and, therefore, the FDM can be used to solve it in the same way as in Example 2.

5.2.4.2 Remarks

Remark 2. About convergence of finite difference method. Every numerical method provides a formalism of generating discrete algorithms for approximating the solution of a PDE. Such a task could be done automatically by a computer if there were no mathematical skills that require human involvement. Consequently, it is necessary to understand the mathematics in this black box which you put in your PDE for processing. This will involve discussion on convergence, stability, and error analysis. However, these topics are beyond the scope of this introductory chapter. We hope the loose ends left here will stimulate your curiosity and further motivate your deep interest in this subject. For the finite difference method used to solve the SFS model, some results related to the issue of convergence can be found in Ulich [64]. For the linear problem, Ulich proved convergence for three methods: forward difference method, backward difference method, and central and forward difference method for certain light directions. For the central and forward finite difference method applied to the linear PDE derived from linearization of a nonlinear shape from shading problem, she was able to prove convergence for all light directions. For the case of application of FDM to the linear shape from shading models, Wei et al. [65] discussed the convergence properties for four explicit, two implicit, and four semi-implicit finite difference algorithms. They also give comparisons of accuracy, solvability, stability, and convergence of these schemes.

Remark 3. About Multiscale methods. Simple iterative methods (such as the Jacobi method) tend to damp out high-frequency components of the error faster than any other method. Multiscale methods are introduced to

improve the convergence and accuracy of the numerical solutions derived in the “single scale” methods. The multigrid method appears to be one of the most successful multiscale methods for solving the SFS problems. The multigrid method associated with vision problems was studied in the early 1960s by Fedorenko and Bakhvalov [3] and later in the work of many others, for example, [1, 56, 57]. For readers without previous experiences in the multigrid method, excellent tutorials can be found in the books [8] and [44, 62]. A classic book on multigrid method is [23]. Some other novel multiscale methods are developed recently, for example, the panel-clustering method, wavelet-based matrix compression, and the hierarchical matrix decomposition technique.

5.3 Numerical Algorithms and Their Implementations

It is understood that the problem of shape from shading is fundamentally a very difficult mathematical problem. Nevertheless, some methods were developed while attempting to solve the problem numerically for some particular cases. Most of the methods work for a particular type of images.

SFS techniques can be roughly divided into four groups:

- (1) *Minimization approach.* The method is to reconstruct the shape of the object by minimizing the well-designed energy function over the entire image. The idea is similar to that in deformable models used in image segmentation which segment the object by minimizing certain predefined energy functions. The constraint functions can be the brightness constraint, the gradient constraint, the smooth constraint, etc. (See Section 5.2.3.2).
- (2) *Propagation approach.* The method starts from some initial curve, which is defined as special points such as the brightest or the darkest points in the image, and propagates the shape information across the entire image. (See 5.2.2).
- (3) *Local approach.* The method reconstructs the shape by assuming local surface type. The surface is derived by matching derivatives of measured intensity to those assumed spherical surface.

- (4) *Linear approach.* The method linearizes the reflectance map in tilts or depth. The linear model can be solved and the shape of the object can be calculated. The basic assumption behind this idea is that the lower order components of the reflectance maps dominate the reflectance maps.

In the rest of this section, we will discuss in detail two widely used methods: the linear approach and the minimization approach.

5.3.1 Linear Approaches

In this approach, the basic idea is to linearize the reflectance map and solve the depth information of the shape from the equations. Different linear functions can be formulated in terms of surface gradient or the height of the surface. In the following contexts we will introduce two approaches which are based on linear equations in terms of gradients and the heights of the surface. Both start with the use of first-order finite difference to discretize the reflectance equation. However, they are different after the initial discretization. Pentland's algorithm uses the Fourier transform and inverse Fourier transform to obtain the depth map, while Tsai-Shah's algorithm uses the Newton method to derive the depth map. We will explain these two linear approaches in the following sections.

5.3.1.1 Pentland's Linear Approach

Pentland [6, 46, 47] introduced a method which takes directly linearization of the reflectance map in the surface gradient (p, q) . It greatly simplifies the shape from shading problem with sacrificing part of the accuracy of the reconstruction result. We start with the expansion of the right-hand side of the irradiance equation (5.2) at $p = p_0, q = q_0$ using Taylor's expansion. We have

$$R(p, q) = R(p_0, q_0) + (p - p_0) \frac{\partial R}{\partial p}(p_0, q_0) + (q - q_0) \frac{\partial R}{\partial q}(p_0, q_0). \quad (5.33)$$

For Lambertian reflectance, Eq. (5.33) at $p_0 = 0, q_0 = 0$ can be reduced to

$$I(x, y) = R(0, 0) + p \frac{\partial R}{\partial p}(0, 0) + q \frac{\partial R}{\partial q}(0, 0). \quad (5.34)$$

Denoting by τ the tilt of the illuminant and by γ the slant of the illuminant, the above equation can be rewritten as

$$I(x, y) = \cos \gamma + p \cos \tau \sin \gamma + q \sin \tau \cos \gamma. \quad (5.35)$$

Using forward difference formula (5.23), we have

$$\begin{aligned} p &= \frac{\partial}{\partial x} Z(x, y) = Z(x+1, y) - Z(x, y), \\ q &= \frac{\partial}{\partial y} Z(x, y) = Z(x, y+1) - Z(x, y). \end{aligned} \quad (5.36)$$

By taking Fourier transform on the two sides of Eq. (5.36), we can get the following results:

$$\begin{aligned} p &= \frac{\partial}{\partial x} Z(x, y) \langle - \rangle^{\mathcal{F}} F_Z(w_1, w_2) (-iw_1), \\ q &= \frac{\partial}{\partial y} Z(x, y) \langle - \rangle^{\mathcal{F}} F_Z(w_1, w_2) (-iw_2). \end{aligned} \quad (5.37)$$

Substituting Eq. (5.37) into Eq. (5.35) and taking Fourier transform on both sides, we obtain

$$F_I(w_1, w_2) = F_Z(w_1, w_2) (-iw_1) \cos \tau \sin \sigma + F_Z(w_1, w_2) (-iw_2) \sin \tau \sin \sigma, \quad (5.38)$$

where $F_I(w_1, w_2)$ and $F_Z(w_1, w_2)$ are the Fourier transform of the input image $I(x, y)$ and depth map $Z(x, y)$, respectively.

After rearranging Eq. (5.38), we obtain

$$F_Z(w_1, w_2) = \frac{F_I(w_1, w_2)}{2\pi(\sqrt{w_1^2 + w_2^2}) \sin \gamma (w_1 \cos \tau + w_2 \sin \tau)}. \quad (5.39)$$

By taking the inverse Fourier transform, we can obtain the depth map

$$Z(x, y) = F^{-1}(F_Z(w_1, w_2)). \quad (5.40)$$

It is obvious that this approach does not need iterative computation and can provide an approximate solution quickly. However, like all the other linear approaches, this method makes an assumption that the reflectance map is locally linear. Comparing Eq. (5.35) with the normal reflectance equation:

$$R(p, q) = \frac{\cos \gamma + p \cos \tau \sin \gamma + q \sin \tau \cos \gamma}{\sqrt{1 + p^2 + q^2}}. \quad (5.41)$$

We observe that the linear method ignores the quadratic terms in Eq. (5.41). If we have a 3D object which has rapid changes in depth, both p and q will dominate $R(p, q)$, Pentland's algorithm may not provide promising results. Fortunately, some objects do change smoothly so that linear approximation is satisfactory to certain extent.

The algorithm can be described by the following procedure:

- *Step 1.* Input the original parameters of the reflectance map,
- *Step 2.* Calculate the Fourier transform of the depth map $Z(w_1, w_2)$ using Eq. (5.39),
- *Step 3.* Calculate the inverse Fourier transform of the depth map $Z(x, y)$ using Eq. (5.40).

The way to realize Pentland's algorithm can be described by the following pseudocode.

Algorithm 1. Pentland's algorithm

Input $Z_{\min}$ (mindepthvalue), $Z_{\max}$ (maxdepthvalue), (x, y, z) (direction of the light source), $I(x, y)$ (input image)

$D \leftarrow \sqrt{x^2 + y^2 + z^2}$, $sx \leftarrow x/D$, $sy \leftarrow y/D$, $sz \leftarrow z/D$.

$\sin \gamma \leftarrow \sin(\arccos(lz))$,

$\sin \tau \leftarrow \sin(\arctan(sy/sx))$,

$\cos \tau \leftarrow \cos(\arctan(sy/sx))$.

for $i = 1$ to $width(I)$ do

 for $j = 1$ to $height(I)$ do

$F_I(w_1, w_2) \leftarrow FFT(I(i, j))$

$B \leftarrow 2\pi (\sqrt{w_1^2 + w_2^2}) \sin \gamma (w_1 \cos \tau + w_2 \sin \tau)$

$Z(x, y) \leftarrow IFFT(FI(w_1, w_2)/B)$

 end do

end do

Normalize($Z(x, y)$, $Z_{\max}$, $Z_{\min}$)

Output $Z(x, y)$

The subfunctions *FFT*, *IFFT*, and *Normalize* are all standard math functions used in signal and image processing.

We now demonstrate this method by using the following example.

Example 4. *Reconstruct the surface of a synthetic vase using Pentland's method. The experiments are based on the synthetic images that are generated using true depth maps. Figure 5.2(a) shows the synthetic vase and the reconstruction results using Pentland's algorithm. The light is from above at $(x = 1,$*

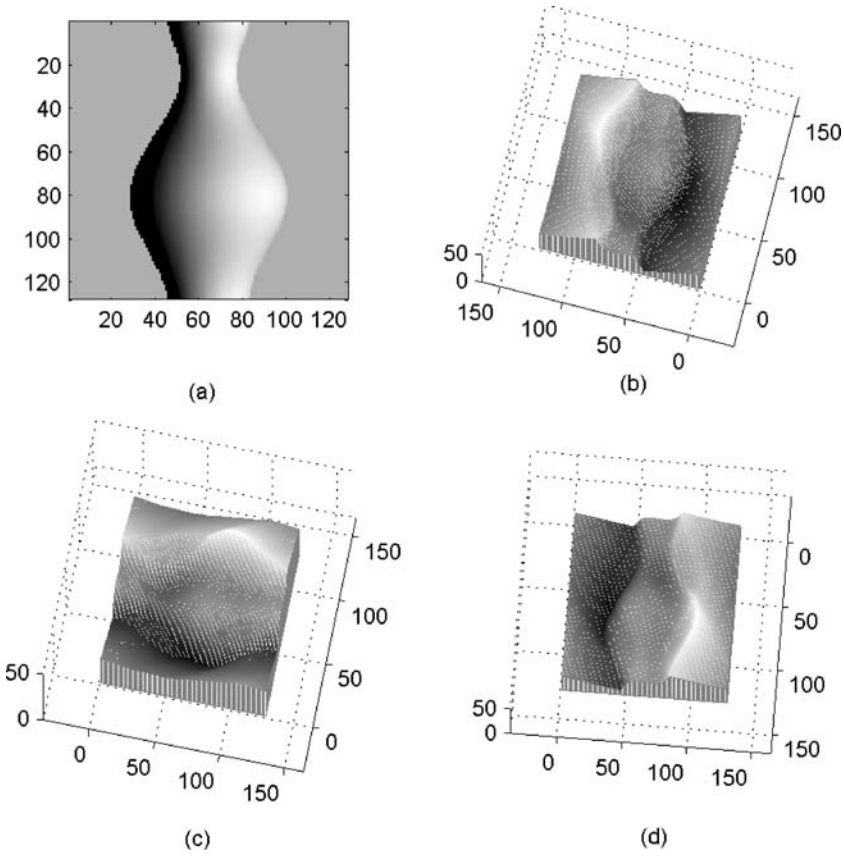


Figure 5.2: Pentland's linear SFS algorithm applied to the synthetic vase image. (a) is the input image with light source ($x = 1$, $y = 0$, $z = 1$). (b), (c), and (d) are the reconstructed surface from three different directions.

$y = 0$, $z = 1$). The input image is showed in Fig. 5.2(a). The surface, showed in Figs. 5.2(b), (c), and (d), is the reconstructed surface from three different directions. Pentland's algorithm produces reasonable results as expected for a vase. In general, the experiment shows that Pentland's algorithm roughly recovered the object on the surface where the reflectance changes linearly with respect to the surface shape.

5.3.1.2 Tsai-Shah's Linear Approach

Tsai-Shah [63, 68] proposed another linearization method to solve the SFS problem. Instead of applying the Fourier transform and inverse Fourier transform,

this method discretizes the reflectance map in a different way. Like Pentland's method, the surface orientation (p, q) is approximated by its linear approximation using the forward difference formula (5.36), while unlike Pentland's method, the reflectance map is then directly linearized in terms of the depth Z using Taylor series expansion. Finally, Newton's iteration method is applied to the discretized equation to get a numerical approximation to the depth Z . In what follows, we will derive this scheme step by step.

To begin with, we rewrite the irradiance equation (5.2) in the following format:

$$0 = f = I - R. \quad (5.42)$$

Replacing p and q by their linear approximation using the forward difference formulas (5.36), we obtain

$$\begin{aligned} 0 &= f(I(x, y), Z(x, y), Z(x-1, y), Z(x, y-1)) \\ &= I(x, y) - R(Z(x, y) - Z(x-1, y), Z(x, y) - Z(x, y-1)). \end{aligned} \quad (5.43)$$

If we take the Taylor series expansion about a given depth map Z^{n-1} , we get the following equation:

$$\begin{aligned} 0 &= f(I(x, y), Z(x, y), Z(x-1, y), Z(x, y-1)) \\ &\approx f(I(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), Z^{n-1}(x, y-1)) \\ &\quad + \left[(Z(x, y) - Z^{n-1}(x, y)) \right. \\ &\quad \times \left. \frac{\partial f(I(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), Z^{n-1}(x, y-1))}{\partial Z(x, y)} \right] \\ &\quad + \left[(Z(x-1, y) - Z^{n-1}(x-1, y)) \right. \\ &\quad \times \left. \frac{\partial f(I(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), Z^{n-1}(x, y-1))}{\partial Z(x-1, y)} \right] \\ &\quad + \left[(Z(x, y-1) - Z^{n-1}(x, y-1)) \right. \\ &\quad \times \left. \frac{\partial f(I(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), Z^{n-1}(x, y-1))}{\partial Z(x, y-1)} \right]. \end{aligned} \quad (5.44)$$

Given an initial value $Z^0(x, y)$, and using the iterative formula:

$$\begin{aligned} Z^n(x, y-1) &= Z^{n-1}(x, y-1), \\ Z^n(x-1, y) &= Z^{n-1}(x-1, y), \end{aligned}$$

each value of the depth map can be iteratively calculated. In fact, (5.44) can be read as

$$0 = f(Z(x, y)) \approx f(Z^{n-1}(x, y)) + \left(Z(x, y) - Z^{n-1}(x, y) \frac{df(Z^{n-1}(x, y))}{dZ(x, y)} \right). \quad (5.45)$$

Rearranging Eq. (5.45), we obtain

$$Z^0(x, y) = \text{initial value} \quad (5.46)$$

$$Z^n(x, y) = Z^{n-1}(x, y) + \frac{-f(Z^{n-1}(x, y))}{\frac{d}{dZ(x, y)}f(Z^{n-1}(x, y))}, \quad n = 1, 2, \dots,$$

where

$$\begin{aligned} \frac{df(Z^{n-1}(x, y))}{dZ(x, y)} = & -1 \left(\frac{\cos \tau \tan \gamma + \sin \tau \tan \gamma}{\sqrt{p^2 + q^2 + 1} \sqrt{\tan^2 \gamma + 1}} \right. \\ & \left. - \frac{(p + q)(p \cos \tau \tan \gamma + q \sin \tau \tan \gamma + 1)}{\sqrt{(p^2 + q^2 + 1)^3} \sqrt{\tan^2 \gamma + 1}} \right). \end{aligned} \quad (5.47)$$

By iteratively using formula (5.46), we obtain the approximation of the depth map $Z(x, y)$. Readers may have noticed that the iterative formula is Newton's formula.

This method has a similar disadvantage as the algorithm based on linear approach. However, it is faster since it does not need to compute the FFT and IFFT.

The algorithm can be described by the following procedure:

- *Step 1.* Input the original parameters of the reflectance map,
- *Step 2.* Set the initial guess of $Z^0(x, y) = 0$,
- *Step 3.* Refine the depth map $Z^k(x, y)$ using Eq. (5.46).

The way to realize Pentland's algorithm can be described by the following pseudocode.

Algorithm 2: Tsai–Shah's linearization method

Input $Z_{\min}$ (mindepthvalue), $Z_{\max}$ (maxdepthvalue), (x, y, z) (direction of the light source), $I(x, y)$ (inputimage)

$z^0 \leftarrow 0$;

$p^0 \leftarrow q^0 \leftarrow 0$;

$p \leftarrow q \leftarrow p^0 \leftarrow q^0$;

$D \leftarrow \sqrt{x^2 + y^2 + z^2}$, $sx \leftarrow x/D$, $sy \leftarrow y/D$, $sz \leftarrow z/D$.

$\sin \gamma \leftarrow \sin(\arccos(lz))$, $\sin \tau \leftarrow \sin(\arctan(sy/sx))$,

$\cos \tau \leftarrow \cos(\arctan(sy/sx))$.

```

for  $i = 1$  to  $\text{width}(I)$  do
  for  $j = 1$  to  $\text{height}(I)$  do
     $dfz \leftarrow -1 \cdot \{(\cos \tau \tan \gamma + \sin \tau \tan \gamma) /$ 
       $\sqrt{(p^2 + q^2 + 1)(\tan^2 \gamma + 1)}$ 
       $- (p + q)(p \cos \tau \tan \gamma + q \sin \tau \tan \gamma + 1) /$ 
       $\sqrt{(p^2 + q^2 + 1)^3(\tan^2 \gamma + 1)}\}$ 
     $Z(i, j) \leftarrow Z^0(i, j) + -f(Z^0(i, j)) / dfz$ 
     $p \leftarrow Z(i, j) - Z(i, j - 1)$ 
     $q \leftarrow Z(i, j) - Z(i - 1, j)$ 
  end do
end do
 $\text{Normalize}(Z(x, y), Z_{\max}, Z_{\min})$ 
Output  $Z(x, y)$ 

```

The subfunction *Normalize* is a standard math function used in signal and image processing.

We now demonstrate this method by using the following example.

Example 5. *Reconstruct the surface of a synthetic vase using Tsai–Shah’s method.*

In order to compare with Pentland’s method, here we consider reconstruction of the same surface as in Example 2—the surface of a synthetic vase. Figure 5.3 shows the synthetic vase and the reconstruction results using Tsai–Shah’s algorithm from three different directions. The light is from above at ($x = 0, y = 0, z = 1$). The input image is showed in Fig. 5.3(a). The surface, shown in Fig. 5.3(b), (c), and (d), is the reconstructed surface from three different directions. Tsai–Shah’s algorithm works well and produces good results as expected for the vase. However, it is sensitive to noises as we will point out in the next section. In general, the experiment shows that Tsai–Shah’s algorithm can reconstruct the object well on the surface where the reflectance changes linearly with respect to the surface shape.

5.3.2 Optimization Approaches

As we pointed out earlier, the problem of recovering the shape from shading can be based on solving the irradiance equation (5.2). The irradiance equation is a first-order PDE. Unfortunately, in general, this PDE is nonlinear and only well posed under limited conditions. To make things worse, in practice,

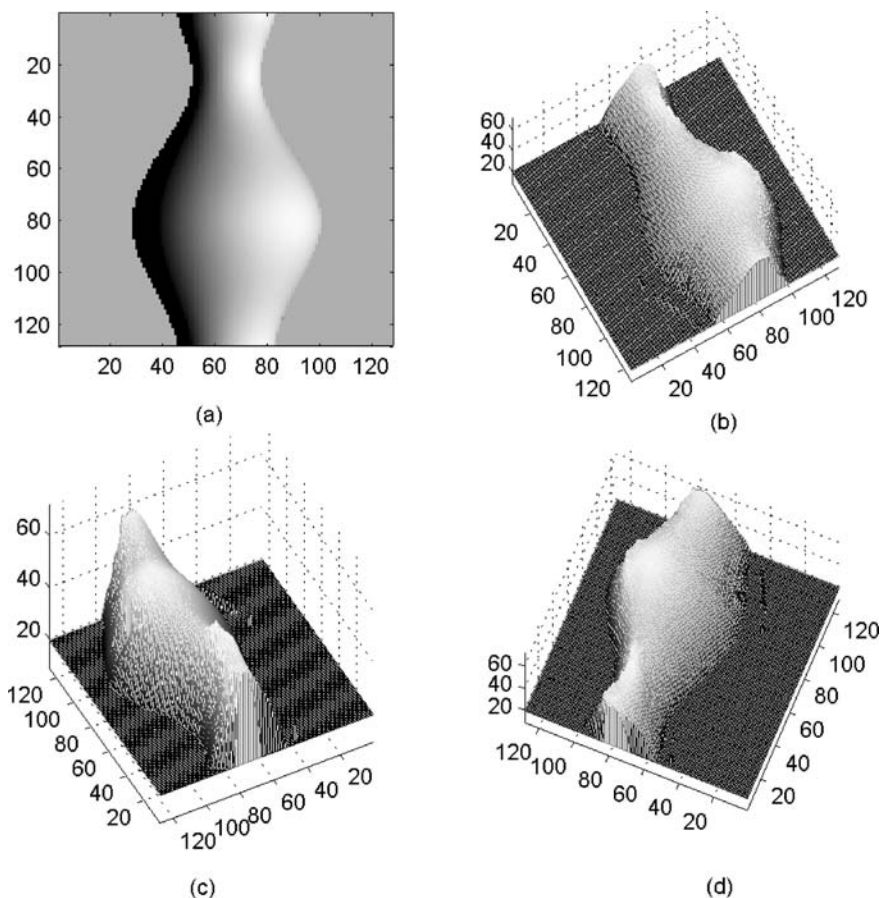


Figure 5.3: Tsai-Shah's linear SFS algorithm applied to the synthetic vase image. (a) is the input image with light source ($x = 1$, $y = 0$, $z = 1$). (b), (c), and (d) are the reconstructed surface from three different directions.

the data available for shape reconstruction is not the complete intensity function, but rather its sampled version—a discrete data set. In addition, the reflectance map is usually determined experimentally as well. Usually people believe that the problem has at least one solution, but it is clear that the uniqueness of the solution is difficult to get. The optimization approach is one of the earliest approaches that has been proposed and researched for several decades. The original work can be traced back to the Ph.D. thesis of Horn [26]. Different constraint functions (see Section 5.2.3.2) can be used to minimize the energy function. First, we consider a general way to construct

the energy function, which contains almost all the common constraints listed in Section 5.2.3.2,

$$\int \int \{(I - R)^2 + (Z_{xx}^2 + Z_{xy}^2 + Z_{yx}^2 + Z_{yy}^2) + (\|\vec{N}\|^2 - 1) + ((Z_x - p)^2 + (Z_y - q)^2) + ((R_x - I_x)^2 + (R_y - I_y)^2)\} dx dy, \quad (5.48)$$

where $\vec{N}$ is defined as the surface normal, I is the input image, R is the reflectance map, (x, y) is an arbitrary pixel of the input image, and (p, q) is orientation at pixel (x, y) . The first term, $(I - R)^2$, is called the brightness error term, which is used to minimize the brightness error between the measured image intensity and the reflectance function. The second term, $(p_x^2 + p_y^2 + q_x^2 + q_y^2)$, is called the regularization term which will always penalize large local changes in the surface orientation and encourage the surface change gradually. The third term, $(\|\vec{N}\|^2 - 1)$, is called unit normal term and is used to normalize the constraints on the recovered normal by forcing the surface normal to be unit vectors. The fourth term, $((Z_x - p)^2 + (Z_y - q)^2)$, is called integrability term which is used to ensure the valid surface. The last term, $(R_x - I_x)^2 + (R_y - I_y)^2$, is defined as the intensity gradient term. It requires that the intensity gradient of the reconstructed image be close to the intensity gradient of the input image in the x and y directions as much as possible. Sometimes, if an algorithm is designed for a particular type of images, adequate constraints should be chosen to meet some specific requirements.

In the following context we will introduce the most popular algorithm which is based on the concept of optimization.

5.3.2.1 Zheng and Chellappa's minimization method

Zheng–Chellappa [70] chose the squared brightness error term (5.14), the integrability term, and the intensity gradient term as their energy function, which is defined to be

$$\int \int ((E - R)^2 + ((R_x - I_x)^2 + (R_y - I_y)^2) + \mu((Z_x - p)^2 + (Z_y - q)^2)) dx dy. \quad (5.49)$$

Recall that most of the traditional methods enforce the requirement that the reconstructed (approximated) image should be close to the input (exact) image,

which satisfies the irradiance equation (5.2):

$$R(p, q) = I(x, y),$$

where $p = \partial Z / \partial x$ and $q = \partial Z / \partial y$, $Z(x, y)$ is the height of image at (x, y) .

Notice that, for each pixel, the right side of Eq. (5.2) is given values and in the left side p and q are free variables. Therefore, we write $p = p(x, y)$ and $q = q(x, y)$. Now we rewrite the energy equation (5.49) as

$$\text{Energy} = \iint F(p, q, Z) dx dy, \quad (5.50)$$

where $F(p, q, Z)$ is the sum of the following three parts:

$$(I - R)^2 = (R(p, q) - I(x, y))^2, \quad (5.51)$$

$$(R_x - I_x)^2 + (R_y - I_y)^2 = (R_p(p, q)p_x + R_q(p, q)q_x - I_x(x, y))^2 \quad (5.52)$$

$$+ (R_p(p, q)p_y + R_q(p, q)q_y - I_y(x, y))^2, \\ \mu((Z_x - p)^2 + (Z_y - q)^2). \quad (5.53)$$

Using the technique of calculus of variations in Section 5.2.3 to minimize the energy function (5.50) is equivalent to solving the following Euler equation:

$$F_p - \frac{\partial}{\partial x} \frac{\partial F}{\partial p_x} - \frac{\partial}{\partial y} \frac{\partial F}{\partial p_y} = 0, \quad (5.54)$$

$$F_q - \frac{\partial}{\partial x} \frac{\partial F}{\partial q_x} - \frac{\partial}{\partial y} \frac{\partial F}{\partial q_y} = 0,$$

$$F_Z - \frac{\partial}{\partial x} \frac{\partial F}{\partial Z_x} - \frac{\partial}{\partial y} \frac{\partial F}{\partial Z_y} = 0.$$

By taking the first-order terms in the Taylor series of the reflectance map, Zheng–Chellappa [70] simplified the Euler equation. For example, F_p can be approximated by the following equation:

$$F_p \approx 2[R - I(x, y)]R_p + \mu(p - Z_x). \quad (5.55)$$

From Eq. (5.55), we observe that the higher order derivatives, R_{pp}, R_{pq}, R_{qp} , and R_{qq} , are omitted because we only take the first-order Taylor expansion. Similarly, we can get F_q and F_Z and all the other variables in Eq. (5.54). Finally, we get the following iterative formula (the current values of p ,

q , and Z are updated by quantities δ_p , δ_q , and δ_Z , respectively):

$$\begin{aligned} p^{k+1} &= p^k + \delta_p, \\ q^{k+1} &= q^k + \delta_q, \\ Z^{k+1} &= Z^k + \delta_Z, \end{aligned} \quad (5.56)$$

where

$$\begin{aligned} \delta_p &= \frac{4}{\Delta} \left[\left(C_1 - \frac{1}{4} \mu C_3 \right) \left(5R_q^2 + \frac{5}{4} \mu \right) - \left(C_2 - \frac{1}{4} \mu C_3 \right) \left(5R_p R_q + \frac{1}{4} \mu \right) \right], \\ \delta_q &= \frac{4}{\Delta} \left[\left(C_1 - \frac{1}{4} \mu C_3 \right) \left(5R_q^2 + \frac{5}{4} \mu \right) - \left(C_2 - \frac{1}{4} \mu C_3 \right) \left(5R_p R_q + \frac{1}{4} \mu \right) \right], \\ \delta_Z &= \frac{1}{4} [C_3 + \delta_p + \delta_q], \end{aligned} \quad (5.57)$$

and

$$\begin{aligned} C_1 &= (-R + I + R_p p_{xx} + R_q q_{xx} - I_{xx} + R_p p_{yy} + R_q q_{yy} - I_{yy}) R_p - \mu(p - Z_x), \\ C_2 &= (-R + I + R_p p_{xx} + R_q q_{xx} - I_{xx} + R_p p_{yy} + R_q q_{yy} - I_{yy}) R_q - \mu(q - Z_y), \\ C_3 &= -p_x + Z_{xx} - q_y + Z_{yy}, \\ \Delta &= 4 \left[\left(5R_q^2 + \frac{5}{4} \mu \right)^2 - \left(5R_p R_q + \frac{1}{4} \mu \right)^2 \right]. \end{aligned} \quad (5.58)$$

In order to solve these equations, we need to know the values of $R(p, q)$, we recall the reflectance equation mentioned before (5.5):

$$R(p, q) = \rho \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2} \sqrt{1 + p^2 + q^2}}. \quad (5.59)$$

If we choose $\vec{L} = (\cos \tau \sin \gamma, \sin \tau \sin \gamma, \cos \gamma)$ as the unit vector for the illuminant direction, where τ is the tilt angle of the illuminant (the angle between the direction of the illuminant and the x - z plane), γ is the slant angle (the angle between the illuminant direction and the positive z axle). Given the input parameters ρ , τ , and γ and setting the initial value as $p^0 = q^0 = 0$, we can solve all the variables in Eq. (5.58) using the following group of equations:

$$\begin{aligned} R &= \rho \frac{\cos \gamma - p \cos \tau \sin \gamma - q \sin \tau \sin \gamma}{\sqrt{1 + p^2 + q^2}}, \\ R_p &= R(p + \delta_{pq}, q) - R(p, q), \\ R_q &= R(p, q + \delta_{pq}) - R(p, q), \\ p_x &= p(x + 1, y) - p(x, y), \end{aligned} \quad (5.60)$$

$$\begin{aligned} p_{xx} &= p(x + 1, y) + p(x - 1, y) - 2p(x, y), \\ p_{yy} &= p(x, y + 1) + p(x, y - 1) - 2p(x, y). \end{aligned} \quad (5.61)$$

Similarly, we can get all the other needed values in (5.59), namely, q_{xx} , q_y , q_{yy} , Z_x , Z_y , Z_{xx} , Z_{yy} , I_{xx} , and I_{yy} . Notice that, in (5.61), the partial derivatives p_x , p_{xx} , and p_{yy} are approximated by linear terms in their Taylor series.

In order to accelerate the computational process, the hierarchical implementation has been used in Zheng–Chellappa’s algorithm. The lowest layer of the image is 32×32 , the higher one is 64×64 , etc. For a detailed discussion about the hierarchical method and its implementation, we refer the readers to [70].

The whole algorithm can be described by the following procedure.

- *Step 1.* Estimate the original parameters of the reflectance map.
- *Step 2.* Normalize the input image. This step can be used to reduce the input image size to that of the lowest resolution layer.
- *Step 3.* Update the current shape reconstruction using Eqs. (5.56)–(5.59), and (5.61).
- *Step 4.* If the current image is in the highest resolution, the algorithm stopped. Otherwise, we will increase the image size and expand the shape reconstruction to the adjacent higher resolution layer; reduce the normalized input image to the current resolution. Then go to step 3.

The following is the pseudocodes used to realize Zheng–Chellappa’s method.

Algorithm 3: Zheng–Chellappa’s method

Input $Z_{\min}$ (mindepthvalue), $Z_{\max}$ (maxdepthvalue),
 (x, y, z) (direction of the light source), $I(x, y)$ (input image)
 $D \leftarrow \sqrt{x^2 + y^2 + z^2}$, $sx \leftarrow x/D$, $sy \leftarrow y/D$, $sz \leftarrow z/D$.
 $p^0 \leftarrow q^0 \leftarrow Z^0 \leftarrow 0$
 $\delta_{pq} \leftarrow 0.001$, $\mu \leftarrow 1.0$ (μ will be used in Eqs. (5.57) and (5.58))
 $\sin \gamma \leftarrow \sin(\arccos(lz))$, $\sin \tau \leftarrow \sin(\arctan(sy/sx))$,
 $\cos \tau \leftarrow \cos(\arctan(sy/sx))$.
for $i = 1$ to $width(I)$ do
 for $j = 1$ to $height(I)$ do
 calculate $(p_x, p_{xx}, p_y, p_{yy}, q_x, q_{xx}, q_y, q_{yy}, Z_x, Z_{xx}, Z_y, Z_{yy})$
 $R \leftarrow (\rho \cos \gamma - p(i, j) \cos \tau \sin \gamma - q(i, j) \sin \tau \sin \gamma) /$
 $\sqrt{1 + p(i, j)^2 + q(i, j)^2}$,
 $R_p \leftarrow R(p(i, j) + \delta_{pq}, q(i, j)) - R(p(i, j), q(i, j))$
 calculate $(\delta_p, \delta_q, \delta_z)$ using Eqs. (5.57) and (5.58)
 $p \leftarrow p^0 + \delta_p$, $q \leftarrow q^0 + \delta_q$
 $Z \leftarrow Z^0 + \delta_z$


```

         $p \leftarrow Z(i, j) - Z(i, j - 1)$ 
         $q \leftarrow Z(i, j) - Z(i - 1, j)$ 
    end do
end do
 $Normalize(Z(x, y), Z_{\max}, Z_{\min})$ 
Output  $Z(x, y)$ 

```

The subfunction *Normalize* is a standard math function used in signal and image processing.

We now demonstrate this method by using the following example.

Example 6. *Reconstruct the surface of a synthetic vase using the Zheng–Chellappa method.*

The experiments are based on the synthetic images that are generated using true depth maps. Figure 5.4(a) shows the same synthetic vase as in the previous section and the reconstruction results using Pentland’s algorithm. The light is from above at $(x = 0, y = 0, z = 1)$. The input image is showed in Fig. 5.4(a). The surface, shown in Figs. 5.4(b), (c), and (d), is the reconstructed depth map from three different directions. Zheng–Chellappa algorithm produces reasonable results as expected for the vase. However, some errors can be seen around the boundary of the vase. In general, the experiment shows that Zheng–Chellappa’s algorithm can reasonably reconstruct the object on the surface. The most important advantage of Zheng–Chellappa’s algorithm is that the optimization approach is not limited to the situation where the reflectance map changes linearly with respect to the surface shape.

Example 7. *Reconstruct the surface of a synthetic Mozart using Zheng–Chellappa’s method.*

The experiments are also based on the synthetic images that are generated using true depth maps. Figure 5.5(a) shows the synthetic Mozart and the reconstruction results using Zheng–Chellappa’s algorithm. The light is from above at $(x = 0, y = 0, z = 1)$. The input image is showed in Fig. 5.5 (a). The result image, shown in Figs. 5.5(b), (c), and (d), is the reconstructed depth map from three different directions. The recovered surface is well outlined as expected for the human’s head. However, the details of Mozart cannot be accurately recovered using their approach. In our opinion, this is due to the rapid changes and complexity of the input image. Although the results can be improved by prefiltering and smoothing the input image, in general, we conclude

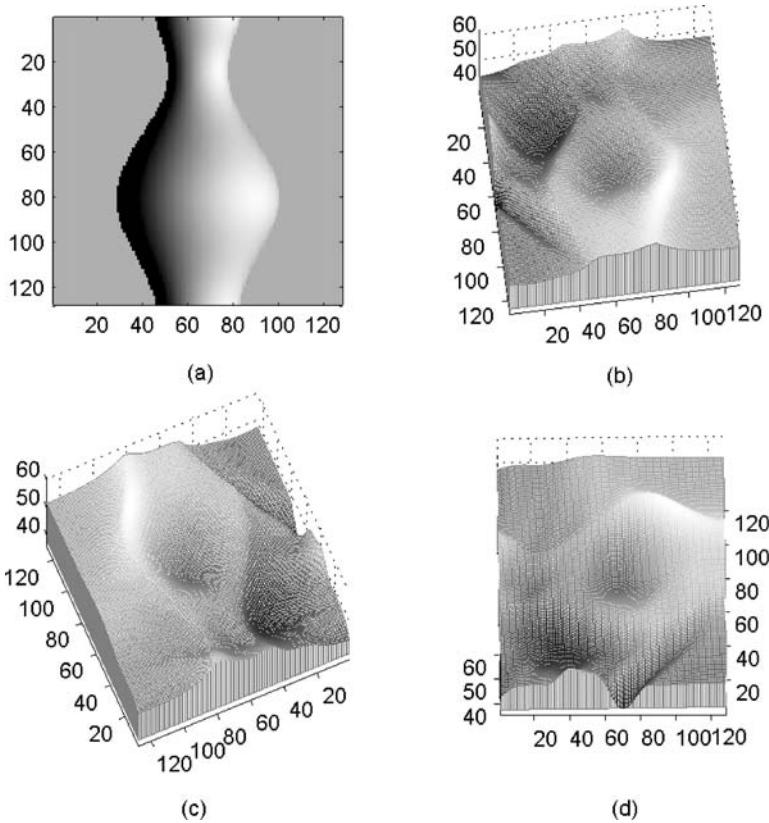


Figure 5.4: Zheng–Chellappa’s linear SFS algorithm applied to the synthetic vase image. (a) is the input image with light source ($x = 1$, $y = 0$, $z = 1$). (b), (c), and (d) are the reconstructed surface from three different directions.

from the experiment that Zheng–Chellappa’s algorithm does encounter some difficulties when the input image is complex. This observation is also true even if we used the simplest light source direction. We expect this experiment to inform the readers that SFS problem is indeed one of the most difficult problems in computer vision. No perfect, or even satisfactory, solution has been proposed yet.

We summarize this section with a few words about the advantage and disadvantage of these two methods we introduced in this section. Pentland’s method uses FFT and IFFT to calculate the depth map. This makes the algorithm relatively nonsensitive to the initial values. However, there are a few disadvantages: (1) When the light source direction and the viewing direction are similar, the Fourier transforms of p^2 and q^2 will have a doubling effect in the frequency

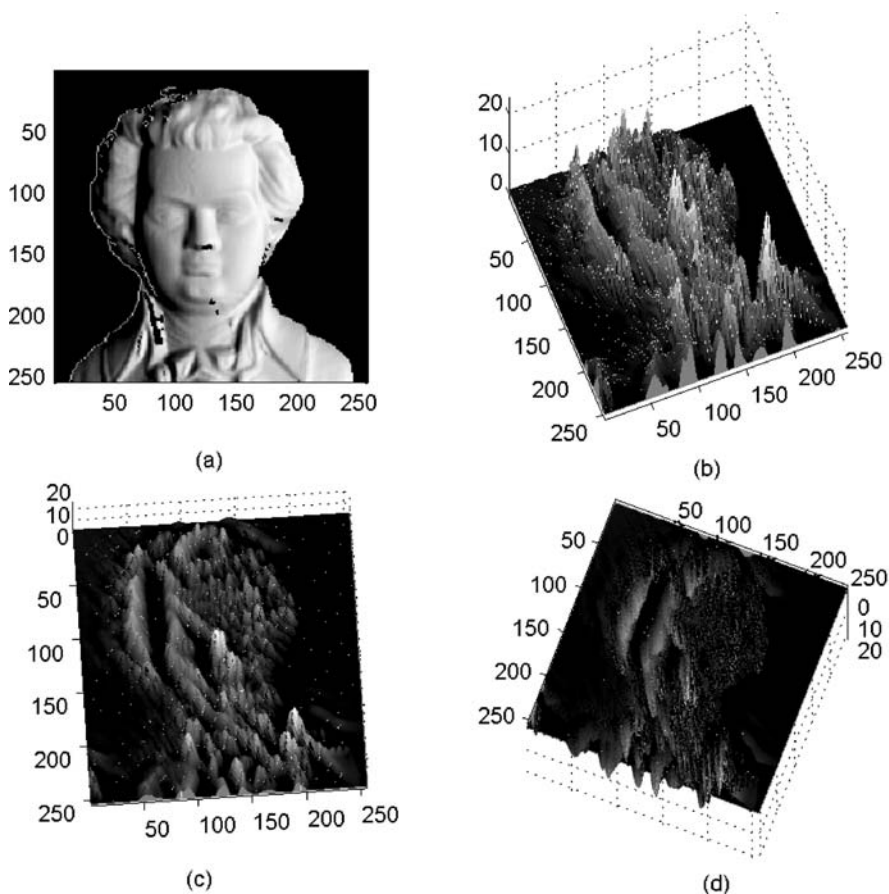


Figure 5.5: Zheng–Chellappa’s linear SFS algorithm applied to the synthetic Mozart image. (a) is the input image with light source $(x = 0, y = 0, z = 1)$. (b), (c), and (d) are the reconstructed surface from three different directions.

domain, which will affect the accuracy of the linear approximation. (2) When applying FFT and IFFT to the whole image, Pentland’s algorithm needs more time than Tsai–Shah’s approach. Tsai–Shah’s algorithm uses Newton’s method to solve the quadratic equations. When the initial value is close to the exact solution, Tsai–Shah’s algorithm converges very fast. Actually, given certain good initial values, Tsai–Shah’s algorithm needs several steps to converge. However, it is well known that Newton’s method cannot always guarantee convergence. This disadvantage makes Tsai–Shah’s approach sensitive to initial estimation than Pentland’s.

The discussion in this subsection has also shown us that the linear approach is conceptually simple. The related algorithms are relatively fast and easy to

implement. However, the reconstruction accuracy of this kind of methods is limited. The assumption of simple linear models is not quite satisfactory for the actual objects (see Section 5.2.3.2). Therefore, more advanced methods, such as multiscale methods, are introduced to overcome the disadvantage of these linear approaches. As an example, we will introduce a wavelet-based method in the following section.

To end this section, we would like to acknowledge the website <http://www.cs.ucf.edu/~vision/source.html>; all the source codes used in this section can be found in this site.

Finally, we will iterate the importance of the direction of the light source. We recall that the brightness of an object depends on the following three factors:

- (1) microstructure of the surface,
- (2) distribution of the incident light,
- (3) orientation of the surface with respect to the view and light source.

It is notable that if we change the direction of the light source, the irradiance map will be changed coordinately. This will have an impact on the convergence properties for certain numerical methods (see Section 5.3 and [64]).

5.4 Wavelet-Based Methods

Wavelet theory has been enthusiastically adopted by people in the area of signal and image processing. It has been proved to be a useful tool in many applications. A wavelet-based shape from shading method was introduced in [31]. Unlike methods introduced in Section 5.3, the objective function in the constrained optimization problem is replaced by its projection to the wavelet subspaces. To understand this approach, we first recall some elements in wavelet theory.

5.4.1 Background of Wavelets Analysis

5.4.1.1 1D Wavelets

To begin with, we present here a few elements of one-dimensional orthogonal wavelet theory, in which an orthonormal basis $\{\psi_{mn}\}$ of $L^2(R)$ is constructed

having the form

$$\psi_{mn}(t) = 2^{m/2} \psi_{mn}(2^m t - n), \quad n, m \in \mathbb{Z},$$

where $\psi(t)$ is the “mother wavelet.” Usually it is not constructed directly but rather from another function called the “scaling function” $\phi(t) \in L^2(\mathbb{R})$. The scaling function ϕ is chosen in such a way that

$$\left\{ \begin{array}{l} \text{(i)} \quad \int \phi(t) \phi(t - n) dt = \delta_{0,n}, \quad n \in \mathbb{Z}, \\ \text{(ii)} \quad \phi(t) = \sum_{k=-\infty}^{\infty} \sqrt{2} c_k \phi(2t - k), \quad \{c_k\}_{k \in \mathbb{Z}} \in l^2, \\ \text{(iii)} \quad \text{for each } f \in L^2(\mathbb{R}), \epsilon > 0, \text{ there is a function} \\ \quad f_m(t) = \sum_{n=-\infty}^{\infty} a_{mn} \phi(2^m t - n) \text{ such that } \|f_m - f\| < \epsilon. \end{array} \right. \quad (5.62)$$

These conditions lead to a “multiresolution approximation” $\{V_m\}_{m \in \mathbb{Z}}$, consisting of closed subspaces of $L^2(\mathbb{R})$. The space V_m is taken to be the closed linear span of $\{\phi(2^m t - n)\}_{n \in \mathbb{Z}}$. Because of (5.62) (ii), the V_m are nested, i.e. $V_m \subseteq V_{m+1}$ and because of (5.62) (iii), $\cup_m V_m$ is dense in $L^2(\mathbb{R})$.

There are many different types of wavelet bases created and employed for different purposes. They can be classified as time-limited wavelets, such as Haar wavelets and Daubechies wavelets, band-limited wavelets, such as Shannon and Meyer wavelets. Another standard prototype is the Haar system in which $\phi(t) = \chi_{[0,1]}(t)$, where

$$\chi_{[0,1]} = \begin{cases} 1, & x \in [0,1], \\ 0, & x \notin [0,1] \end{cases}$$

is the characteristic function of $[0, 1]$. It is an easy exercise to show that (5.62) is satisfied. This prototype has poor frequency localization but good time localization. Most of the other examples found, e.g., in [12] and [66], attempt to get fairly good time and frequency localization simultaneously.

The various scales are related by the dilation equation of the scaling function

$$\begin{aligned} \phi(t) &= \sqrt{2} \sum_{n=-\infty}^{\infty} c_n \phi(2t - n), \\ \psi(t) &= \sqrt{2} \sum_{n=-\infty}^{\infty} d_n \phi(2t - n), \end{aligned} \quad (5.63)$$

where $d_n = c_{1-n}(-1)^n$.

In addition, the Fourier transform of the mother wavelet $\psi(t)$ vanishes in a neighborhood of the origin. We denote by W_m the closed linear span of $\{\psi(2^m t - n)\}$. This set of functions form an orthogonal basis of $L^2(R)$. That is,

$$V_m = V_{m-1} \oplus W_{m-1}$$

$$L^2(R) = \bigoplus_{m=-\infty}^{\infty} W_m$$

For $f \in L^2(R)$, we have the projections onto the subspace V_m and W_m respectively given by

$$f_m(t) = P_m f(t) = \sum_{n=-\infty}^{\infty} a_{m,n} 2^{m/2} \phi(2^m t - n), \quad (5.64)$$

$$f^m(t) = P^m f(t) = \sum_{n=-\infty}^{\infty} b_{m,n} 2^{m/2} \psi(2^m t - n), \quad (5.65)$$

where

$$a_{m,n} = 2^{-m/2} \int_{-\infty}^{\infty} f(x) \phi(2^m t - n) dx,$$

$$b_{m,n} = 2^{-m/2} \int_{-\infty}^{\infty} f(x) \psi(2^m t - n) dx.$$

The coefficients $a_{j,n}$ and $b_{j,n}$ at resolution $j = m$ and $j = m + 1$ are related by a tree algorithm. To see this, we space V_1 , we have two distinct orthonormal bases:

$$\left\{ \sqrt{2} \phi(2x - n) \right\}_{n=-\infty}^{\infty}$$

and

$$\{\phi(x - n), \psi(x - k)\}_{n,k=-\infty}^{\infty}.$$

Hence each $f \in V_1$ has an expansion

$$f(x) = \sum_{n=-\infty}^{\infty} a_{1,n} \sqrt{2} \phi(2x - n)$$

$$= \sum_{n=-\infty}^{\infty} a_{0,n} \phi(x - n) + \sum_{n=-\infty}^{\infty} b_{0,n} \psi(x - n).$$

By (5.63) we have

$$a_{1,n} = \sum_{k=-\infty}^{\infty} c_{n-2k} a_{0,k} + \sum_{k=-\infty}^{\infty} (-1)^{n-1} c_{1-n+2k} b_{0,k}, \quad (5.66)$$

which is the reconstruction part. The decomposition is even easier: We need merely use the formulas for a_n^0 and b_n^0 to find

$$\begin{aligned}
 a_{0,n} &= \int_{-\infty}^{\infty} f(x)\phi(x-n)dx = \int_{-\infty}^{\infty} f(x) \sum_k c_k \sqrt{2}\phi(2x-2n-k)dx \quad (5.67) \\
 &= \sum_k c_k a_{1,2n+k} = \sum_k a_{1,k} c_{k-2n}, \\
 b_{0,n} &= \sum_k a_{1,k} (-1)^{k-1} c_{1-k+2n}.
 \end{aligned}$$

This works at each scale to give us the tree algorithm for decomposition (5.67),

$$\begin{array}{ccccccc}
 & & b_{m-1,n} & & & b_{0,n} & \\
 & \nearrow & & \nearrow & \nearrow & \nearrow & \nearrow \\
 \cdots & \longrightarrow & a_{m,n} & \longrightarrow & a_{m-1,n} & \longrightarrow & \cdots \longrightarrow a_{1,n} \longrightarrow a_{0,n} \longrightarrow \cdots
 \end{array}$$

and for reconstruction (5.66),

$$\begin{array}{ccccccc}
 & b_{0,n} & b_{1,n} & & & b_{m-1,n} & \\
 & \searrow & \searrow & \searrow & \searrow & \searrow & \\
 \cdots & \longrightarrow & a_{0,n} & \longrightarrow & a_{1,n} & \longrightarrow & \cdots \longrightarrow a_{m-1,n} \longrightarrow a_{m,n} \longrightarrow \cdots
 \end{array}$$

Thus we need calculate the coefficients from the function $f(t)$ only once at the finest scale of interest. Then we work down to successively coarser scales by using this decomposition algorithm, with the error at each successive scale corresponding to the wavelet coefficients. These algorithms are called Mallat algorithms (see [39]).

5.4.1.2 2D Separable Wavelets

In order to represent an image using wavelet bases, we need to construct a basis for $L_2(R^2)$. There are two different methods to do so. One way is based on the multiresolution analysis in 2D space to construct 2D wavelet basis directly, while another way is based on the tensor product of the 1D wavelets. The former usually leads to a nonseparable basis, while the latter derives a separable basis. Here we merely consider the separable basis, which is based on the separable multiresolution analysis of $L_2(R^2)$.

Let $\{V_m\}$ be a multiresolution of $L_2(R)$; a separable two-dimensional multiresolution is composed of the tensor product spaces

$$V_m^2 = V_m \otimes V_m.$$

The space V_m^2 is the set of the finite energy functions that are linear expansions of the set of the separable bases $\{\phi_{m,k,l}(x, y)\}_{k,l=0}^\infty$, while the correspondent wavelet subspace W_m^2 is given by the close linear span of

$$\{\phi_{m,k}(x)\psi_{m,l}(y), \psi_{m,l}(x)\phi_{m,k}(y), \psi_{m,k}(x)\psi_{m,l}(y)\}_{k,l=0}^\infty$$

where

$$\begin{aligned}\phi_{m,k}(x) &:= 2^{\frac{m}{2}} \phi(2^m x - k), \\ \psi_{m,k}(x) &:= 2^{\frac{m}{2}} \psi(2^m x - k), \\ \phi_{m,k,l}(x, y) &:= \phi_{m,k}(x) \phi_{m,l}(y).\end{aligned}\tag{5.68}$$

Like in 1D case, we have

$$\begin{aligned}V_m^2 &= V_{m-1}^2 \oplus W_{m-1}^2 = (V_m \otimes V_m) \oplus W_{m-1}^2, \\ W_m^2 &= (V_m \otimes W_m) \oplus (W_m \otimes V_m) \oplus (W_m \otimes W_m),\end{aligned}$$

and

$$L_2(R^2) = \bigoplus_{m=-\infty}^\infty W_m^2.$$

Wells *et al.* [67] proved the following theorem.

Theorem 8 (Wells and Zhou). Assume the function $f \in C^2(\Omega)$, where Ω is a bounded open set in R^2 . Let

$$f_m(x, y) := \frac{1}{2^m} \sum_{k,l \in \Lambda} f\left(\frac{k+c}{2^j}, \frac{l+c}{2^j}\right) \phi_{m,k}(x) \phi_{m,l}(y), \quad x, y \in \Omega, \tag{5.69}$$

where $\Lambda = \{k \in Z : \text{supp}(\phi_{m,k}) \cap \Omega \neq \emptyset\}$ is the index set and

$$c = \frac{1}{\sqrt{2}} \sum_{k=0}^{2N-1} kc_k.$$

Then

$$\|f - f_m\|_{L^2(\Omega)} \leq C(1/2^m)^2,$$

where C is dependent on the diameter of Ω , the first and second moduli of the first- and second-order derivatives of f on $\bar{\Omega}$.

Formula (5.69) is the one which was used in the wavelet-based SFS method. Now we are ready to introduce this method.

5.4.2 The Wavelet-Based SFS

A wavelet-based method was developed in [31]. Instead of using the constraints in Zheng–Chellappa’s method (see Section 5.3.2.1), the authors introduced a new constraint (5.20). It is said that “the new constraint not only enforces integrability but also introduces a smoothness constraint in an implicit manner.” Now the energy function is defined as

$$W = \int \int_{\Omega} [(E(x, y) - R(p, q))^2 + (p_x^2 + p_y^2 + q_x^2 + q_y^2) + ((z_x - p)^2 + (z_y - q)^2) + (z_{xx} - p)^2 + (z_{yy} - q)^2] dx dy. \quad (5.70)$$

The objective function is first replaced by its approximation in scaling space V_0 of Daubechies wavelets. Then the variational problem is solved by an iterative algorithm. We now describe this method.

We assume that the given image size is $M \times M$. The surface $Z(x, y)$, its partial derivatives $\frac{\partial Z}{\partial x} = p(x, y)$, and $\frac{\partial Z}{\partial y} = q(x, y)$ have projection to V_0 , the scaling space at level 0:

$$\begin{aligned} Z(x, y) &= \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} Z_{k,l} \phi_{0,k,l}(x, y), \\ p(x, y) &= \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} p_{k,l} \phi_{0,k,l}(x, y), \\ q(x, y) &= \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} q_{k,l} \phi_{0,k,l}(x, y). \end{aligned} \quad (5.71)$$

Denoting

$$\begin{aligned} \phi_{0,k,l}^{(x)}(x, y) &= \frac{\partial}{\partial x} \phi_{0,k,l}(x, y), & \phi_{0,k,l}^{(xx)}(x, y) &= \frac{\partial^2}{\partial x^2} \phi_{0,k,l}(x, y), \\ \phi_{0,k,l}^{(y)}(x, y) &= \frac{\partial}{\partial y} \phi_{0,k,l}(x, y), & \phi_{0,k,l}^{(yy)}(x, y) &= \frac{\partial^2}{\partial y^2} \phi_{0,k,l}(x, y), \end{aligned}$$

substitute (5.71) in each term of (5.70) to get

$$\begin{aligned} W &= \int \int_{\Omega} \left[E(x, y) - R\left(\sum_{k,l=0}^{M-1} p_{k,l} \phi_{0,k,l}(x, y), \sum_{k,l=0}^{M-1} q_{k,l} \phi_{0,k,l}(x, y)\right) \right]^2 dx dy \\ &\quad + \int \int_{\Omega} \left(\left(\sum_{k,l=0}^{M-1} p_{k,l} \phi_{0,k,l}^{(x)}(x, y) \right)^2 + \left(\sum_{k,l=0}^{M-1} p_{k,l} \phi_{0,k,l}^{(y)}(x, y) \right)^2 \right. \end{aligned} \quad (5.72)$$

$$\begin{aligned}
& + \left(\sum_{k,l=0}^{M-1} q_{k,l} \phi_{0,k,l}^{(x)}(x, y) \right)^2 + \left(\sum_{k,l=0}^{M-1} q_{k,l} \phi_{0,k,l}^{(y)}(x, y) \right)^2 dx dy \\
& + \int \int_{\Omega} \left(\left(\sum_{k,l=0}^{M-1} Z_{k,l} \phi_{0,k,l}^{(xx)}(x, y) - \sum_{k,l=0}^{M-1} p_{k,l} \phi_{0,k,l}^{(xx)}(x, y) \right)^2 \right. \\
& + \left. \left(\sum_{k,l=0}^{M-1} Z_{k,l} \phi_{0,k,l}^{(yy)}(x, y) - \sum_{k,l=0}^{M-1} q_{k,l} \phi_{0,k,l}^{(yy)}(x, y) \right)^2 \right) dx dy \\
& + \int \int_{\Omega} \left(\left(\sum_{k,l=0}^{M-1} Z_{k,l} \phi_{0,k,l}^{(xy)}(x, y) - \sum_{k,l=0}^{M-1} p_{k,l} \phi_{0,k,l}^{(xy)}(x, y) \right)^2 \right. \\
& + \left. \left(\sum_{k,l=0}^{M-1} Z_{k,l} \phi_{0,k,l}^{(yx)}(x, y) - \sum_{k,l=0}^{M-1} q_{k,l} \phi_{0,k,l}^{(yx)}(x, y) \right)^2 \right) dx dy.
\end{aligned}$$

There are total of $3M^2$ unknown variables (they are the function samples of Z , p , and q):

$$\{p_{k,l}\}, \{q_{k,l}\}, \text{ and } \{Z_{k,l}\},$$

where the indices run on $M \times M$ grid (see (5.69)).

It is remarkable that the interpolating property (5.69) simplified the computation significantly. The integrals we need to compute in energy function are only involved with the integrals which are the inner product of the scaling function $\phi(x, y) := \phi_{0,0,0}(x, y)$, its shifting $\phi_{k,l}(x, y) := \phi_{0,k,l}(x, y)$, and their partial derivatives of first and second orders. Note that we have dropped the scale (or the resolution) index 0 for simplicity, since the discussion here does not relate to other scales. Now we assume that the scaling function ϕ is the Daubechies scaling function with $2N + 1$ filter coefficients c_i (see (5.63)). These definite integrals are called connection coefficients [5]:

$$\begin{aligned}
\Gamma_x^{(4)}(k, l) &= \int \int_{\Omega} \phi^{(xx)}(x, y) \phi_{k,l}^{(xx)}(x, y) dx dy = \Gamma^{(4)}(k) D(l), \\
\Gamma_y^{(4)}(k, l) &= \int \int_{\Omega} \phi^{(yy)}(x, y) \phi_{k,l}^{(yy)}(x, y) dx dy = D(k) \Gamma^{(4)}(k), \\
\Gamma_{xy}^{(4)}(k, l) &= \int \int_{\Omega} \phi^{(xy)}(x, y) \phi_{k,l}^{(xy)}(x, y) dx dy = \Gamma^{(2)}(k) \Gamma^{(2)}(l),
\end{aligned}$$

$$\begin{aligned}
\Gamma_{yx}^{(4)}(k, l) &= \int \int_{\Omega} \phi^{(yx)}(x, y) \phi_{k,l}^{(yx)}(x, y) dx dy = \Gamma^{(2)}(k) \Gamma^{(2)}(l), \\
\Gamma_x^{(3)}(k, l) &= \int \int_{\Omega} \phi^{(xx)}(x, y) \phi_{k,l}^{(x)}(x, y) dx dy = D(l) \Gamma^{(3)}(k), \\
\Gamma_y^{(3)}(k, l) &= \int \int_{\Omega} \phi^{(y)}(x, y) \phi_{k,l}^{(yy)}(x, y) dx dy = D(k) \Gamma^{(3)}(l), \\
\Gamma_x^{(2)}(k, l) &= \int \int_{\Omega} \phi^{(x)}(x, y) \phi_{k,l}^{(x)}(x, y) dx dy = D(l) \Gamma^{(2)}(k), \\
\Gamma_y^{(2)}(k, l) &= \int \int_{\Omega} \phi^{(y)}(x, y) \phi_{k,l}^{(y)}(x, y) dx dy = D(k) \Gamma^{(2)}(l), \\
\Gamma_x^{(1)}(k, l) &= \int \int_{\Omega} \phi^{(x)}(x, y) \phi_{k,l}(x, y) dx dy = D(l) \Gamma^{(1)}(k), \\
\Gamma_y^{(1)}(k, l) &= \int \int_{\Omega} \phi^{(y)}(x, y) \phi_{k,l}(x, y) dx dy = D(k) \Gamma^{(1)}(l),
\end{aligned}$$

where

$$\begin{aligned}
\Gamma^{(1)}(k) &= \int \phi^{(x)}(x) \phi(x-k) dx, & \Gamma^{(2)}(k) &= \int \phi^{(x)}(x) \phi^{(x)}(x-k) dx, \\
\Gamma^{(3)}(k) &= \int \phi^{(xx)}(x) \phi^{(x)}(x-k) dx, & \Gamma^{(4)}(k) &= \int \phi^{(xx)}(x) \phi^{(xx)}(x-k) dx
\end{aligned}$$

are 1D connection coefficients and $D(0) = 1$, $D(n) = 0$, $n \neq 1$. Note that since the 2D basis here is constructed from the tensor product of 1D basis, these 2D connection coefficients can be computed by using 1D coefficients. We also notice that these connection coefficients are independent of the input images; therefore, they only need to be computed once.

The energy function is then linearized by taking the linear term in its Taylor expansion at (p, q) . The next step is to solve the optimization problem associated with the linearized energy function by iterations. Let $\delta p_{i,j}$, $\delta q_{i,j}$, and $\delta z_{i,j}$ be the small variation of $p_{i,j}$, $q_{i,j}$, and $z_{i,j}$, respectively, and set

$$\frac{\partial \delta W}{\partial \delta p_{i,j}} = \frac{\partial \delta W}{\partial \delta q_{i,j}} = \frac{\partial \delta W}{\partial \delta z_{i,j}} = 0.$$

We obtain

$$\begin{aligned}
\delta p_{i,j} &= [C_1 D_{22} - C_2 \frac{\partial R}{\partial p}(i, j) \frac{\partial R}{\partial q}(i, j)] / D, \\
\delta q_{i,j} &= [C_2 D_{11} - C_1 \frac{\partial R}{\partial p}(i, j) \frac{\partial R}{\partial q}(i, j)] / D, \\
\delta z_{i,j} &= C_3 / D_{33},
\end{aligned} \tag{5.73}$$

where

$$\begin{aligned}
 D_{11} &= R_{p_{i,j}}^2 + 3\Gamma^{(2)}(0) + 1, \\
 D_{22} &= R_{q_{i,j}}^2 + 3\Gamma^{(2)}(0) + 1, \\
 D_{33} &= 2\Gamma^{(2)}(0) + 2\Gamma^{(4)}(0), \\
 D &= D_{11}D_{22} - R_{p_{i,j}}^2 R_{q_{i,j}}^2
 \end{aligned} \tag{5.74}$$

and

$$\begin{aligned}
 C_1 &= (E - R)R_p - p_{i,j} \\
 &\quad + \sum_{k=-2N+2}^{2N-2} Z_{i-k,j}(\Gamma^{(3)}(k) + \Gamma^{(1)}(k)) - (2p_{i-k,j} + p_{i,j-k})\Gamma^{(2)}(k), \\
 C_2 &= (E - R)R_q - q_{i,j} \\
 &\quad + \sum_{k=-2N+2}^{2N-2} Z_{i,j-k}(\Gamma^{(3)}(k) + \Gamma^{(1)}(k)) - (q_{i-k,j} + 2q_{i,j-k})\Gamma^{(2)}(k), \\
 C_3 &= - \sum_{k=-2N+2}^{2N-2} (p_{i-k,j} + q_{i,j-k})(\Gamma^{(3)}(k) + \Gamma^{(1)}(k)) \\
 &\quad + (Z_{i-k,j} + Z_{i,j-k})(\Gamma^{(2)}(k) + \Gamma^{(4)}(k)).
 \end{aligned} \tag{5.75}$$

Finally, we can write the iterative formula

$$\begin{aligned}
 p_{i,j}^{m+1} &= p_{i,j}^m + \delta p_{i,j}, \\
 q_{i,j}^{m+1} &= q_{i,j}^m + \delta q_{i,j}, \\
 z_{i,j}^{m+1} &= z_{i,j}^m + \delta z_{i,j}.
 \end{aligned} \tag{5.76}$$

We now summarize this method as the follows:

Step 0. Compute 1D connection coefficients and 2D connection coefficients.

Step 1. Compute the set of coefficients (5.75) and (5.74).

Step 2. Compute the set of variations $\delta p_{i,j}$, $\delta q_{i,j}$, and $\delta z_{i,j}$ (5.73).

Step 3. Update the current $(p_{i,j}^m, q_{i,j}^m)$ and then the current shape reconstruction $Z_{i,j}^m$ using Eq. (5.76).

5.4.3 Summary

The wavelet-based method we demonstrated in this section is based on the approximation of the objective function in V_0 . It should be pointed out that it

did not use the multiscale structure possessed by the wavelet bases, nor the Mallat algorithm to speed up the computation. Since the selected wavelet bases are time-limited (therefore it is not band-limited), it may be not the best choice for approximating differential operators.

At this point, we would like to mention the idea of regularization. The shape from shading problems can be regarded as inverse problems since they attempt to recover physical properties of a 3D surface from a 2D image associated with the surface. Therefore, the Tikhonov regularization approach can be applied to this problem. The time-limited filters, such as the difference boxes [22] or the Daubechies wavelets used in Section 5.4.2, do not satisfy one of the conditions requested by the Tikhonov regularization [61]. In contrast with time-limited filters, band-limited filters are commonly used for regularizing differential operators, since the simplest way to avoid harmful noise is to filter out high frequencies that are amplified by differentiation. Meyer wavelet family constitutes an interesting class of such type of band-limited filters. The ill-posedness/ill-conditionness of the SFS model and its connection to the regularization theory have been discussed in [7]. Minimization (5.21) will lead to a smoother solution (the regularization solution). In some cases, the Lagrange multipliers are the “regularizers.” However, the numerical experiments presented in Section 5.3 are treated by choosing those regularizers equal to 1. The nonlinear ill-posed problems are quite difficult and basically no general approaches seem to exist [7]. For the classic theory of regularization, we highly recommend Tikhonov *et al.* [60].

A 2D basis constructed from the tensor product of 1D wavelet basis is much easier to compute than the nonseparable wavelets. There is also some ongoing research on nonseparable wavelets for use in image processing. For a detailed discussion on nonseparable wavelets, we recommend [37, 38, 40] and references therein.

The development of a wavelet-based method which reflects the multiscale nature with an effective algorithm, namely, using Mallat algorithm, is still an open problem.

5.5 Concluding Remarks

In this chapter, we have given a super brief introduction of the shape from shading problems. A variety of elementary numerical techniques related to solution

of this problem is discussed and implemented to show the basic ideas. However, a short chapter like this one has to omit many related topics, which are both important and exciting. In fact, there are many other techniques and advanced developments in the area. Fortunately, most of them are very well documented in the literature. For instance, the following two approaches reflect different flavors:

1. *Statistical learning and neural network*. [2] introduced a statistical method to solve the SFS model; the principal component analysis (PCA) was used to derive a low-dimensional parameterization of head shape space, and an algorithm was presented for solving shape from shading based on this approach.
2. *Fast matching method*. The schemes are of use in a variety of applications, including problems of shape from shading. An excellent review about this method is given by its pioneer [54]. Applications related to vision problems can be found in [55] and [53].

We conclude this chapter by pointing out that there is, in general, no proof of the convergence for the numerical methods introduced in Sections 5.3 and 5.4. An interesting example related to this topic can be found in [30].

5.6 Acknowledgements

The authors would like to thank Dr. Gilbert G. Walter for his encouragement and his valuable suggestions which led to significant improvement of this paper. The first author was partially supported by Professor Naoki Saito's grant ONR YIP N00014-00-1-0469 while completing this paper. She also wishes to thank Dr. Jianbo Gao for introducing her the reference [31].

Bibliography

- [1] Ascher, U. M. and Carter, P. M., A multigrid method for shape from shading, *SIAM J. Numer. Anal.* Vol. 30, No. 1, pp. 102–115, 1993.
- [2] Atick, J. J., Griffin, P. A., and Redlich, A. N., Statistical approach to shape from shading: Reconstruction of 3D face surfaces from single 2D images, *Neural Comput.*, Vol. 8, pp. 1321–1340, 1996.
- [3] Bakhvalov, N. S., On the convergence of a relaxation method with natural constraints on the elliptic operator, *USSR Comput. Math. Phys.*, Vol. 6, pp. 101–135, 1966.
- [4] Barnes, I. and Zhang, K., Instability of the Eikonal equation and shape from shading, *M2AN Math. Model. Numer. Anal.* Vol. 34, No. 1, pp. 127–138, 2000.
- [5] Beylkin, G., On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.*, Vol. 29, pp. 1716–1740, 1992.
- [6] Bichsel, M. and Pentland, A. P., A simple algorithm for shape from shading, *IEEE Proc. Comput. Vis. Pattern Recognit.*, pp. 459–465, 1992.
- [7] Bertero, M., Poggio, T. A., and Torre, V., Ill-posed problems in early vision, *Proc. IEEE*, Vol. 76, No. 8, pp. 869–889, 1988.
- [8] Briggs, W. L., Henson, V. E., and McCormick, S. F., *A Multigrid Tutorial*, 2nd edn., Society for Industrial and Applied Mathematics, 193 pp. c 2000.
- [9] Choe, Y. and Kashyap, R. L., 3-D shape from a shading and textural surface image, *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 13, No. 9, pp. 907–999, 1999.
- [10] Chabrowski, J. and Zhang, K., On shape from shading problem, In: *Functional Analysis, Approximation Theory and Numerical Analysis*, World Scientific Publishing, River Edge, NJ, pp. 93–105, 1994.
- [11] Courant, R. and Hilbert, D., *Methods of Mathematical Physics*, 1st edn., Vol. 1, Wiley-Interscience, New York, 560 pp., 1989.

- [12] Daubechies, I, Ten Lectures on Wavelets, CBMS-NSF Series in Appl. Math., SIAM Publications, Philadelphia, PA, 1992.
- [13] Debnath, L., Integral Transforms and Their Applications, CRC Press, Boca Raton, FL, Vol. xi, 457 pp., c1995.
- [14] Deng, Y. and Li, J., Existence and uniqueness in shape from shading, J. Comput. Sci. Technol. Vol. 12, No. 1, pp. 58–64, 1997.
- [15] Duffy, D. G., Transform Methods for Solving Partial Differential Equations, CRC Press, Boca Raton, FL, 1994.
- [16] Dupuis, P. and Oliensis, J., An optimal control formulation and related numerical methods for a problem in shape reconstruction, Ann. Appl. Probab., Vol. 4, No. 2, pp. 287–346, 1994.
- [17] Dupuis, P. and Oliensis, J., Direct method for reconstructing shape from shading, IEEE Computer Society Conference proceedings on CVPR '92, Comput. Vis. Pattern Recognit., pp. 453–458, 1992.
- [18] Durou, J. D. and Piau, D., Ambiguous shape from shading with critical points, J. Math. Imaging Vis., Vol. 12, No. 2, pp. 99–108, 2000.
- [19] Evans, G., Blackledge, J. M., and Yardley, P. D., Analytic Methods for Partial Differential Equations, Springer-Verlag, New York, 1999.
- [20] Evans, L., Partial Differential Equations, Series of graduate studies in mathematics, Vol. 19, 662 pp., American Mathematical Society, Providence, RI, 1998.
- [21] Ewing, G. M., Calculus of Variations with Applications, 352 pp., Dover Publications, New York, 1985.
- [22] Herskovitz, A. and Binford, T. O., On boundary detection, Artificial Intelligence Laboratory Memo 183, Massachussetts Institute of Technology, Cambridge, MA, 1980.
- [23] Hackbusch, W., Multi-Grid Methods and Applications, Springer, Berlin, 1985.
- [24] Gautschi, W., Numerical Analysis: An Introduction, 506 pp., Birkhauser, Boston, 1997.

- [25] Harris, J. W. and Stocker, H., *Handbook of Mathematics and Computational Science*, Springer-Verlag, Berlin, 1998.
- [26] Horn, B. K. P., *Shape from shading: A Method of Obtaining the Shape of a Smooth Opaque Object from One View*, Ph.D Thesis, Massachusetts Inst. of Technology, 1970.
- [27] Horn, B. K. P. and Brooks, M. J., *The variational approach to shape from shading*, MIT A. T. Memo 813, 1985.
- [28] Horn, B. K. P., *Robot Vision*, MIT Eng. Comput Sci. Ser., McGraw-Hill, New York, 1986.
- [29] Horn, B. K. P. and Brooks, M. J. (eds.), *Shape from Shading*, MIT Press, Cambridge, MA, 1989.
- [30] Horn, B. K. P., Szeliski, R. S., and Yuille, A. L., *Impossible shaded images*, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 15, No. 2, pp. 166–170, 1993.
- [31] Hsieh, J., Liao, H., Ko, M., and Fan, K., *Wavelet-based shape from shading*, *Graph. Models Image Process.*, Vol. 57, No. 4, pp. 343–362, 1995.
- [32] Hurt, N., *Mathematical methods in shape-from-shading: A review of recent results*, *Acta Appl. Math.* Vol. 23, pp. 163–188, 1991.
- [33] Ikeuchi, K. and Horn, B. K. P., *Numerical shape from shading and occluding boundaries*, *Artif. Intell.*, Vol. 17, pp. 141–184, 1981.
- [34] Kozera, R., *Uniqueness in shape from shading revisited*, *J. Math. Imaging Vision* Vol. 7, No. 2, pp. 123–138, 1997.
- [35] Lee, K. M. and Kuo, C. J., *shape from shading with a linear triangular element surface model*, *IEEE Trans. Patten Anal. Mach. Intell.*, Vol. 15, No. 8, pp. 815–822, 1993.
- [36] Lee, C. H. and Rosenfeld, A., *Improved methods of estimating shape from shading using the light source coordinate system*, *Artif. Intell.*, Vol. 26, pp. 125–143, 1985.
- [37] Lin, E.-B. and Ling, Y., *Image compression and denoising via nonseparable wavelet approximation*, *J. Comput. Appl. Math.*, Vol. 155, No. 1, pp. 131–152, 2003.

- [38] Lin, E.-B. and Ling, Y., 2-D nonseparable multiscaling function interpolation and approximation with an arbitrary dilation matrix, *Commun. Nonlinear Sci. Numer. Simul.*, Vol. 5, No. 3, pp. 125–133, 2000.
- [39] Mallat, S., *A Wavelet Tour of Signal Processing*, Academic Press, New York, 1998.
- [40] Mendivil, F. and Piché, D., Two algorithms for non-separable wavelet transforms and applications to image compression, In: *Fractals: Theory and applications in Engineering*, Springer, London, pp. 325–345, 1999.
- [41] Oliensis, J., Shape from shading as a partially well-constrained problem, *Comput. Vis., Graph., Image Process.* Vol. 54, pp. 163–183, 1991.
- [42] Oliensis, J., Uniqueness in shape from shading, *Int. J. Comput. Vis.* Vol. 6, pp. 75–104, 1991.
- [43] Neumaier, A., Solving ill-conditioned and singular linear systems: A tutorial on regularization, *SIAM Rev.*, Vol. 40, No. 3 (Sep., 1998), pp. 636–666.
- [44] Ortega, J. M. and Rheinboldt, W. C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970. Reprinted as *Classics in Applied Mathematics*, Vol. 30, SIAM, Publications, Philadelphia, PA, 2000.
- [45] Peleg, S. and Ron, G., Nonlinear multiresolution: A shape-from-shading example, *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 11, No. 2, pp. 198–206, 1989.
- [46] Pentland, A. P., Local analysis of the image, *IEEE Trans. Pattern Anal. Mach. Recognit.*, Vol. 6, pp. 170–187, 1984.
- [47] Pentland, A. P., Linear shape-from-shading, *Int. J. Comput. Vis.*, Vol. 4, pp. 153–162, 1999.
- [48] Pong, T. C., Haralick, R. M., and Shapiro, L. G., Shape from shading using the facet model, *Pattern Recognit.* Vol. 22, No. 6, pp. 683–695, 1989.
- [49] Poggio, T., Torre, V., and Koch, C., Computational vision and regularization theory, *Nature*, Vol. 317, pp. 314–319, 1985.

- [50] Rouy, E. and Tourin, A., A viscosity solutions approach to shape from shading, *SIAM J. Numer. Anal.*, Vol. 29, No. 3, pp. 867–884, 1992.
- [51] Saxberg, B. V. H., A modern differential geometric approach to shape from shading, MIT AI Lab, Tech. Rep. 1117, 1989.
- [52] Saxberg, B. V. H., Existence and Uniqueness for shape from shading around critical points, *Theory Algorithm*, Vol. IJRR (11), pp. 202–224, 1992.
- [53] Sethian, J. A., Evolution, implementation, and application of level set and fast marching methods for advancing fronts, *J. Comput. Phys.*, Vol. 169, No. 2, pp. 503–555, 2001.
- [54] Sethian, J. A., Fast marching methods, *SIAM Rev.* Vol. 41, No. 2, pp. 199–235 (electronic), 1999.
- [55] Sethian, J. A., *Level Set Methods and Fast Marching Methods*, 2nd edn., Cambridge Monographs on Applied and Computational Mathematics, 3, Cambridge University Press, Cambridge, xx+378 pp., 1999.
- [56] Stuben, K. and Trottenber, U., Multigrid methods: Fundamental algorithms, model problem analysis and applications, In: *Multigrid Methods: Proceedings of the Conference*, Kolm-Porz, Springer-Verlag, Berlin, pp. 1–176, 1982.
- [57] Sweldens, W. and Roose, D., Shape from Shading Using Parallel Multigrid Relaxation, *Multigrid Methods, III* (Bonn, 1990), *Internat. Ser. Numer. Math.*, Vol. 98, Birkhäuser, Basel, pp. 353–364, 1991.
- [58] Terzopoulos, D., Image analysis using multigrid relaxation methods, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, pp. 129–139, 1986.
- [59] Tikhomirov, V. M., *Stories about Maxima and Minima*. Translated from the Russian by Abe Shenitzer, American Mathematical Society, Providence, RI, 1990.
- [60] Tikhonov, A. N. and Arsenin, V. A., *Solutions of Ill-Posed Problems*, Winston & Sons, Washington, D.C., 1977.

- [61] Tikohonov, A. N. and Samariskii, A. A., *Equations of Mathematical Physics*, Macmillan, London, 765 pp., 1963.
- [62] Trottenberg, U., Schuller, A., and Oosterlee, C., *Multigrid*, 1st edn., Academic Press, New York, 2000.
- [63] Tsai, P. S. and Shah, M., Shape form shading using linear approximation, *Image Vis. Comput.*, Vol. 12, No. 8, pp. 487–498, 1994.
- [64] Ulich, G., Provably convergent methods for the linear and nonlinear shape from shading problem, *J. Math. Imaging Vis.*, Vol. 9, No. 1, pp. 69–82, 1998.
- [65] Wei, T. and Klette, R., Theoretical analysis of finite difference algorithms for linear shape from shading, In: *Proceedings of Computer Analysis of Images and Patterns: 9th International Conference, CAIP 2001 Warsaw, Poland, September 5–7, 2001*, W. Skarbek (Ed.), *Lecture Notes in Computer Science*, Vol. 2124, Springer-Verlag, Heidelberg, pp. 638–645, 2001.
- [66] Walter, G. G. and Shen, X., *Wavelets and Other Orthogonal System*, 2nd edn., CRC Press, Boca Raton, FL, 2000.
- [67] Wells, R. O. and Zhou, X., Wavelet interpolation and approximate solutions of elliptic partial differential equations, *Technical Report*, Computational Mathematics Laboratory, Rice University, 1993.
- [68] Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M., Shape from shading: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 21, No. 8, pp. 690–706, 1999.
- [69] Zhao, W., Chellappa, R., Rosenfeld, A., and Phillips, P. J., Face recognition: A literature survey. Available at <http://citeseer.nj.nec.com/zhao00face.html>, 2000.
- [70] Zheng, Q. and Chellappa, R., Estimation of illuminant direction, albedo, and shape from shading, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 13, No. 7, pp. 680–702, 1991.

Chapter 6

Wavelets in Medical Image Processing: Denoising, Segmentation, and Registration

Yinpeng Jin¹, Elsa Angelini¹, and Andrew Laine¹

6.1 Introduction

Wavelets have been widely used in signal and image processing for the past 20 years. Although a milestone paper by Grossmann *et al.* [3] was considered the beginning of modern wavelet analysis, similar ideas and theoretical bases can be found back in the early twentieth century [4]. Following two important papers in the late 1980s by Mallat [5] and Daubechies [6], more than 9000 journal papers and 200 books related to wavelets have been published [7].

Wavelets were first introduced to medical imaging research in 1991 in a journal paper describing the application of wavelet transforms for noise reduction in MRI images [8]. Ever since, wavelet transforms have been successfully applied to many topics including tomographic reconstruction, image compression, noise reduction, image enhancement, texture analysis/segmentation, and multiscale registration. Two review papers, in 1996 [9] and 2000 [10], provide a summary and overview of research works related to wavelets in medical image processing from the past few years. Many related works can also be found in the book edited by Aldroubi *et al.* [11]. More currently, a special issue of *IEEE Transactions on Medical Imaging* [7] provides a large collection of most recent research works using wavelets in medical image processing.

The purpose of this chapter is to summarize the usefulness of wavelets in various problems of medical imaging. The chapter is organized as follows. Section 6.2

¹Department of Biomedical Engineering, Columbia University, New York, NY, USA

overviews the theoretical fundamentals of wavelet theory and related multiscale representations. As an example, the implementation of an overcomplete dyadic wavelet transform will be illustrated. Section 6.3 includes a general introduction of image denoising and enhancement techniques using wavelet analysis. Sections 6.4 and 6.5 summarize the basic principles and research works in literature for wavelet analysis applied to image segmentation and registration.

6.2 Wavelet Transform and Multiscale Analysis

One of the most fundamental problems in signal processing is to find a suitable representation of the data that will facilitate an analysis procedure. One way to achieve this goal is to use transformation, or decomposition of the signal on a set of basis functions prior to processing in the transform domain. Transform theory has played a key role in image processing for a number of years, and it continues to be a topic of interest in theoretical as well as applied work in this field. Image transforms are used widely in many image processing fields, including image enhancement, restoration, encoding, and description [12].

Historically, the Fourier transform has dominated linear time-invariant signal processing. The associated basis functions are complex sinusoidal waves $e^{i\omega t}$ that correspond to the eigenvectors of a linear time-invariant operator. A signal $f(t)$ defined in the temporal domain and its Fourier transform $\hat{f}(\omega)$, defined in the frequency domain, have the following relationships [12, 13]:

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt, \quad (6.1)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega)e^{i\omega t} d\omega. \quad (6.2)$$

Fourier transform characterizes a signal $f(t)$ via its frequency components. Since the support of the bases function $e^{i\omega t}$ covers the whole temporal domain (i.e. infinite support), $\hat{f}(\omega)$ depends on the values of $f(t)$ for all times. This makes the Fourier transform a global transform that cannot analyze local or transient properties of the original signal $f(t)$.

In order to capture frequency evolution of a nonstatic signal, the basis functions should have compact support in both time and frequency domains. To achieve this goal, a windowed Fourier transform (WFT) was first introduced

with the use of a window function $w(t)$ into the Fourier transform [14]:

$$Sf(\omega, t) = \int_{-\infty}^{+\infty} f(\tau)w(t - \tau)e^{-i\omega t}d\tau. \quad (6.3)$$

The energy of the basis function $g_{\tau, \xi}(t) = w(t - \tau)e^{-i\xi t}$ is concentrated in the neighborhood of time τ over an interval of size σ_t , measured by the standard deviation of $|g|^2$. Its Fourier transform is $\hat{g}_{\tau, \xi}(\omega) = \hat{w}(\omega - \xi)e^{-i\tau(\omega - \xi)}$, with energy in frequency domain localized around ξ , over an interval of size σ_ω . In a time–frequency plane (t, ω) , the energy spread of what is called the atom $g_{\tau, \xi}(t)$ is represented by the Heisenberg rectangle with time width σ_t and frequency width σ_ω . The uncertainty principle states that the energy spread of a function and its Fourier transform cannot be simultaneously arbitrarily small, verifying:

$$\sigma_t \sigma_\omega \geq \frac{1}{2}. \quad (6.4)$$

The shape and size of Heisenberg rectangles of a WFT determine the spatial and frequency resolution offered by such transform.

Examples of spatial-frequency tiling with Heisenberg rectangles are shown in Fig. 6.1. Notice that for a windowed Fourier transform, the shapes of the time–frequency boxes are identical across the whole time–frequency plane, which means that the analysis resolution of a windowed Fourier transform remains the same across all frequency and spatial locations.

To analyze transient signal structures of various supports and amplitudes in time, it is necessary to use time–frequency atoms with different support sizes for different temporal locations. For example, in the case of high-frequency structures, which vary rapidly in time, we need higher temporal resolution to accurately trace the trajectory of the changes; on the other hand, for lower frequency, we will need a relatively higher absolute frequency resolution to give a better measurement of the value of frequency. We will show in the next section that wavelet transform provides a natural representation which satisfies these requirements, as illustrated in Fig. 6.1(d).

6.2.1 Continuous Wavelet Transform

A wavelet function is defined as a function $\psi \in L^2(\mathbb{R})$ with a zero average [3, 14]:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0. \quad (6.5)$$

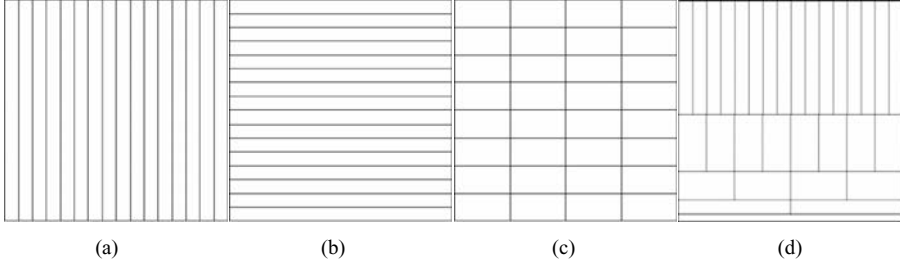


Figure 6.1: Example of spatial-frequency tiling of various transformations. x -axis: spatial resolution and y -axis: frequency resolution. (a) Discrete sampling (no frequency localization), (b) Fourier transform (no temporal localization). (c) windowed Fourier transform (constant Heisenberg boxes), and (d) wavelet transform (variable Heisenberg boxes).

It is normalized $\|\psi\| = 1$, and centered in the neighborhood of $t = 0$. A family of time–frequency atoms is obtained by scaling ψ by s and translating it by u :

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right). \quad (6.6)$$

A continuous wavelet transform decomposes a signal over dilated and translated wavelet functions. The wavelet transform of a signal $f \in L^2(\mathbb{R})$ at time u and scale s is performed as:

$$Wf(u, s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt = 0. \quad (6.7)$$

Assuming that the energy of $\hat{\psi}(\omega)$ is concentrated in a positive frequency interval centered at η , the time–frequency support of a wavelet atom $\psi_{u,s}(t)$ is symbolically represented by a Heisenberg rectangle centered at $(u, \eta/s)$, with time and frequency supports spread proportional to s and $1/s$ respectively. When s varies, the height and width of the rectangle change but its area remains constant, as illustrated by Fig. 6.1 (d).

For the purpose of multiscale analysis, it is often convenient to introduce the scaling function ϕ , which is an aggregation of wavelet functions at scales larger than 1. The scaling function ϕ and the wavelet function ψ are related through the following relations:

$$|\hat{\phi}(\omega)|^2 = \int_1^{+\infty} |\hat{\psi}(s\omega)|^2 \frac{ds}{s}. \quad (6.8)$$

The low-frequency approximation of a signal f at the scale s is computed as:

$$Lf(u, s) = \langle f(t), \phi_s(t - u) \rangle \quad (6.9)$$

with

$$\phi_s(t) = \frac{1}{\sqrt{s}} \phi\left(\frac{t}{s}\right). \quad (6.10)$$

For a one-dimensional signal f , the continuous wavelet transform (6.7) is a two-dimensional representation. This indicates the existence of redundancy that can be reduced and even removed by subsampling the scale parameter s and translation parameter u .

An orthogonal (nonredundant) wavelet transform can be constructed constraining the dilation parameter to be discretized on an exponential sampling with fixed dilation steps and the translation parameter by integer multiples of a dilation-dependent step [15]. In practice, it is convenient to follow a dyadic scale sampling where $s = 2^i$ and $u = 2^i \cdot k$, with i and k being integers. With dyadic dilation and scaling, the wavelet basis function, defined as:

$$\left\{ \psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - 2^j n}{2^j}\right) \right\}_{(j,n) \in \mathbb{Z}^2},$$

forms an orthogonal basis of $L^2(\mathbb{R})$.

For practical purpose, when using orthogonal basis functions, the wavelet transform defined in Eq. (6.7) is only computed for a finite number of scales (2^J) with $\{J = 0, \dots, N\}$, and a low-frequency component $Lf(u, 2^J)$ (often referred to as the DC component) is added to the set of projection coefficients corresponding to scales larger than 2^J for a complete signal representation.

In medical image processing applications, we usually deal with discrete data. We will therefore focus the rest of our discussion on discrete wavelet transform rather than continuous ones.

6.2.2 Discrete Wavelet Transform and Filter Bank

Given a 1D signal of length N , $\{f(n), n = 0, \dots, N - 1\}$, the discrete orthogonal wavelet transform can be organized as a sequence of discrete functions according to the scale parameter $s = 2^j$:

$$\{L_J f, \{W_j f\}_{j \in [I, J]}\}, \quad (6.11)$$

where $L_J f = Lf(2^J n, 2^J)$ and $W_j f = Wf(2^j n, 2^j)$.

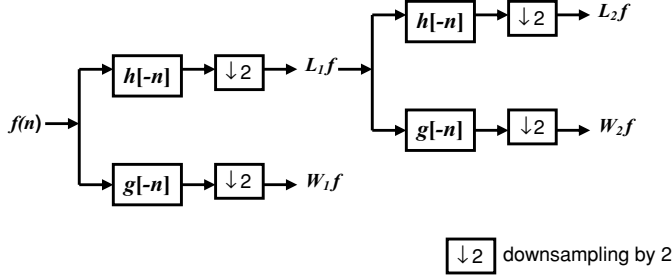


Figure 6.2: Illustration of orthogonal wavelet transform of a discrete signal $f(n)$ with CMF. A two-level expansion is shown.

Wavelet coefficients $W_j f$ at scale $s = 2^j$ have a length of $N/2^j$ and the largest decomposition depth J is bounded by the signal length N as $(\sup(J) = \log_2 N)$.

For fast implementation (such as filter bank algorithms), a pair of conjugate mirror filters (CMF) h and g can be constructed from the scaling function ϕ and wavelet function ψ as follows:

$$h[n] = \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle \quad \text{and} \quad g[n] = \left\langle \frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle. \quad (6.12)$$

A conjugate mirror filter k satisfies the following relation:

$$\left| \hat{k}(\omega) \right|^2 + \left| \hat{k}(\omega + \pi) \right|^2 = 2 \quad \text{and} \quad \hat{k}(0) = 2. \quad (6.13)$$

It can be proven that h is a low-pass filter and g is a high-pass filter. The discrete orthogonal wavelet decomposition in Eq. (6.11) can be computed by applying these two filters to the input signal and recursively decomposing the low-pass band, as illustrated in Fig. 6.2. A detailed proof can be found in [15].

For orthogonal basis, the input signal can be reconstructed from wavelet coefficients computed in Eq. (6.11) using the same pair of filters, as illustrated in Fig. 6.3.

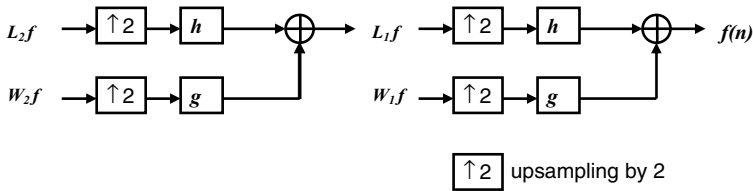


Figure 6.3: Illustration of inverse wavelet transform implemented with CMF. A two-level expansion is shown.

It is easy to prove that the total amount of data after a discrete wavelet expansion as shown in Fig. 6.2 has the same length to the input signal. Therefore, such transform provides a compact representation of the signal suited to data compression as wavelet transform provides a better spatial-frequency localization. On the other hand, since the data was downsampled at each level of expansion, such transform performs poorly on localization or detection problems. Mathematically, the transform is variant under translation of the signal (i.e. is dependent on the downsampling scheme used during the decomposition), which makes it less attractive for analysis of nonstationary signals. In image analysis, translation invariance is critical to the preservation of all the information of the signal and a redundant representation needs to be applied.

In the dyadic wavelet transform framework proposed by Mallat *et al.* [16], sampling of the translation parameter was performed with the same sampling period as that of the input signal in order to preserve translation invariance.

A more general framework of wavelet transform can be designed with different reconstruction and decomposition filters that form a biorthogonal basis. Such generalization provides more flexibility in the design of the wavelet functions. In that case, similar to Eq. (6.11), the discrete dyadic wavelet transform of a signal $s(n)$ is defined as a sequence of discrete functions:

$$\{S_M s(n), \{W_m s(n)\}_{m \in [I, M]}\}_{n \in \mathbb{Z}}, \quad (6.14)$$

where $S_M s(n) = s^* \phi_M(n)$ represents the DC component, or the coarsest information from the input signal.

Given a pair of wavelet function $\psi(x)$ and reconstruction function $\chi(x)$, the discrete dyadic wavelet transform (decomposition and reconstruction) can be implemented with a fast filter bank scheme using a pair of decomposition filters H , G and a reconstruction filter K [16]:

$$\begin{aligned} \hat{\phi}(2\omega) &= e^{-i\omega s} H(\omega) \hat{\phi}(\omega), \\ \hat{\psi}(2\omega) &= e^{-i\omega s} G(\omega) \hat{\psi}(\omega), \\ \hat{\chi}(2\omega) &= e^{i\omega s} K(\omega) \hat{\chi}(\omega), \end{aligned} \quad (6.15)$$

where s is a $\psi(x)$ -dependent sampling shift. The three filters satisfy:

$$|H(\omega)|^2 + G(\omega)K(\omega) = 1. \quad (6.16)$$

Defining $F_s(\omega) = e^{-i\omega s} F(\omega)$, where F is H , G , or K , we can construct a filter bank implementation of the discrete dyadic wavelet transform as illustrated in

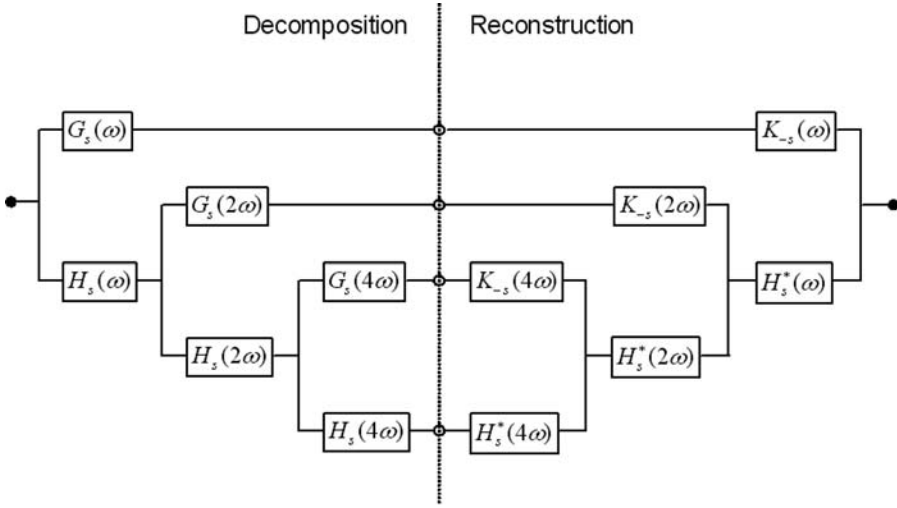


Figure 6.4: Filter bank implementation of a one-dimensional discrete dyadic wavelet transform decomposition and reconstruction for three levels of analysis. $H_s^*(\omega)$ denotes the complex conjugate of $H_s(\omega)$.

Fig. 6.4. Filters $F(2^m \omega)$ defined at level $m + 1$ (i.e., filters applied at wavelet scale 2^m) are constructed by inserting $2^m - 1$ zeros between subsequent filter coefficients from level 1 ($F(\omega)$). Noninteger shifts at level 1 are rounded to the nearest integer. This implementation design is called “*algorithme à trous*” [17, 18] and has a complexity that increases linearly with the number of analysis levels.

In image processing applications, we often deal with two, three, or even higher dimensional data. Extension of the framework to higher dimension is quite straightforward. Multidimensional wavelet bases can be constructed with tensor products of separable basis functions defined along each dimension. In that context, an N -dimensional discrete dyadic wavelet transform with M analysis levels is represented as a set of wavelet coefficients:

$$\{S_M s, \{W_m^1 s, W_m^2 s, \dots, W_m^N s\}_{m=[1, M]}\}, \quad (6.17)$$

where $W_m^k s = \langle s, \psi_m^k \rangle$ represents the detailed information along the k th coordinate at scale m . The wavelet basis is dilated and translated from a set of separable wavelet functions ψ^k , $k = 1, \dots, N$, for example in 3D:

$$\psi_{m, n_1, n_2, n_3}^k(x, y, z) = \frac{1}{2^{3m/2}} \psi^k\left(\frac{x - n_1}{2^m}, \frac{y - n_2}{2^m}, \frac{z - n_3}{2^m}\right), \quad k = 1, 2, 3. \quad (6.18)$$

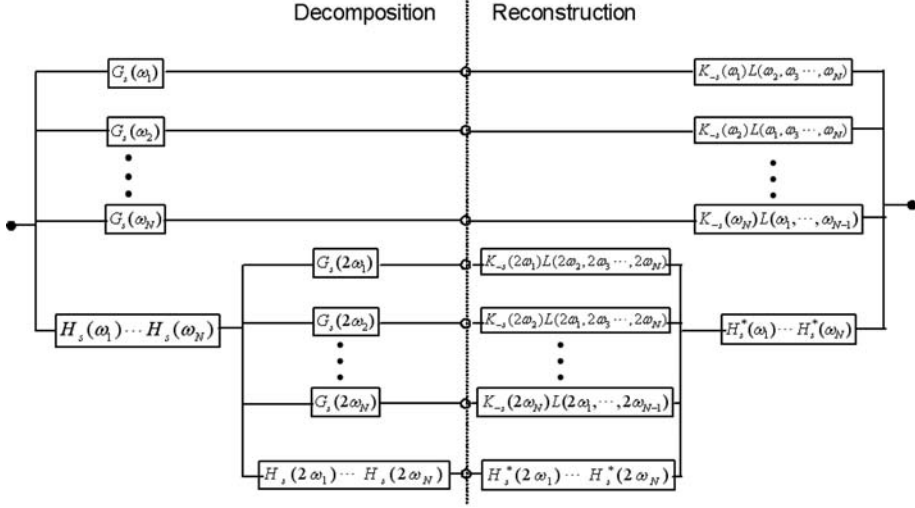


Figure 6.5: Filter bank implementation of a multidimensional discrete dyadic wavelet transform decomposition (left) and reconstruction (right) for two levels of analysis.

In this framework, reconstruction with an N -dimensional dyadic wavelet transform requires a nonseparable filter L_N to compensate the interdimension correlations. This is formulated in a general context as:

$$\sum_{l=1}^N K(\omega_l)G(\omega_l)L_N(\omega, \dots, \omega_{l-1}, \omega_{l+1}, \dots, \omega_N) + \prod_{l=1}^N |H(\omega_l)|^2 = 1. \quad (6.19)$$

Figure 6.5 illustrates a filter bank implementation with a multidimensional discrete dyadic wavelet transform. For more details and discussions we refer to [19].

6.2.3 Other Multiscale Representations

Wavelet transforms are part of a general framework of multiscale analysis. Various multiscale representations have been derived from the spatial-frequency framework offered by wavelet expansion, many of which were introduced to provide more flexibility for the spatial-frequency selectivity or better adaptation to real-world applications.

In this section, we briefly review several multiscale representations derived from wavelet transforms. Readers with an intention to investigate more

theoretical and technical details are referred to the textbooks on Gabor analysis [20], wavelet packets [21], and the original paper on brushlet [22].

6.2.3.1 Gabor Transform and Gabor Wavelets

In his early work, Gabor [23] suggested an expansion of a signal $s(t)$ in terms of time–frequency atoms $g_{m,n}(t)$ defined as:

$$s(t) = \sum_{m,n} c_{m,n} g_{m,n}(t), \quad (6.20)$$

where $g_{m,n}(t)$, $m, n \in Z$, are constructed with a window function $g(x)$, combined to a complex exponential:

$$g_{m,n}(t) = g(t - na)e^{i2\pi mbt}. \quad (6.21)$$

Gabor also suggested that an appropriate choice for the window function $g(x)$ is the Gaussian function due to the fact that a Gaussian function has the theoretically best joint spatial-frequency resolution (uncertainty principle). It is important to note here that the Gabor elementary functions $g_{m,n}(t)$ are not orthogonal and therefore require a biorthogonal dual function $\gamma(x)$ for reconstruction [24]. This dual window function is used for the computation of the expansion coefficients $c_{m,n}$ as:

$$c_{m,n} = \int f(x) \bar{\gamma}(x - na) e^{-i2\pi mbx} dx, \quad (6.22)$$

while the Gaussian window is used for the reconstruction.

The biorthogonality of the two window functions $\gamma(x)$ and $g(x)$ is expressed as:

$$\int g(x) \bar{\gamma}(x - na) e^{-i2\pi mbx} dx = \delta_m \delta_n. \quad (6.23)$$

From Eq. (6.21), it is easy to see that all spatial-frequency atom $g_{m,n}(t)$ share the same spatial-frequency resolution defined by the Gaussian function $g(x)$. As pointed out in the discussion on short-time Fourier transforms, such design is suboptimal for the analysis of signals with different frequency components.

A wavelet-type generalization of Gabor expansion can be constructed such that different window functions are used instead of a single one [25] according to their spatial-frequency location. Following the design of wavelets, a Gabor

wavelet $\psi(x) = g(t)e^{i\eta t}$ is then obtained with a Gaussian function

$$g(t) = \frac{1}{(\sigma^2\pi)^{1/4}} e^{-\frac{t^2}{2\sigma^2}}$$

(see [14]).

Extension of Gabor wavelet to 2D is expressed as:

$$\psi^k(s, y) = g(x, y)e^{-i\eta(x \cos \alpha_k + y \sin \alpha_k)}. \quad (6.24)$$

Different translation and scaling parameters of $\psi^k(x, y)$ constitute the wavelet basis for expansion. An extra parameter α_k provides selectivity for the orientation of the function. We observe here that the 2D Gabor wavelet has a non-separable structure that provides more flexibility on orientation selection than separable wavelet functions.

It is well known that optical sensitive cells in animal's visual cortex respond selectively to stimuli with particular frequency and orientation [26]. Equation (6.24) described a wavelet representation that naturally reflects this neurophysiological phenomenon. Gabor expansion and Gabor wavelets have therefore been widely used for visual discrimination tasks and especially texture recognition [27, 28].

6.2.3.2 Wavelet Packets

Unlike dyadic wavelet transform, wavelet packets decompose the low-frequency component as well as the high-frequency component in every subbands [29]. Such adaptive expansion can be represented with binary trees where each subband high- or low-frequency component is a node with two children corresponding to the pair of high- and low-frequency expansion at the next scale. An admissible tree for an adaptive expansion is therefore defined as a binary tree where each node has either 0 or 2 children, as illustrated in Fig. 6.6(c). The number of all different wavelet packet orthogonal basis (also called a wavelet packets dictionary) is equal to the number of different admissible binary trees, which is of the order of 2^{2^J} , where J is the depth of decomposition [14].

Obviously, wavelet packets provide more flexibility on partitioning the spatial-frequency domain, and therefore improve the separation of noise and signal into different subbands in an approximated sense (this is referred to the near-diagonalization of signal and noise). This property can greatly facilitate

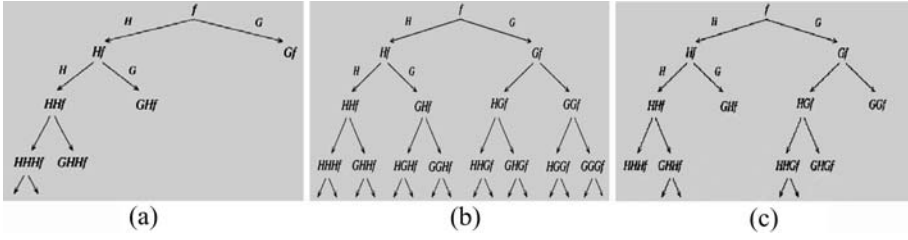


Figure 6.6: (a) Dyadic wavelet decomposition tree. (b) Wavelet packets decomposition tree. (c) An example of an orthogonal basis tree with wavelet packets decomposition.

the enhancement and denoising task of a noisy signal if the wavelet packets basis are selected properly [30]. In practical applications for various medical imaging modalities and applications, features of interest and noise properties have significantly different characteristics that can be efficiently characterized separately with this framework.

A fast algorithm for wavelet-packets best basis selection was introduced by Coifman and Wickerhauser in [30]. This algorithm identifies the “best” basis for a specific problem inside the wavelet packets dictionary according to a criterion (referred to as a cost function) that is minimized. This cost function typically reflects the entropy of the coefficients or the energy of the coefficients inside each subband and the optimal choice minimizes the cost function comparing values at a node and its children. The complexity of the algorithm is $O(N \log N)$ for a signal of N samples.

6.2.3.3 Brushlets

Brushlet functions were introduced to build an orthogonal basis of transient functions with good time–frequency localization. For this purpose, lapped orthogonal transforms with windowed complex exponential functions, such as Gabor functions, have been used for many years in the context of sine–cosine transforms [31].

Brushlet functions are defined with true complex exponential functions on subintervals of the real axis as:

$$u_{j,n}(x) = b_n(x - c_n)e_{j,n}(x) + v(x - a_n)e_{j,n}(2a_n - x) - v(x - a_{n+1})e_{j,n}(2a_{n+1} - x), \quad (6.25)$$

where $l_n = a_{n+1} - a_n$ and $c_n = l_n/2$. The two window functions b_n and v are derived from the ramp function r :

$$r(t) = \begin{cases} 0 & \text{if } t \leq -1 \\ 1 & \text{if } t \geq 1 \end{cases} \quad (6.26)$$

and

$$r^2(t) + r^2(-t) = 1, \quad \forall t \in \mathbb{R}. \quad (6.27)$$

The bump function v is defined as:

$$v(t) = r\left(\frac{t}{\varepsilon}\right) r\left(\frac{-t}{\varepsilon}\right), \quad t \in [\varepsilon, \varepsilon]. \quad (6.28)$$

The bell function b_n is defined by:

$$b_n(t) = \begin{cases} r^2\left(\frac{t + l_n/2}{\varepsilon}\right) & \text{if } t \in [-l_n/2 - \varepsilon, -l_n/2 + \varepsilon] \\ 1 & \text{if } t \in [-l_n/2 + \varepsilon, l_n/2 - \varepsilon] \\ r^2\left(\frac{l_n/2 - t}{\varepsilon}\right) & \text{if } t \in [l_n/2 - \varepsilon, l_n/2 + \varepsilon] \end{cases} \quad (6.29)$$

An illustration of the windowing functions is provided in Fig. 6.7.

Finally, the complex-valued exponentials $e_{j,n}$ are defined as:

$$e_{j,n}(x) = \frac{1}{\sqrt{l_n}} e^{-2i\pi j \frac{(x-a_n)}{l_n}}. \quad (6.30)$$

In order to decompose a given signal f along directional texture components, the Fourier transform $\hat{f}$ of the signal and not the signal itself is projected on the

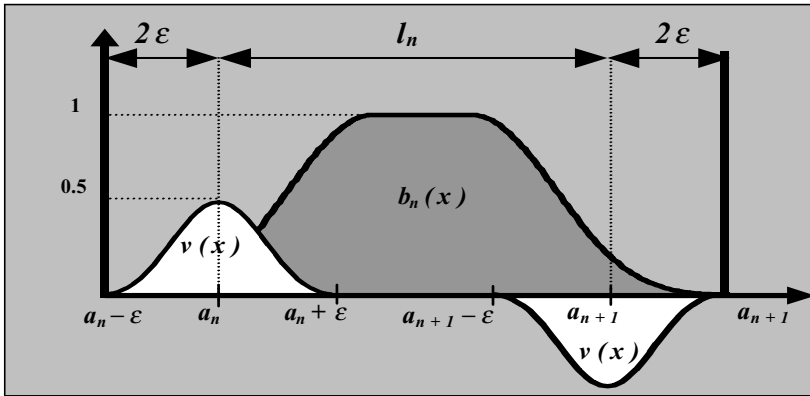


Figure 6.7: Windowing functions b_n and bump functions v defined on the interval $[a_n - \varepsilon, a_{n+1} + \varepsilon]$.

brushlet basis functions:

$$\hat{f} = \sum_n \sum_j \hat{f}_{n,j} u_{n,j}, \quad (6.31)$$

with $u_{n,j}$ being the brushlet basis functions and $\hat{f}_{n,j}$ being the brushlet coefficients.

The original signal f can then be reconstructed by:

$$f = \sum_n \sum_j \hat{f}_{n,j} w_{n,j}, \quad (6.32)$$

where $w_{n,j}$ is the inverse Fourier transform of $u_{n,j}$, which is expressed as:

$$w_{n,j}(x) = \sqrt{l_n} e^{2i\pi a_n x} e^{i\pi l_n x} \left\{ (-1)^j \hat{b}_n(l_n x - j) - 2i \sin(\pi l_n x) \hat{v}(l_n x + j) \right\}, \quad (6.33)$$

with $\hat{b}_n$ and $\hat{v}$ being the Fourier transforms of the window functions b_n and v . Since the Fourier operator is a unitary operator, the family of functions $w_{n,j}$ is also an orthogonal basis of the real axis. We observe here the wavelet-like structure of the $w_{n,j}$ functions with scaling factor l_n and translation factor j . An illustration of the brushlet analysis and synthesis functions is provided in Fig. 6.8.

Projection on the analysis functions $u_{n,j}$ can be implemented efficiently by a folding operator and Fourier transform. The folding technique was introduced by Malvar [31] and is described for multidimensional implementation by Wickerhauser in [21]. These brushlet functions share many common properties with Gabor wavelets and wavelet packets regarding the orientation and frequency selection of the analysis but only brushlet can offer an orthogonal framework

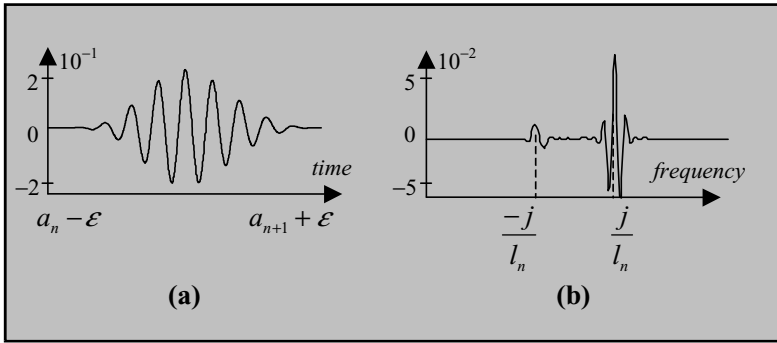


Figure 6.8: (a) Real part of analysis brushlet function $u_{n,j}$. (b) Real part of synthesis brushlet function $w_{n,j}$.

with a single expansion coefficient for a particular pair of frequency and orientation.

6.3 Noise Reduction and Image Enhancement Using Wavelet Transforms

Denoising can be viewed as an estimation problem trying to recover a true signal component X from an observation Y where the signal component has been degraded by a noise component N :

$$Y = X + N. \quad (6.34)$$

The estimation is computed with a thresholding estimator in an orthonormal basis $B = \{g_m\}_{0 \leq m < N}$ as [32]:

$$\hat{X} = \sum_{m=0}^{N-1} \rho_m(\langle X, g_m \rangle) g_m, \quad (6.35)$$

where ρ_m is a thresholding function that aims at eliminating noise components (via attenuating or decreasing some coefficient sets) in the transform domain while preserving the true signal coefficients. If the function ρ_m is modified to rather preserve or increase coefficient values in the transform domain, it is possible to enhance some features of interest in the true signal component with the framework of Eq. (6.35).

Figure 6.9 illustrates a multiscale enhancement and denoising framework using wavelet transforms. An overcomplete dyadic wavelet transform using biorthogonal basis is used. Notice that since the DC cap contains the overall energy distribution, it is usually not thresholded during the procedure. As shown in this figure, thresholding and enhancement functions can be implemented independently from the wavelet filters and easily incorporated into the filter bank framework.

6.3.1 Thresholding Operators for Denoising

As a general rule, wavelet coefficients with larger magnitude are correlated with salient features in the image data. In that context, denoising can be achieved by applying a thresholding operator to the wavelet coefficients (in the transform

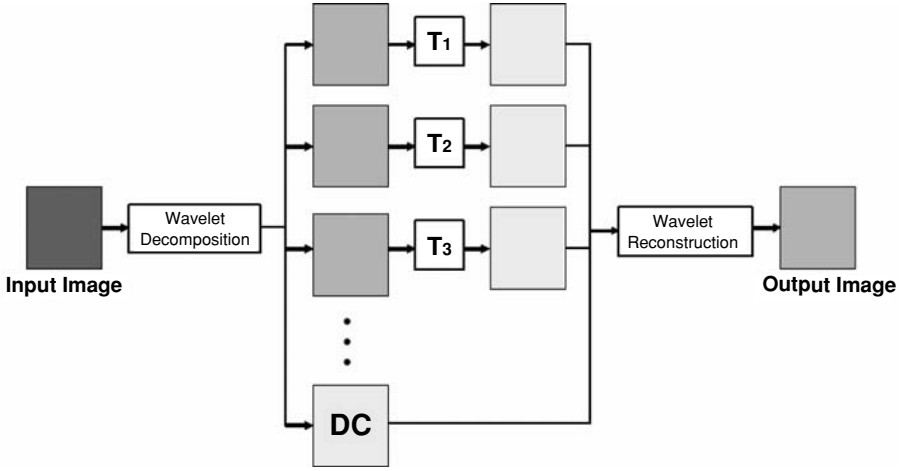


Figure 6.9: A Multiscale framework of denoising and enhancement using discrete dyadic wavelet transform. A three-level decomposition was shown.

domain) followed by reconstruction of the signal to the original image (spatial) domain.

Typical threshold operators for denoising include *hard thresholding*:

$$\rho_T(x) = \begin{cases} x, & \text{if } |x| > T \\ 0, & \text{if } |x| \leq T \end{cases}, \quad (6.36)$$

soft thresholding (wavelet shrinkage) [33]:

$$\rho_T(x) = \begin{cases} x - T, & \text{if } x \geq T \\ x + T, & \text{if } x \leq -T, \\ 0, & \text{if } |x| < T \end{cases}, \quad (6.37)$$

and *affine (firm) thresholding* [34]:

$$\rho_T(x) = \begin{cases} x, & \text{if } |x| \geq T \\ 2x + T, & \text{if } -T \leq x \leq -T/2 \\ 2x - T, & \text{if } T/2 \leq x \leq T \\ 0, & \text{if } |x| < T \end{cases}. \quad (6.38)$$

The shapes of these thresholding operators are illustrated in Fig. 6.10.

6.3.2 Enhancement Operators

Magnitude of wavelet coefficients measures the correlation between the image data and the wavelet functions. For first-derivative-based wavelet, the magnitude

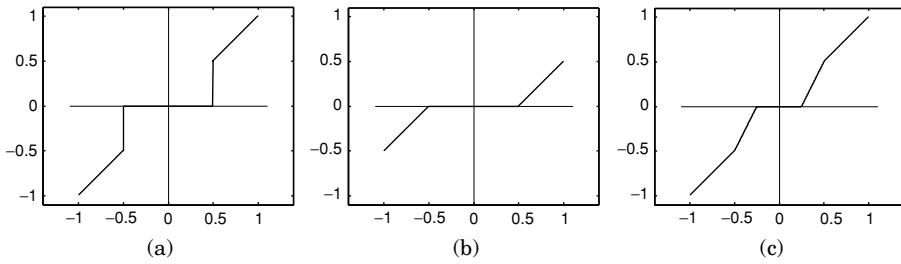


Figure 6.10: Example of thresholding functions, assuming that the input data was normalized to the range of $[-1, 1]$. (a) Hard thresholding, (b) soft thresholding, and (c) affine thresholding. The threshold level was set to $T = 0.5$.

therefore reflects the “strength” of signal variation. For second-derivative-based wavelets, the magnitude is related to the local contrast around a signal variation. In both cases, large wavelet coefficient magnitude occurs around strong edges. To enhance weak edges or subtle objects buried in the background, an enhancement function should be designed such that wavelet coefficients within certain magnitude range are amplified.

General guidelines for designing a nonlinear enhancement function $E(x)$ are [35]:

1. An area of low contrast should be enhanced more than an area of high contrast. This is equivalent to saying that smaller values of wavelet coefficients should be assigned larger gains.
2. A sharp edge should not be blurred.

In addition, an enhancement function may be further subjected to the following constraints [36]:

1. Monotonically increasing: Monotonicity ensures the preservation of the relative strength of signal variations and avoids changing location of local extrema or creating new extrema.
2. Antisymmetry: ($E(-x) = -E(x)$): This property preserves the phase polarity for “edge crispening.”

A simple piecewise linear function [37] that satisfies these conditions is plotted in Fig. 6.11(a):

$$E(x) = \begin{cases} x - (K - 1)T, & \text{if } x < -T \\ Kx, & \text{if } |x| \leq T. \\ x + (K - 1)T, & \text{if } x > T \end{cases} \quad (6.39)$$

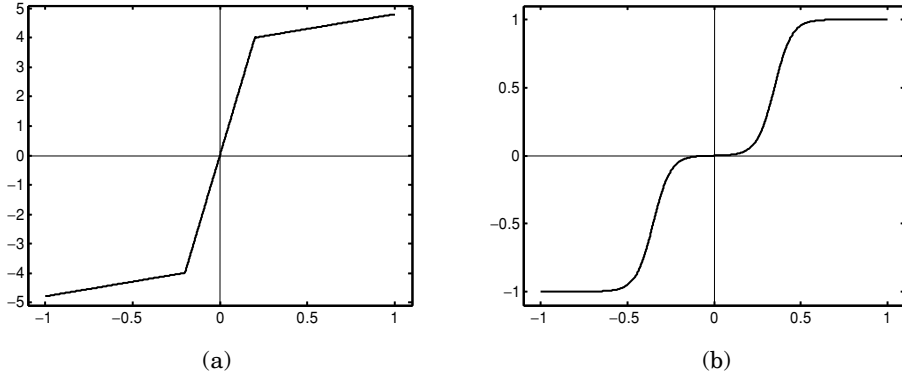


Figure 6.11: Example of enhancement functions, assuming that the input data was normalized to the range of $[-1, 1]$. (a) Piecewise linear function, $T = 0.2$, $K = 20$. (b) Sigmoid enhancement function, $b = 0.35$, $c = 20$. Notice the different scales of the y -axis for the two plots.

Such enhancement is simple to implement, and was used successfully for contrast enhancement on mammograms [19, 38, 39].

From the analysis in the previous subsection, wavelet coefficients with small-magnitude were also related to noise. A simple amplification of small-magnitude coefficients as performed in Eq. (6.39) will certainly also amplify noise components. This enhancement operator is therefore limited to contrast enhancement of data with very low noise level, such as mammograms or CT images. Such a problem can be alleviated by combining the enhancement with a denoising operator presented in the previous subsection [35].

A more careful design can provide more reliable enhancement procedures with a control of noise suppression. For example, a sigmoid function [37], plotted in Fig. 6.11 (b), can be used:

$$E(x) = a[\text{sigm}(c(x - b)) - \text{sigm}(-c(x + b))], \quad (6.40)$$

where

$$a = \frac{1}{\text{sigm}(c(1 - b)) - \text{sigm}(-c(1 + b))}, \quad 0 < b < 1,$$

and $\text{sigm}(y)$ is defined as $\text{sigm}(y) = \frac{1}{1 + e^{-y}}$. The parameters b and c respectively control the threshold and rate of enhancement. It can be easily shown that $E(x)$ in Eq. (6.40) is continuous and monotonically increasing within the interval

$[-1, 1]$. Furthermore, any order of derivatives of $E(x)$ exists and is continuous. This property avoids creating any new discontinuities after enhancement.

6.3.3 Selection of Threshold Value

Given the basic framework of denoising using wavelet thresholding as discussed in the previous sections, it is clear that the threshold level parameter T plays an essential role. Values too small cannot effectively get rid of noise component, while values too large will eliminate useful signal components. There are a variety of ways to determine the threshold value T as will be discussed in this section.

Depending on whether or not the threshold value T changes across wavelet scales and spatial locations, the thresholding can be:

1. *global threshold*: a single value T is to be applied globally to all empirical wavelet coefficients at different scales. $T = \text{const}$.
2. *level-dependent threshold*: a different threshold value T is selected for each wavelet analysis level (scale). $T = T(j)$, $j = 1, \dots, J$, J being the coarsest level for wavelet expansion to be processed.
3. *spatial adaptive threshold*: the threshold value T varies spatially depending on local properties of individual wavelet coefficients. Usually, T is also level dependent. $T = T_j(x, y, z)$.

While a simple way of determining T is as a percentage of coefficients maxima, there are different adaptive ways of assigning the T value according to the noise level (estimated via its variance σ):

1. *universal threshold*: $T = \sigma \sqrt{2 \log n}$ [40], with n equal to the sample size. This threshold was determined in an optimal context for soft thresholding with random Gaussian noise. This scheme is very easy to implement, but typically provides a threshold level larger than with other decision criteria, therefore resulting in smoother reconstructed data. Also such estimation does not take into account the content of the data, but only depends on the data size.
2. *minimax threshold*: $T = \sigma T_n$ [41], where T_n is determined by a minimax rule such that the maximum risk of estimation error across all locations of

the data is minimized. This threshold level depends on the noise and signal relationships in the input data.

3. *stein unbiased estimated of risk*: Similar to minimax threshold but T_n is determined by a different risk rule [42, 43].
4. *spatial adaptive threshold*: $T = \sigma^2/\sigma_X$ [44], where σ_X is the local variance of the observation signal, which can be estimated using a local window moving across the image data or, more accurately, by a context-based clustering algorithm.

In many automatic denoising methods to determine the threshold value T , an estimation of the noise variance σ is needed. Donoho *et al.* [45] proposed a robust estimation of noise level σ based on the median absolute value of the wavelet coefficients as:

$$\sigma = \frac{\text{median}(|W_1(x, y, z)|)}{0.6745}, \quad (6.41)$$

where W_1 is the most detailed level of wavelet coefficients. Such estimator has become very popular in practice and is widely used.

6.3.4 Summary

In general, multiscale denoising techniques involve a transformation process and a thresholding operator in the transform domain. Research dedicated to the improvement of such a technique has been explored along both directions. Various multiscale expansions have been proposed, aimed at better adaptation to signal and feature characteristics. Traditionally, an orthogonal base was used for expansion [33], which leads to a spatial-variant transform. Various artifacts, e.g. pseudo-Gibbs phenomena, were exhibited in the vicinity of discontinuities. Coifman *et al.* [40] proposed a translation-invariant thresholding scheme, which averages several denoising results on different spatial shifts of the input image. Laine *et al.* [38] prompted to an overcomplete representation which allows redundancy in the transform coefficients domain and provides a translation-invariant decomposition. Wavelet coefficients in an overcomplete representation have the same size as the input image, when treated as a subband image. Many denoising and enhancement techniques can be applied within a

multiscale framework for spatial-frequency adaptation and solve certain noise amplification problems. For a better separation of noise and signal components in the transform domain, other multiscale representations have also been widely investigated. Examples of such multiscale representations can be found in Section 6.2.3.

The magnitude of the wavelet coefficients is related to the correlations between the signal and the wavelet basis function, which is the only criterion to determine whether or not noise variation appears. Therefore, the selection of the wavelet basis is a critical step in the design of the denoising and enhancement procedure. Wavelet basis constructed from derivatives of spline functions [46] were shown to have many advantages in denoising and enhancement. Such wavelet functions, either symmetric or antisymmetric, are smooth with compact support. Higher order spline function resembles Gaussian function, therefore providing ideal spatial-frequency resolution for signal analysis. Moreover, moduli of wavelet coefficients using first-derivative spline wavelets are proportional to the magnitude of a gradient vector [47]. Analysis over such modulus therefore provides extra information on directional correlations, and is especially important for three or higher dimensional data analysis. Other wavelet basis functions have also been developed to provide specific adaptation to different type of signals. To name a few, slantlet [48], curvelet [49, 50], and ridgelet [51] were designed to improve the correlations with edge information and were used for edge-preserved denoising, while Fresnelets functions, based on B-spline functions [52], were designed for processing of digital holography.

In a parallel direction, many research works on multiscale denoising focused on improving thresholding operators. In the following discussion, “thresholding operator” is a rather general concept that includes both denoising and enhancement operators as described before. A determination of thresholding method includes both selection of the thresholding operator and a decision or estimation of the threshold parameters (threshold level, enhancement gain, etc.). Some examples of thresholding operators designed to improve the basic thresholding rules as shown in Eqs. (6.36)–(6.38) include the non-negative garrote thresholding [53]:

$$\rho_T^G(x) = \begin{cases} 0, & \text{if } |x| \leq T \\ x - \frac{T^2}{x}, & \text{if } |x| > T \end{cases}, \quad (6.42)$$

and the SCAD thresholding [53, 54]:

$$\rho_T^{\text{SCAD}}(x) = \begin{cases} \text{sign}(x)\max(0, |x| - T), & \text{if } |x| \leq 2T \\ ((\alpha - 1)x - \alpha T \text{sign}(x))/(\alpha - 2), & \text{if } 2T < |x| \leq \alpha T. \\ x, & \text{if } |x| > \alpha T \end{cases} \quad (6.43)$$

On the other hand, cross-validation [55–57] and recursive hypothesis testing procedure [58] were investigated for automatically determining the threshold level T .

6.3.5 State-of-the-Art and Applications

In this section, we review two examples of multiscale denoising. To illustrate the power of multiscale analysis, two extreme cases of medical imaging modalities (ultrasound and PET/SPECT) with high noise level and complicated noise patterns were considered. A more detailed description of these clinical applications can be found in [59, 60].

6.3.5.1 Spatial-Temporal Analysis of Real-Time 3D Cardiac Ultrasound Using Brushlet [59]

Recent development of a real-time three-dimensional (RT3D) ultrasound imaging modality that captures an entire cardiac volume instantaneously with fixed geometric parameters over a complete cardiac cycle raises new issues and challenges for denoising and volume extraction. On one hand, resolution of RT3D is lower than with previous 2D and 3D generations of ultrasound modalities and the level of speckle noise is very high. On the other hand the amount of information recorded per cardiac cycle is much more as this is a true 3D+time modality. Because of the fast acquisition time and the true three-dimensional nature of the transducer, there exists a strong coherence of surfaces in 3D space and time for echocardiograms recorded from moving cardiac tissue that should be exploited for optimal denoising and enhancement.

A simple observation of ultrasound images reveals the absence of true boundaries between the blood cavity and the myocardium muscle tissue. The myocardial wall is rather depicted as a field of bright moving texture and the denoising problem can therefore be approached as a texture characterization task. Approaches for texture classification and denoising can be divided into structural

and statistical methods adapted respectively to macro- and micro-textural elements. Recent work on texture characterization and more specifically denoising of ultrasound data via spatio-temporal analysis include steerable filters and Gabor oriented filters [61, 62]. Both techniques are nonorthogonal and therefore suffer from noncomplete partitioning of the Fourier domain. As we showed in previous section, brushlets allow more flexibility on the partitioning of the Fourier domain and work with an orthogonal basis that provides perfect reconstruction of an original signal. In this application, modifications from the original implementation, which extended the analysis to three and four dimensions and performed the analysis in an overcomplete framework, have been made.

Brushlet basis functions decompose an N -dimensional signal along specific spatial-directions via analysis of its Fourier domain. As they only depend on spatial-frequency content, brushlet decompositions are invariant to the intensity or contrast range in the original data. This makes them very suitable and a powerful basis for the analysis of RT3D ultrasound where choosing a single global-intensity-based edge threshold is not possible due to position-dependent attenuation of the signal. There are as many basis functions as there are subintervals in the Fourier domain defining brushstrokes associated with the center frequency of each interval. The tiling of the Fourier domain therefore determines the resolution and orientation of the brushlet basis functions as illustrated in Fig. 6.12(a).

The resolution of each brushstroke is inversely proportional to the size of the interval, as illustrated in Fig. 6.12(b). The major difference between the brushlet basis and wavelet packets is the possibility of any arbitrary tiling of the time–frequency plane and the perfect localization of a single frequency in one coefficient.

Spatial Denoising via Thresholding. Denoising was performed via thresholding of the brushlet coefficients. In the case of RT3D ultrasound, speckle noise components are concentrated in the high-frequency coefficients without specific direction whereas cardiac structures are decomposed into the low-frequency components along different orientations. Decorrelation of signal and noise in the frequency domain was therefore performed by removing the higher frequency components and thresholding only the lower frequency components prior to reconstruction. Denoising performance was compared for processing in 2D and

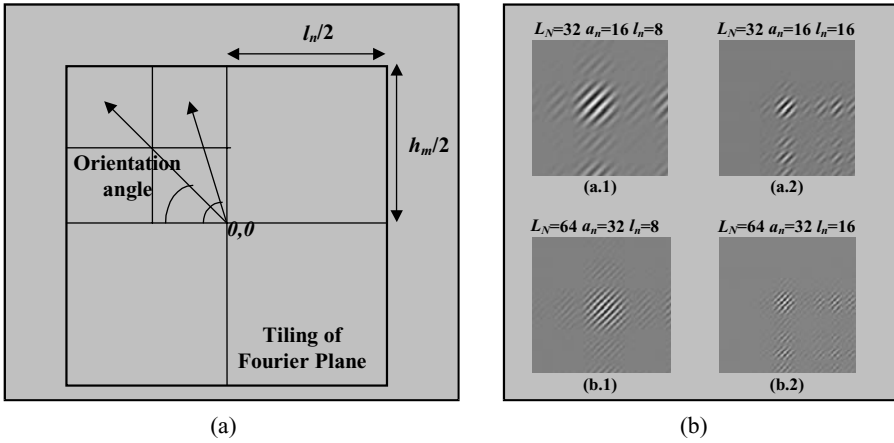


Figure 6.12: (a) Orientation and oscillation frequency of brushlet analysis functions in 2D. The size of each subquadrant in the Fourier plane determines the resolution of the analysis function while the position of the subquadrant center determines the orientation of the analysis function. (b) Illustration of selected brushlet orientation and oscillation frequencies. Fourier plane size L_N , center frequency a_n , and subintervals size l_n are provided for each 2D brushlet basis function.

3D to demonstrate the advantage of extending the brushlet analysis to 3D as illustrated in Fig. 6.13, for a set of six long-axis and six short-axis slices.

Qualitatively, it was observed that the third dimension improved the quality of the denoised data in terms of spatial resolution at the cost of losing some contrast. When compared to 2D denoising, 3D denoising produced smoother features with better-localized contours. Specifically, small local artifacts not persistent in adjacent slices were eliminated and inversely weak contours persistent in adjacent slices were enhanced. This phenomenon can be best appreciated in the short-axis examples where the resolution is the lowest.

Improving Denoising by Including Time: Results on a Mathematical Phantom. To quantitatively evaluate potential denoising performance improvement brought about by including the temporal dimension, initial testing was performed on a mathematical phantom. The phantom, plotted in Fig. 6.14, consisted of an ovoid volume growing in time that schematically mimicked aspects of the left ventricle with an inner gray cavity surrounded by a thick white

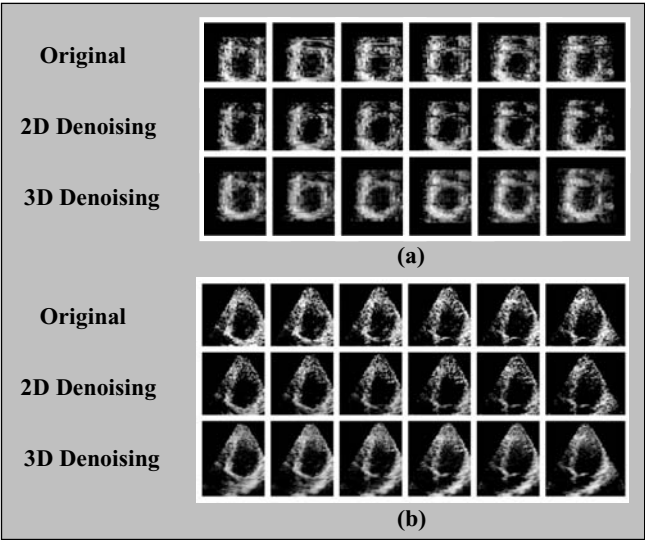


Figure 6.13: 2D versus 3D spatial denoising on RT3D ultrasound data. (a) Series of six consecutive short-axis slices extracted from a clinical data set. (b) Series of six consecutive long-axis slices extracted from the same clinical data set.

wall on a black background. The size of a single volume was $64 \times 64 \times 64$ and there were 16 frames growing in time. The volume increased by 70% over 16 time frames, similar to the average ejection fraction in normal patients.

The phantom was corrupted with two types of noise: (1) multiplicative speckle noise with uniform distribution and (2) multiplicative speckle noise with Rayleigh distribution.

The level of speckle noise was set so that the signal-to-noise ratio (SNR) of the noisy data was equal to -15 dB. Cross-sectional slices through a single volume of the noisy phantoms are displayed in Fig. 6.15.

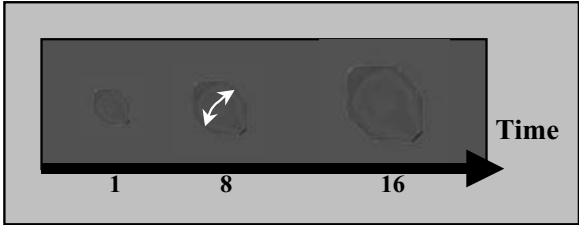


Figure 6.14: Mathematical phantom. Ovoid volume with 16 frames growing in time.

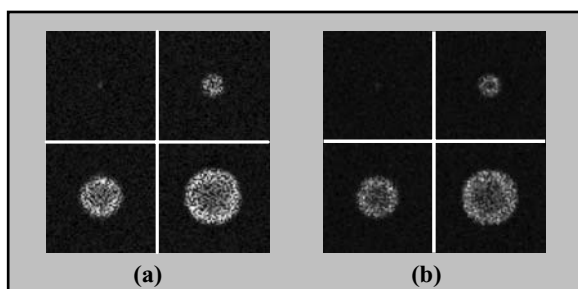


Figure 6.15: Mathematical phantom corrupted with speckle noise. (a) Speckle noise with uniform distribution. (b) Speckle noise with Rayleigh distribution.

Denoising was carried out with both 3D and 4D brushlet analyses. Regular tiling was applied with four subintervals in each dimension. Volumes were reconstructed after resetting the higher frequency coefficients and hard thresholding the lower frequency coefficients at 25% of their maxima. Results for a single slice are provided in Fig. 6.16.

These results revealed that inclusion of the temporal dimension greatly improved the denoising performance. From a qualitative point of view, the contrast

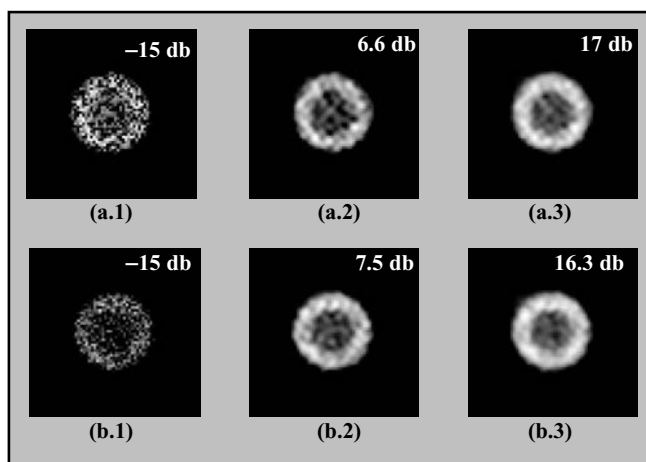


Figure 6.16: Denoising of mathematical phantom with 3D and 4D brushlet analyses. (a) Results for phantom corrupted with uniformly distributed speckle noise. (b) Results for phantom corrupted with Rayleigh distributed speckle noise. ((a.1)–(b.1)) Original slices. ((a.2)–(b.2)) Slices denoised with 3D brushlet expansion. ((a.3)–(b.3)) Slices denoised with 4D brushlet expansion. SNR values are indicated for each slice.

of the denoised slices improved and with a better definition of borders and more homogeneity inside the white and gray areas. Quantitatively, SNR values improved by 50% between 3D and 4D denoising.

A second motivation for performing multidimensional analysis on cardiac clinical data is to take full advantage of the continuity of spatial and temporal frequency content of multidimensional RT3D signals. The high level of speckle noise in ultrasound clinical data sets recorded with the real-time 3D transducer, the nonuniform absorption coefficients of cardiac tissues, and the motion of the heart contribute to the addition of artifacts that can either add echolike signals inside the cavity or suppress echo signals from the myocardium wall. These artifacts complicate the segmentation task by introducing artificial edges inside the cavity or destroying edges at the epicardial and endocardial borders. Since these artifacts are not persistent in time, inclusion of the temporal component in the analysis helps resolve them. To illustrate the aptitude of the brushlet analysis to provide missing contour information, the previous mathematical phantom was modified by removing a part of the white wall in the eighth time frame. Both 3D analysis on the time frame with the defect and 4D brushlet analysis applied to the 16 time frames were computed after corruption with Rayleigh speckle noise. Results are displayed in Fig. 6.17.

Results showed a remarkable correction of the wall defect with the 4D (3D+time) brushlet denoising that could not be obtained with 3D analysis alone. This type of artifact is similar to the dropouts in echo signals that result in loss of myocardium tissue in some frames or the introduction of tissuelike signals inside the cavity. Such artifacts are not persistent in time and could be removed with the inclusion of temporal dimension in the denoising process.

Finally, experiments on clinical data sets, as illustrated in Fig. 6.18, showed the superior performance of spatio-temporal denoising versus simple spatial

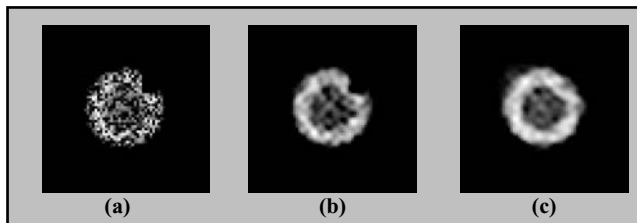


Figure 6.17: (a) Original noisy slice with defect, (b) denoised slice with 3D brushlet analysis, and (c) denoised slice with 4D brushlet analysis.

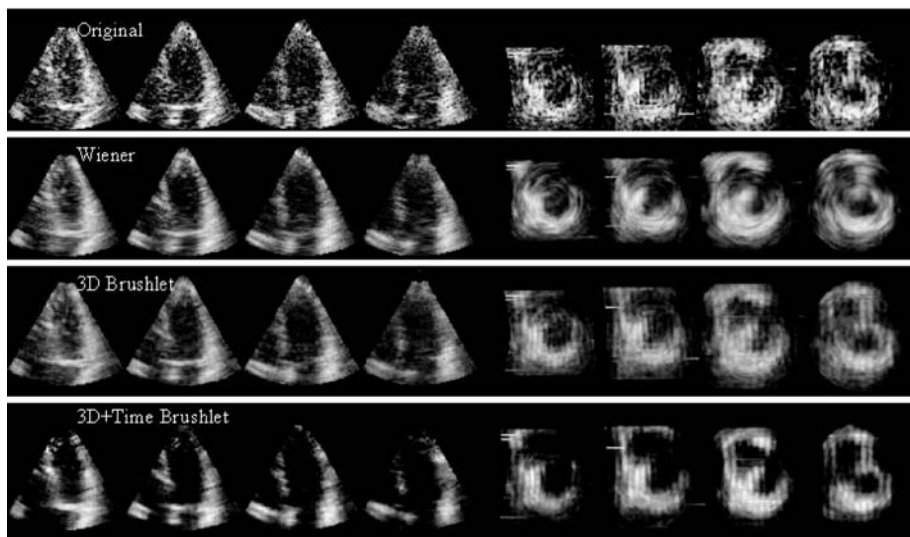


Figure 6.18: Spatio-temporal denoising with brushlet expansion on RT3D ultrasound data illustrated on four long-axis and four short-axis slices.

denoising and Wiener filtering on RT3D ultrasound data. Adding the time dimension leads to images with better contrast and sharper contours while preserving the original textural aspect of the ultrasound data. Wiener filtering provided good results but introduced blurring artifacts that severely altered the quality of the short-axis denoised images. This type of artifact is unacceptable in medical applications where anatomical structure detail needs to be preserved. It was also observed that the epicardium borders were enhanced with sharper contrast when combining brushlet spatial and temporal denoising. Such enhancement is very desirable for quantification of LV mass and wall thickness analysis that requires segmentation of both the myocardial endocardial and epicardial borders.

6.3.5.2 Cross-Scale Regularization for Tomographic Images [60]

Tomographic image modalities such as PET and SPECT rely on an unstable inverse problem of spatial signal reconstruction from sampled line projections. Tomographic reconstruction includes backprojection of the sinogram signal via Radon transform and regularization for removal of noisy artifacts. Because the Radon transform is a smoothing process, backprojection in the presence of additive noise is an ill-posed inverse problem that requires a regularization of the reconstructed noise component, which can become very large. Standard

regularization methods include filtered backprojection (FBP) with nonlinear filtering corrections, expectation-maximization and maximum *a posteriori* estimators [63–66]. The most commonly used tomographic reconstruction method combines a low-pass filter, for noise suppression, and a ramp filter for standard filtered backprojection algorithm. The cut-off frequency of the low-pass filter controls the balance between SNR and spatial resolution. While high-frequency noise is eliminated after low-pass filtering, useful high-frequency information, such as sharp varied signals and edges, is also attenuated. In addition, noise components in low-frequency bands still exist. For these two reasons, tomographic images reconstructed with FBP algorithms often suffer from over-smoothness or/and low SNR. Post-processing including denoising and enhancement is therefore helpful in improving image qualities for reliable clinical interpretation.

As low-pass filtering has always been considered one of the most fundamental denoising techniques, embedding a multiscale denoising module to partially replace the low-pass filtering operator in the FBP algorithm can potentially improve the image quality of reconstruction in terms of both spatial resolution and signal-to-noise ratio. The intuitive approach to combine FBP and denoising is therefore to preserve more high-frequency features during the FBP reconstruction by using a low-pass filter with higher cut-off frequency, or removing the low-pass prefiltering. The noise mixed with the high-frequency signal components is then further processed via a multiscale denoising operator. An illustration of the denoising performance is provided in Fig. 6.19 for simple comparison between traditional FBP using a clinical console (low-pass filter using Hann filter with cut-off frequency set to 0.4) and the proposed two-step processing. It can be observed that the second method, based on FBP using Hann filter with a higher cut-off frequency, generates a reconstructed image containing more detailed information as well as more significant noisy features. After multiscale denoising (combining wavelet packets thresholding and brushlet thresholding), image quality markedly improved, showing more anatomical details and spatial information.

Thresholding on Three-Dimensional Wavelet Modulus. Both PET and SPECT image reconstructed using FBP display strong directional noise patterns. Most feature-based denoising methods, including wavelet thresholding, are based on edge information and are not suited to directional noise components that resemble strong edges. Indeed, edge information alone cannot accurately separate noise from meaningful signal features in a single image. A novel approach to overcome this limitation is to apply the multiscale analysis and

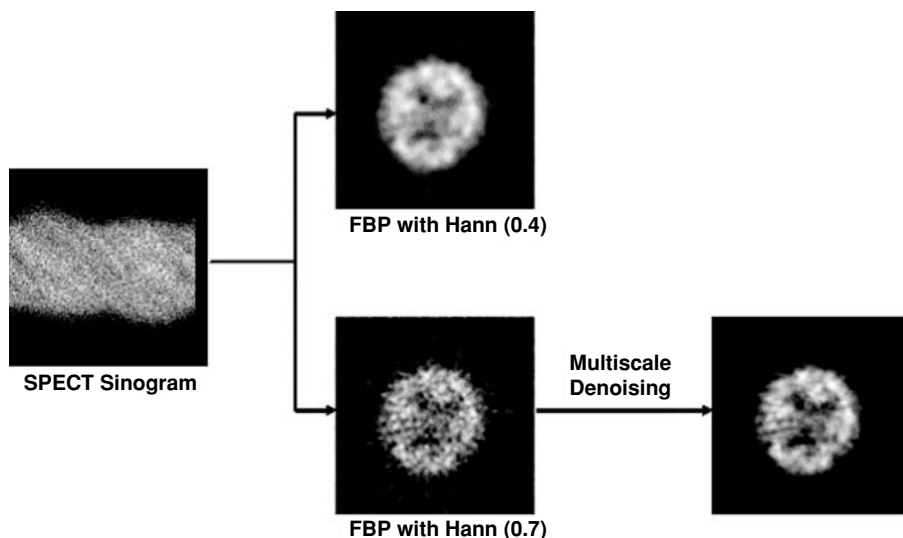


Figure 6.19: Illustration, on a clinical brain SPECT slice, of the combination of multiscale denoising and traditional FBP with higher cut-off frequency to improve tomographic reconstruction.

denoising scheme using three-dimensional wavelet expansion that integrates edge information along continuous boundaries in 3D space. In three dimensions, such integration can accurately separate anatomical surfaces from noisy components that do not exhibit a directional pattern across adjacent tomographic slices. Unlike traditional wavelet denoising techniques, thresholding was performed on the modulus of the wavelet coefficients (“wavelet modulus”). A first derivative of the cubic spline function was used for the wavelet basis function which approximates the first derivatives of a Gaussian function and therefore benefits from the following properties:

1. By the uncertainty principle [14], the Gaussian probability density function is optimally concentrated in both time and frequency domains, and thus is suitable for time–frequency analysis.
2. Derivatives of Gaussian function can be used for rotation-invariant processing [67].
3. The Gaussian function generates a causal (in a sense that a coarse scale depends exclusively on the previous finer scale) scale space. This makes scale-space “tracking” of emergent features possible [68].

Because the wavelet basis ψ^1 , ψ^2 , and ψ^3 are first derivatives of a cubic spline function θ , the three components of a wavelet coefficient $W_m^k s(n_1, n_2, n_3) = \langle s, \psi_{m, n_1, n_2, n_3}^k \rangle$, $k = 1, 2, 3$, are proportional to the coordinates of the gradient vector of the input image s smoothed by a dilated version of θ . From these coordinates, one can compute the angle of the gradient vector, which indicates the direction in which the first derivative of the smoothed s has the largest amplitude (or the direction in which s changes most rapidly). The amplitude of this maximized first derivative is equal to the modulus of the gradient vector, and therefore proportional to the wavelet modulus:

$$M_m s = \sqrt{|W_m^1 s|^2 + |W_m^2 s|^2 + |W_m^3 s|^2}. \quad (6.44)$$

Thresholding this modulus value instead of the coefficient value consists of first selecting a direction in which the partial derivative is maximum at each scale, and then thresholding the amplitude of the partial derivative in this direction. The modified wavelet coefficients are then computed from the thresholded modulus and the angle of the gradient vector. Such paradigm applies an adaptive choice of the spatial orientation in order to best correlate the signal features with the wavelet coefficients. It can therefore provide a more robust and accurate selection of correlated signals compared to traditional orientation selection along three orthogonal Cartesian directions.

Figure 6.20 illustrates the performance of this approach at denoising a clinical brain PET data set reconstructed by FBP with a ramp filter. The reconstructed PET images, illustrated for one slice in Fig. 6.20(a), contain prominent noise in high frequency but do not express strong edge features in the wavelet modulus expansions at scale 1 through 5 as illustrated in Fig. 6.20(b)–(f).

Cross-Scale Regularization for Images with Low SNR. As shown in Fig. 6.20(b), very often in tomographic images, the first level of expansion (level with more detailed information) is overwhelmed by noise in a random pattern. Thresholding operators determined only by the information in this multiscale level can hardly recover useful signal features from the noisy observation. On the other hand, wavelet coefficients in the first level contain the most detailed information in a spatial-frequency expansion, and therefore influence directly the spatial resolution of the reconstructed image.

To have more signal-related coefficients recovered, additional information or *a priori* knowledge is needed. Intuitively, an edge indication map could

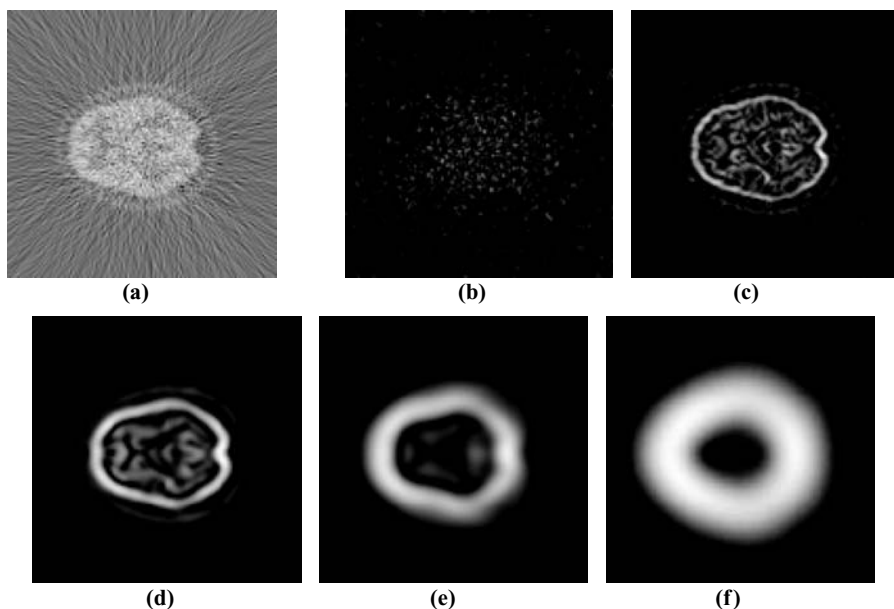


Figure 6.20: (a) A brain PET image from a 3D data set with high level of noise. (b)–(f) Modulus of wavelet coefficients at expansion scale 1 to 5.

beneficially assist such wavelet expansion based on first derivative of spline wavelets. Without seeking external *a priori* information, it was observed that wavelet modulus from the next higher wavelet level can serve as a good edge estimation. An edge indication map with values between 0 and 1 (analogous to the probability that a pixel is located on an edge) was therefore constructed by normalizing the modulus of this subband. A pixel-wise multiplication of the edge indication map and the first level wavelet modulus can identify the location of wavelet coefficients that are more likely to belong to a true anatomical edge and should be preserved, as well as the locations of the wavelet coefficients that are unlikely to be related to real edge signal and that should be attenuated. This approach is referred to as cross-scale regularization. A comparison between traditional wavelet shrinkage and cross-scale regularization for recovering useful signals from the most detailed level of wavelet modulus is provided in Fig. 6.21.

A cross-scale regularization process does not introduce any additional parameter avoiding extra complexity for algorithm optimization and automation. We point out that an improved edge indication prior can be built upon a modified wavelet modulus in the next spatial-frequency scale processed using traditional thresholding and enhancement operator.

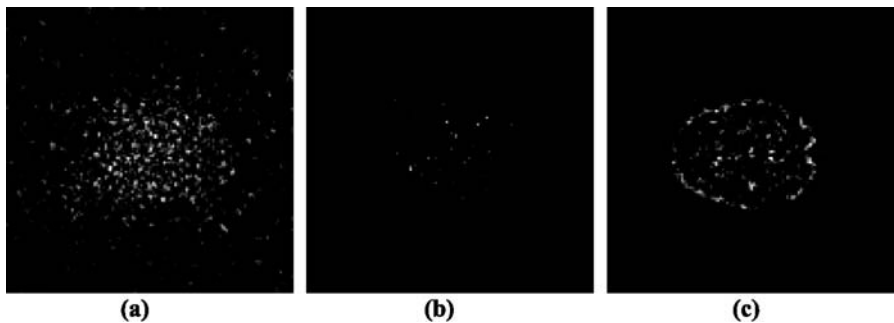


Figure 6.21: (a) Wavelet modulus in first level of a PET brain image as shown in Figs. 6.20 (a) and (b). (b) Thresholding of the wavelet modulus from (a) using a wavelet shrinkage operator. (c) Thresholding of the wavelet modulus from (a) with cross-scale regularization.

Spatial-frequency representations of a signal after wavelet expansion offer the possibility to adaptively process an image data in different sub-bands. Such adaptive scheme can for example combine enhancement of wavelet coefficients in the coarse levels, and resetting of the most detailed levels for noise suppression. We show in Fig. 6.22 how such adaptive processing can remarkably

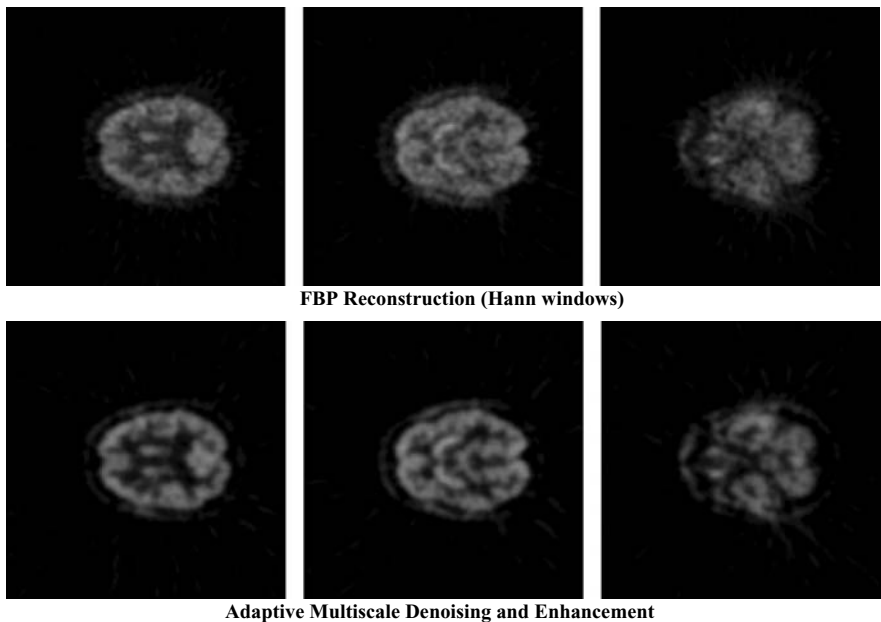


Figure 6.22: Denoising of PET brain data and comparison between unprocessed and multiscale processed images.

improve image quality for PET images that were usually degraded by low resolution and high level of noise.

6.4 Image Segmentation Using Wavelets

6.4.1 Multiscale Texture Classification and Segmentation

Texture is an important characteristic for analyzing many types of images, including natural scenes and medical images. With the unique property of spatial-frequency localization, wavelet functions provide an ideal representation for texture analysis. Experimental evidence on human and mammalian vision support the notion of spatial-frequency analysis that maximizes a simultaneous localization of energy in both spatial and frequency domains [69–71]. These psychophysical and physiological findings lead to several research works on texture-based segmentation methods based on multiscale analysis.

Gabor transform, as suggested by the uncertainty principle, provides an optimal joint resolution in the space-frequency domain. Many early works utilized Gabor transforms for texture characteristics. In [27] an example is given on the use of Gabor coefficient spectral signatures [72] to separate distinct textural regions characterized by different orientations and predominant anisotropic texture moments. Porat *et al.* proposed in [28] six features derived from Gabor coefficients to characterize a local texture component in an image: the dominant localized frequency; the second moment (variance) of the localized frequency; center of gravity; variance of local orientation; local mean intensity; and variance of the intensity level. A simple minimum-distance classifier was used to classify individual textured regions within a single image using these features.

Many wavelet-based texture segmentation methods had been investigated thereafter. Most of these methods follow a three-step procedure: multiscale expansion, feature characterization, and classification. As such, they are usually different from each other in these aspects.

Various multiscale representations have been used for texture analysis. Unser [73] used a redundant wavelet frame. Laine *et al.* [74] investigated a wavelet packets representation and extended their research to a redundant wavelet packets frame with Lemarié–Battle filters in [75]. Modulated wavelets

were used in [76] for better orientation adaptivity. To further extend the flexibility of the spatial-frequency analysis, a multiwavelet packet, combining multiple wavelet basis functions at different expansion levels, was used in [77]. An M -band wavelet expansion, which differs from a dyadic wavelet transform in the fact that each expansion level contains M channels of analysis, was used in [78] to improve orientation selectivity.

Quality and accuracy of segmentation ultimately depend on the selection of the characterizing features. A simple feature selection can use the amplitude of the wavelet coefficients [76]. Many multiscale texture segmentation methods construct the feature vector from various local statistics of the wavelet coefficients, such as its local variance [73, 79], moments [80], or energy signature [74, 78, 81]. Wavelet extrema density, defined as the number of extrema of wavelet coefficients per unit area, was used in [77]. In [75], a 1D envelope detection was first applied to the wavelet packets coefficients according to their orientation, and a feature vector was constructed as the collection of envelope values for each spatial-frequency component. More sophisticated statistical analyses involving Bayesian analysis and Markov random fields (MRF) were also used to estimate local and long-range correlations [82, 83]. Other multiscale textural features were also reported, for example χ^2 test and histogram testing were used in [84], “Roughness” based on fractal dimension measurement was used in [85].

Texture-based segmentation is usually achieved by texture classification. Classic classifiers, such as the minimum distance classifier [28], are easier to implement when the dimension of the feature vector is small and the groups of samples are well segregated. The most popular classification procedures reported in the literature are the K-mean classifier [73, 75, 76, 78, 79, 81, 85] and the neural networks classifiers [27, 74, 80, 82].

As an example, we illustrate in Fig. 6.23 a texture-based segmentation method on a synthetic texture image and a medical image from a brain MRI data set. The algorithm used for this example from [75] uses the combination of *wavelet packets frame* with Lemarié–Battle filters, *multiscale envelope* features, and a *K-mean* classifier.

6.4.2 Wavelet Edge Detection and Segmentation

Edge detection plays an important role in image segmentation. In many cases, boundary delineation is the ultimate goal for an image segmentation and a good

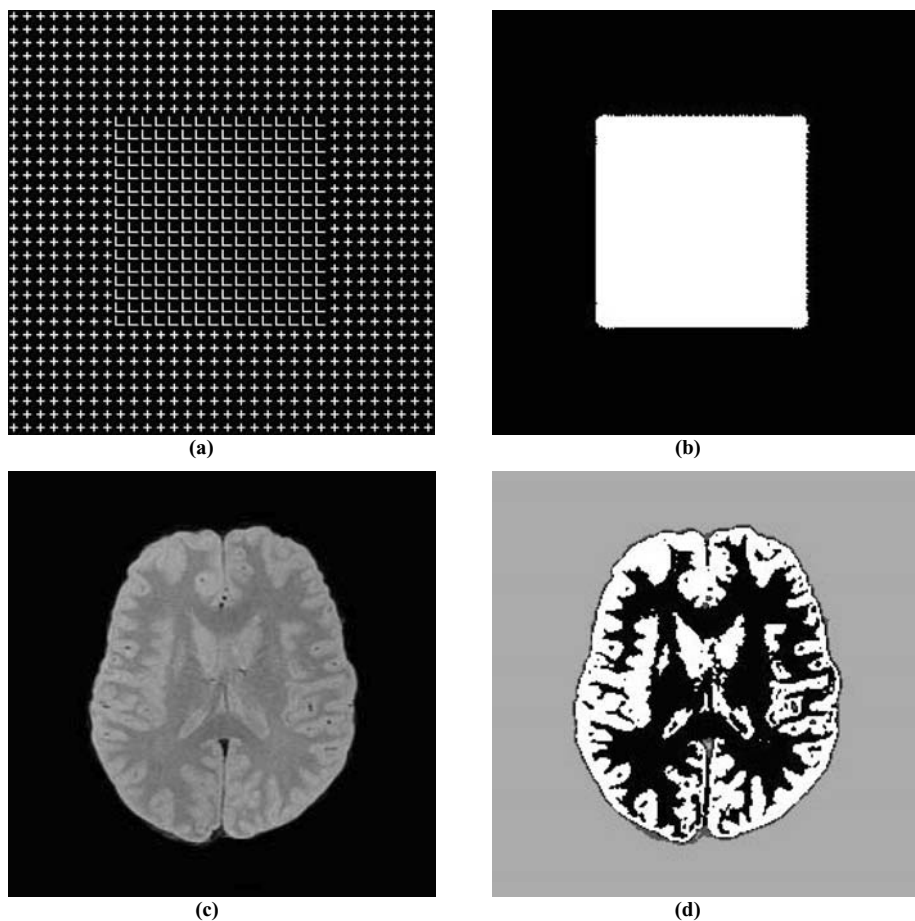


Figure 6.23: Sample results using multiscale texture segmentation. (a) Synthetic texture image. (b) Segmentation result for image (a) with a 2-class labeling. (c) MRI T1 image of a human brain. (d) Segmentation result for image (c) with a 4-class labeling.

edge detector itself can then fulfill the requirement of segmentation. On the other hand, many segmentation techniques require an estimation of object edges for their initialization. For example, with standard gradient-based deformable models, an edge map is used to determine where the deforming interface must stop. In this case, the final result of the segmentation method depends heavily on the accuracy and completeness of the initial edge map. Although many research works have made some efforts to eliminate this type of interdependency by

introducing nonedge constraints [86, 87], it is necessary and equally important to improve the edge estimation process itself.

As pointed out by the pioneering work of Mallat *et al.* [16], first- or second-derivative-based wavelet functions can be used for multiscale edge detection. Most multiscale edge detectors smooth the input signal at various scales and detect sharp variation locations (edges) from their first or second derivatives. Edge locations are related to the extrema of the first derivative of the signal and the zero crossings of the second derivative of the signal. In [16], it was also pointed out that first-derivative wavelet functions are more appropriate for edge detection since the magnitude of wavelet modulus represents the relative “strength” of the edges, and therefore enable to differentiate meaningful edges from small fluctuations caused by noise.

Using the first derivative of a smooth function $\theta(x, y)$ as the mother wavelet of a multiscale expansion results in a representation where the two components of wavelet coefficients at a certain scale s are related to the gradient vector of the input image $f(x, y)$ smoothed by a dilated version of $\theta(x, y)$ at scale s :

$$\begin{pmatrix} W_s^1 f(x, y) \\ W_s^2 f(x, y) \end{pmatrix} = s \vec{\nabla}(f * \theta_s)(x, y). \quad (6.45)$$

The direction of the gradient vector at a point (x, y) indicates the direction in the image plane along which the directional derivative of $f(x, y)$ has the largest absolute value. Edge points (local maxima) can be detected as points (x_0, y_0) such that the modulus of the gradient vector is maximum in the direction toward which the gradient vector points in the image plane. Such computation is closely related to a Canny edge detector [88]. Extension to higher dimension is quite straightforward.

Figure 6.24 provides an example of a multiscale edge detection method based on a first derivative wavelet function.

To further improve the robustness of such a multiscale edge detector, Mallat and Zhong [16] also investigated the relations between singularity (Lipschitz regularity) and the propagation of multiscale edges across wavelet scales. In [89], the dyadic expansion was extended to an M -band expansion to increase directional selectivity. Also, continuous scale representation was used for better adaptation to object sizes [90]. Continuity constraints were applied to fully recover a reliable boundary delineation from 2D and 3D cardiac ultrasound in [91]

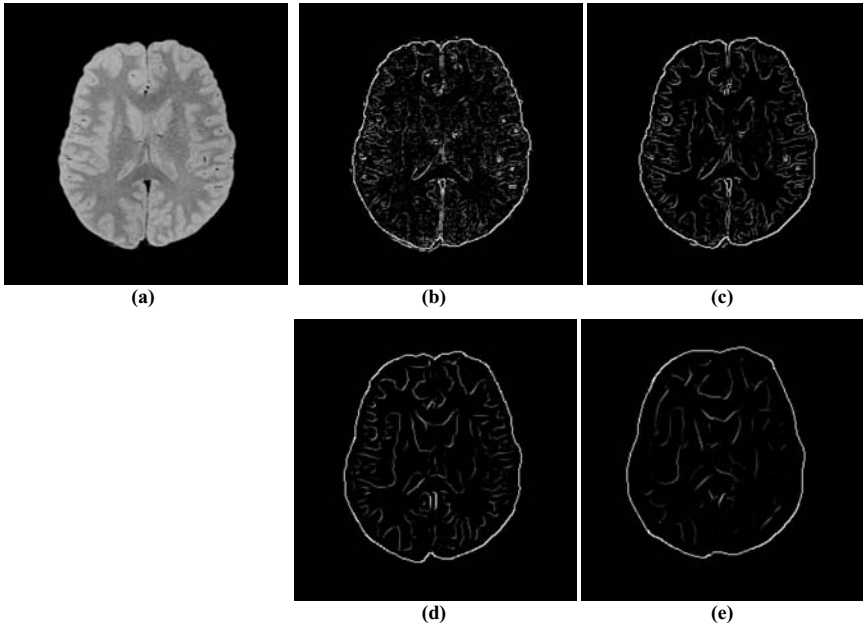


Figure 6.24: Example of a multiscale edge detection method finding local maxima of wavelet modulus, with a first-derivative wavelet function. (a) Input image and (b)–(e) multiscale edge map at expansion scale 1 to 4.

and [92]. In [93], both cross-scale edge correlations and spatial continuity were investigated to improve the edge detection in the presence of noise. Wilson *et al.* in [94] also suggested that a multiresolution Markov model can be used to track boundary curves of objects from a multiscale expansion using a generalized wavelet transform.

Given their robustness and natural representation as boundary information within a multiresolution representation, multiscale edges have been used in deformable model methods to provide a more reliable constraint on the model deformation {Yoshida, 1997 #3686; de Rivaz, 2000 #3687; Wu, 2000 #3688; Sun, 2003 #3689}, as an alternative to traditional gradient-based edge map. In [99], it was used as a presegmentation step in order to find the markers that are used by watershed transform.

6.4.3 Other Wavelet-Based Segmentation

One important feature of wavelet transform is its ability to provide a representation of the image data in a multiresolution fashion. Such hierarchical

decomposition of the image information provides the possibility of analyzing the coarse resolution first, and then sequentially refines the segmentation result at more detailed scales. In general, such practice provides additional robustness to noise and local maxima.

In [100], image data was first decomposed into “channels” for a selected set of resolution levels using a wavelet packets transform. An MRF segmentation was then applied to the subbands coefficients for each scale, starting with the coarsest level and propagating the segmentation result from one level to initialize the segmentation at the next level.

More recently, Davatzikos *et al.* [101] proposed hierarchical active shape models where the statistical properties of the wavelet transform of a deformable contour were analyzed via principal component analysis and used as priors for constraining the contour deformations.

Many research works beneficially used image features within a spatial-frequency domain after wavelet transform to assist the segmentation. In [102] Strickland *et al.* used image features extracted in the wavelet transform domain for detection of microcalcifications in mammograms using a matching process and *a priori* knowledge on the target objects (microcalcification). In [103], Zhang *et al.* used a Bayes classifier on wavelet coefficients to determine an appropriate scale and threshold that can separate segmentation targets from other features.

6.5 Image Registration Using Wavelets

In this section, we give a brief overview of another very important application of wavelets in image processing: image registration. Readers interested in this topic are encouraged to read the references listed in the context.

Image registration is required for many image processing applications. In medical imaging, co-registration problems are important for many clinical tasks:

1. multimodalities study,
2. cross-subject normalization and template/atlas analysis,
3. patient monitoring over time with tracking of the pathological evolution for the same patient and the same modality.

Many registration methods follow a feature matching procedure. Feature points (often referred to as “control points,” or CP) are first identified in both the reference image and the input image. An optimal spatial transformation (rigid or nonrigid) is then computed that can connect and correlate the two sets of control points with minimal error. Registration has always been considered as very costly in terms of computational load. Besides, when the input image is highly deviated from the reference image, the optimization process can be easily trapped into local minima before reaching the correct transformation mapping. Both issues can be alleviated by embedding the registration into a “coarse to fine” procedure. In this framework, the initial registration is carried out on a relatively low resolution image data, and sequentially refined to higher resolution. Registration at higher resolution is initialized with the result from the lower resolution and only needs to refine the mapping between the two images with local deformations for updating the transformation parameters.

The powerful representation provided by the multiresolution analysis framework with wavelet functions has lead many researchers to use a wavelet expansion for such “coarse to fine” procedures [104–106]. As already discussed previously, the information representation in the wavelet transform domain offers a better characterization of key spatial features and signal variations. In addition to a natural framework for “coarse to fine” procedure, many research works also reported the advantages of using wavelet subbands for feature characterization. For example, in [107] Zheng *et al.* constructed a set of feature points from a Gabor wavelet model that represented local curvature discontinuities. They further required that a feature point should have maximum energy among a neighborhood and above a certain threshold. In [108], Moigne *et al.* used wavelet coefficients with magnitude above 13–15% of the maximum value to form their feature space. In [109], Dinov *et al.* applied a frequency adaptive thresholding (shrinkage) to the wavelet coefficients to keep only significant coefficients in the wavelet transform domain for registration.

6.6 Summary

This chapter provided an introduction to the fundamentals of multiscale transform theory using wavelet functions. The versatility of these multiscale

transforms makes them a suitable tool for several applications in signal and image processing that can benefit from the following advantages:

1. A wavelet transform decomposes a signal to a hierarchy of subbands with sequential decrease in resolution. Such expansions are especially useful when a multiresolution representation is needed. Some image segmentation and registration techniques can benefit from a “coarse to fine” paradigm based on a multiresolution framework.
2. A signal can be analyzed with a multiresolution framework into a spatial-frequency representation. By carefully selecting the wavelet function and the space-frequency plane tiling of the transform, distinct components from a noisy observation signal can be easily separated based on their spatial-frequency characteristics.
3. Many important features from an image data can be characterized more efficiently in the spatial-frequency domain. Such feature characterization was shown to be extremely useful in many applications including registration and data compression.

In this chapter we summarized some important applications in medical image processing using wavelet transforms. Noise reduction and enhancement can be easily implemented by combining some very simple linear thresholding techniques with wavelet expansion. Efficient denoising and enhancement improve image quality for further analysis including segmentation and registration.

Feature characteristics in wavelet domain were proven to be potentially more efficient and reliable when compared to spatial analysis only, and therefore provided more effective segmentation and registration algorithms. We point out that many other important applications of multiresolution wavelet transforms, which are beyond the scope of this book, have not been covered in this chapter, especially image compression, which is considered as one of the greatest achievements of wavelet transform in recent years [110]. Other important applications include tomographic image reconstruction, analysis of functional MRI images, and data encoding for MRI acquisition.

Despite the great success of multiresolutions wavelet transform in medical imaging applications for the past 20 years, it continues to be a very active area of research. We list a few resources below that are of interest to readers willing to acquire more knowledge in research and applications in this area.

Conference

SPIE—The International Society for Optical Engineering—has been offering for several years two annual dedicated conferences related to wavelet applications:

1. *Wavelets: Applications in Signal and Image Processing*. (1993–current)
2. *Independent Component Analyses, Wavelets, and Neural Networks* (previously *Wavelet Application*).

These conferences are held annually during the SPIE Annual Meeting and AeroSense conference.

Software

1. *Wavelet Toolbox for MATLAB*: commercial package included in MATLAB (<http://www.mathworks.com>).
2. *Wavelab*: free MATLAB package for wavelet (<http://www-stat.stanford.edu/~wavelab>).
3. *The Rice Wavelet Tools*: MATLAB toolbox for filter bank and wavelets provided by Rice University (<http://www.dsp.ece.rice.edu/software/>).
4. *WVLT*: a wavelet library written in C, which also includes demos and documentation (<http://www.cs.ubc.ca/nest/imager/contributions/bobl/wvlt/top.html>).
5. *LastWave*: a wavelet signal and image processing environment, written in C for X11/Unix and Macintosh platforms. It mainly consists of a powerful command line language with MATLAB-like syntax which includes a high-level object-oriented graphic language (<http://www.cmap.polytechnique.fr/~bacry/LastWave/>).

Web Links

1. www.wavelet.org: offers a “wavelet digest,” an email list that reports most recent news in the wavelet community. It also offers a gallery of links to

many Web resources including books, software, demos, research groups, and tutorials. Important future events are also listed.

2. www.multiresolution.com: includes useful documentation about multiresolution image and data analysis. It also proposes a software package and demos for a wide range of applications.

Questions

1. *What is the uncertainty principle in spatial-frequency analysis? How does the “uncertainty principle” affect the selection of signal representation?*
2. *How “redundant” is an over-complete wavelet expansion? Use an example of a three-dimensional signal, with a five level decomposition using the filter bank implementation shown in Figure 6.5.*
3. *What is the difference between a Gabor transform and a windowed Fourier transform using a Gaussian window?*
4. *What is the difference between a wavelet transform and a wavelet packet transform?*
5. *What is the advantage of temporal analysis in image denoising?*
6. *Why is a true 3D denoising needed for PET/SPECT images?*
7. *Describe the three major components for accomplishing multiscale texture segmentation.*
8. *Between first and second derivatives, which one is preferred for multiscale edge detection?*
9. *What are the two most useful aspects of wavelet transforms in image registration problems?*

Bibliography

- [1] Mallat, S. and Hwang, W. L., Singularity detection and processing with wavelets, *IEEE Trans. Inf. Theory*, Vol. 38, No. 2, pp. 617–643, 1992.
- [2] Strickland, R. N. and Hahn, H. I., Wavelet transform matched filters for the detection and classification of microcalcifications in mammography, In: *Proceedings of the International Conference on Image Processing*, Washington, D.C., Vol. 1, pp. 422–425, 1995.
- [3] Grossman, A. and Morlet, J., Decomposition of Hardy functions into square integrable wavelets of constant shape, *SIAM J. Math. Anal.*, Vol. 15, No. 4, pp. 723–736, 1984.
- [4] Haar, A., Zur Theorie der Orthogonalen Funktionensysteme, *Math. Ann.*, Vol. 69, pp. 331–371, 1910.
- [5] Mallat, S., A theory for multiresolution signal decomposition: The wavelet representation *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 11, No. 7, pp. 674–693, 1989.
- [6] Daubechies, I., Orthonormal bases of compactly supported wavelets, *Commun. Pure Appl. Math.*, Vol. 41, No. 7, pp. 909–996, 1988.
- [7] Unser, M., Aldroubi, A., and Laine, A., *IEEE transactions on medical imaging: Special issue on wavelets in medical imaging*, Vol. 22, No. 3, 2003.
- [8] Weaver, J. B., Yansun, X., Healy, D. M., and Cromwell, L. D., Filtering noise from images with wavelet transforms *Magn. Reson. Med.*, Vol. 21, No. 2, pp. 288–295, 1991.
- [9] Unser, M. and Aldroubi, A., A review of wavelets in biomedical applications *Proceedings of the IEEE*, Vol. 84, No. 4, pp. 626–638, 1996.
- [10] Laine, A., Wavelets in spatial processing of biomedical images, *Ann. Rev. Biomed. Eng.*, Vol. 2, pp. 511–550, 2000.
- [11] Aldroubi, A. and Unser, M., *Wavelets in Medicine and Biology*, CRC Press, Boca Raton, FL, 1996.

- [12] Jain, A. K., Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [13] Papoulis, A., The Fourier Integral and its Applications, McGraw-Hill, New York, NY, 1987.
- [14] Mallat, S., A Wavelet Tour of Signal Processing, Academic Press, San Diego, CA, 1998.
- [15] Daubechies, I., Ten Lectures on Wavelets, Siam, Philadelphia, PA, 1992.
- [16] Mallat, S. and Zhong, S., Characterization of signals from multi-scale edges, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 14, No. 7, pp. 710–732, 1992.
- [17] Holschneider, M., Kronland-Martinet, K., Morlet, J., and Tchamitchian, P., Wavelets, Time Frequency Methods and Phase Space, Springer-Verlag, Berlin, 1989.
- [18] Shensa, M., The discrete wavelet transform: Wedding the *a trous* and mallat algorithms, IEEE Trans. Signal Process., Vol. 40, No. 10, pp. 2464–2482, 1992.
- [19] Koren, I. and Laine, A., A discrete dyadic wavelet transform for multidimensional feature analysis, In: Time Frequency and Wavelets in Biomedical Signal Processing, IEEE Press Series in Biomedical Engineering, M. Akay, Ed., IEEE Press, Piscataway, NJ, pp. 425–448, 1998.
- [20] Feichtinger, H. and Strohmer, T., eds, Gabor Analysis and Algorithms: Theory and Applications, Birkhäuser, Boston, MA, 1998.
- [21] Wickerhauser, M. V., Adapted Wavelet Analysis from Theory to Software, Wellesley, Boston, MA, 1993.
- [22] Meyer, F. and Coifman, R., Brushlets: A tool for directional image analysis and image compression, Appl. Comput. harmonic Anal., Vol. 4, pp. 147–187, 1997.
- [23] Gabor, D., Theory of communication, J. IEE, Vol. 93, pp. 429–457, 1946.

- [24] Bastiaans, M., A sampling theorem for the complex spectrogram and Gabor's expansion of a signal in Gaussian elementary signals, *Opt. Eng.*, Vol. 20, No. 4, pp. 594–598, 1981.
- [25] Porat, M. and Zeevi, Y., The generalized Gabor scheme of image representation in biological and machine vision, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 10, No. 4, pp. 452–468, 1988.
- [26] Hubel, D. and Wiesel, T., Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J. Physiol.*, Vol. 160, pp. 106–154, 1962.
- [27] Daugman, J., Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression, *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. 36, No. 7, pp. 1169–1179, 1988.
- [28] Porat, M. and Zeevi, Y., Localized texture processing in vision: Analysis and synthesis in the Gaborian space, *IEEE Trans. Biomed. Eng.*, Vol. 36, No. 1, pp. 115–129, 1989.
- [29] Coifman, R. R., Meyer, Y., and Wickerhauser, M. V., Wavelet Analysis and signal processing, In: *Wavelets and Their Applications*, B. Ruskai, Ed., Jones and Barlett, Boston, pp. 153–178, 1992.
- [30] Coifman, R. R. and Woog, L. J., Adapted waveform analysis, wavelet packets, and local cosine libraries as a tool for image processing, In: *Investigative and Trial Image Processing*, San Diego, CA, Vol. 2567, 1995.
- [31] Malvar, H., Lapped transforms for efficient transform/subband coding, *IEEE Trans. Acoust. Sign. Speech Process.*, Vol. 38, pp. 969–978, 1990.
- [32] Donoho, D. L. and Johnstone, I. M., Ideal de-noising in an orthonormal basis chosen from a library of bases, Statistics Department, Stanford University, Technical Report, 1994.
- [33] Donoho, D., De-noising by soft-thresholding, *IEEE Trans. Inf. Theory*, Vol. 41, No. 3, pp. 613–627, 1995.
- [34] Gao, H. and Bruce, A., Waveshrink with firm shrinkage, *Statist. Sinica*, Vol. 7, pp. 855–874, 1997.

- [35] Laine, A., Fan, J., and Yang, W., Wavelets for contrast enhancement of digital mammography, *IEEE Eng. Med. Biol.* (September), pp. 536–550, 1995.
- [36] Koren, I., Laine, A., and Taylor, F., Image fusion using steerable dyadic wavelet transform, In: *Proceedings of the International Conference on Image Processing*, Washington, D.C., pp. 232–235, 1995.
- [37] Laine, A., Fan, J., and Schuler, S., A framework for contrast enhancement by dyadic wavelet analysis, In: *Digital Mammography*, A. Gale, S. Astley, D. Dance, and A. Cairns, Eds., Elsevier, Amsterdam, 1994.
- [38] Laine, A., Schuler, S., Fan, J., and Huda, W., Mammographic feature enhancement by multi-scale analysis, *IEEE Trans. Med. Imaging*, Vol. 13, No. 4, pp. 725–740, 1994.
- [39] Fan, J. and Laine, A., multi-scale contrast enhancement and de-noising in digital radiographs, In: *Wavelets in Medicine and Biology*, A. Aldroubi and M. Unser, Eds., CRC Press, Boca Raton FL, pp. 163–189, 1996.
- [40] Coifman, R. and Donoho, D., Translation-invariant de-noising, In: *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, Eds., Springer-Verlag, New York, NY, 1995.
- [41] Donoho, D. and Johnstone, I., Ideal spatial adaptation via wavelet shrinkage, *Biometrika*, Vol. 81, pp. 425–455, 1994.
- [42] Stein, C., Estimation of the mean of a multivariate normal distribution, *Ann. Stat.*, Vol. 9, pp. 1135–1151, 1981.
- [43] Donoho, D., Nonlinear solution of linear inverse problems by wavelet-vaguelette decompositions, *J. Appl. Comput. Harmonic Anal.*, Vol. 2, No. 2, pp. 101–126, 1995.
- [44] Chang, S., Yu, B., and Vetterli, M., Spatially adaptive wavelet thresholding with context modeling for image de-noising, *IEEE Trans. Image Process.*, Vol. 9, No. 9, pp. 1522–1531, 2000.
- [45] Donoho, D. and Johnstone, I., Adapting to unknown smoothness via wavelet shrinkage, *J. Am. Stat. Assoc.*, Vol. 90, No. 432, pp. 1200–1224, 1995.

- [46] Koren, I., A Multi-Scale Spline Derivative-Based Transform for Image Fusion and Enhancement, Ph.D. Thesis, Electrical Engineering, University of Florida, 1996.
- [47] Kalifa, J., Laine, A., and Esser, P., Regularization in tomographic reconstruction using thresholding estimators, *IEEE Trans. Med. Imaging*, Vol. 22, No. 3, pp. 351–359, 2003.
- [48] Selesnick, I., The slantlet transform, *IEEE Trans. Signal Process.*, Vol. 47, No. 5, pp. 1304–1313, 1999.
- [49] Candes, E. and Donoho, D., Curvelets—a surprisingly effective non-adaptive representation for objects with edges, In: *Curve and Surface Fitting: Saint-Malo 1999*, A. Cohen, C. Rabut, and L. Schumaker, Eds., Vanderbilt University Press, Nashville, TN, 1999.
- [50] Starck, J., Candes, E., and Donoho, D., The curvelet transform for image de-noising, *IEEE Trans. Image Process.*, Vol. 11, No. 6, pp. 670–684, 2002.
- [51] Candes, E. and Donoho, D., Ridgelets: The key to higher-dimensional intermittency?, *Phil. Trans. R. Soc. A*, Vol. 357, pp. 2495–2509, 1999.
- [52] Liebling, M., Blu, T., and Unser, M., Fresnelets: New Multiresolution Wavelet Bases for Digital Holography, *IEEE Trans. Image Process.*, Vol. 12, No. 1, pp. 29–43, 2003.
- [53] Gao, H., Wavelet shrinkage de-noising using the non-negative Garrote, *J. Comput. Graph. Stat.*, Vol. 7, pp. 469–488, 1998.
- [54] Antoniadis, A. and Fan, J., Regularization of wavelet approximations, *J. Am. Stat. Assoc.*, Vol. 96, No. 455, pp. 939–967, 2001.
- [55] Nason, G., Wavelet shrinkage using cross-validation, *J. R. Stat. Soc.*, Vol. 58, pp. 463–479, 1996.
- [56] Weyrich, N. and Warhola, G., De-noising using wavelets and cross-validation, *NATA Adv. Study Inst.*, Vol. 454, pp. 523–532, 1995.
- [57] Jansen, M., Malfait, M., and Bultheel, A., Generalised cross-validation for wavelet thresholding, *Signal Process.*, Vol. 56, pp. 33–44, 1997.

- [58] Ogden, R. T. and Parzen, E., Change-point approach to data analytic wavelet thresholding, *Stat. Comput.*, Vol. 6, pp. 93–99, 1996.
- [59] Angelini, E., Laine, A., Takuma, S., Holmes, J., and Homma, S., LV volume quantification via spatio-temporal analysis of real-time 3D echocardiography, *IEEE Trans. Med. Imaging*, Vol. 20, pp. 457–469, 2001.
- [60] Jin, Y., Angelini, E., Esser, P., and Laine, A., De-noising SPECT/PET images using cross-scale regularization, In: *Proceedings of the Sixth International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI 2003)*, Montreal, Canada, Vol. 2879, No. 2, pp. 32–40, 2003.
- [61] Mulet-Parada, M. and Noble, J. A., 2D+T acoustic boundary detection in echocardiography, In: *Medical Image Computing and Computer-Assisted Intervention-MICCAI'98*, Cambridge, MA, pp. 806–813, 1998.
- [62] Chen, C., Lu, H., and Han, K., A textural approach based on Gabor functions for texture edge detection in ultrasound images, *Ultrasound Med. Biol.*, Vol. 27, No. 4, pp. 515–534, 2001.
- [63] McLachlan, G. J. and Krishnan, T., *The EM Algorithm and Extensions*, Wiley & Sons, Inc., New York, 1997.
- [64] Shepp, L. and Vardi, V., Maximum likelihood reconstruction for emission computed tomography, *IEEE Trans. Med. Imaging*, Vol. 1, pp. 113–122, 1982.
- [65] Farquhar, T. H., Chatziioannou, A., Chinn, G., Dahlbom, M., and Hoffman, E. J., An investigation of filter choice for filtered back-projection reconstruction in PET, *IEEE Trans. Nucl. Sci.*, Vol. 45 (3 Part 2), pp. 1133–1137, 1998.
- [66] Hudson, H. and Larkin, R., Accelerated image reconstruction using ordered subsets of projection data, *IEEE Trans. Med. Imaging*, Vol. 13, No. 4, pp. 601–609, 1994.
- [67] Freeman, W. and Adelson, E., The design and use of steerable filters, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 13, pp. 891–906, 1991.

- [68] Babaud, J., Witkin, A., Baudin, M., and Duba, R., Uniqueness of the Gaussian kernel for scale-space filtering, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, pp. 26–33, 1986.
- [69] Julesz, B., A Theory of preattentive texture discrimination based on first-order statistics of textons, *Biol. Cybern.*, Vol. 41, pp. 131–138, 1981.
- [70] Watson, A., Barlow, H., and Robson, J., What dose the eye see best?, *Nature*, Vol. 302, pp. 419–422, 1983.
- [71] Beck, J., Sutter, A., and Ivry, R., Spatial frequency channels and perceptual grouping in texture segregation, *Comput. Vis., Graph. Image Process.*, Vol. 37, pp. 299–325, 1987.
- [72] Daugman, J., Image analysis by local 2-D spectral signatures, *J. Opt. Soc. Am. A*, Vol. 2, pp. 74, 1985.
- [73] Unser, M., Texture classification and segmentation using wavelet frames, *IEEE Trans. Image Process.*, Vol. 4, No. 11, pp. 1549–1560, 1995.
- [74] Laine, A. and Fan, J., Texture classification by wavelet packet signatures, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 15, No. 11, pp. 1186–1191, 1993.
- [75] Laine, A. and Fan, J., Frame representation for texture segmentation, *IEEE Trans. Image Process.*, Vol. 5, No. 5, pp. 771–780, 1996.
- [76] Hsin, H. and Li, C., An experiment on texture segmentation using modulated wavelets, *IEEE Trans. Syst., Man Cybern.*, Vol. 28, No. 5, pp. 720–725, 1998.
- [77] Wang, J., Multiwavelet packet transform with application to texture segmentation, *Electron. Lett.*, Vol. 38, No. 18, pp. 1021–1023, 2002.
- [78] Acharyya, M. and Kundu, M., Document image segmentation using wavelet scale-space features, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 12, No. 12, pp. 1117–1127, 2002.

- [79] Wang, J., Li, J., Gray, R., and Wiederhold, G., Unsupervised multiresolution segmentation for images with low depth of field, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 23, No. 1, pp. 85–90, 2001.
- [80] Etemad, K., Doermann, D., and Chellappa, R., Multi-scale segmentation of unstructured document pages using soft decision integration, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 19, No. 1, pp. 92–96, 1997.
- [81] Porter, R. and Canagarajah, N., A robust automatic clustering scheme for image segmentation using wavelets, *IEEE Trans. Image Process.*, Vol. 5, No. 4, pp. 662–665, 1996.
- [82] Zhang, J., Wang, D., and Tran, Q., A wavelet-based multiresolution statistical model for texture, *IEEE Trans. Image Process.*, Vol. 7, No. 11, pp. 1621–1627, 1998.
- [83] Choi, H. and Baraniuk, R., Multiscale image segmentation using wavelet-domain hidden markov models, *IEEE Trans. Image Process.*, Vol. 10, No. 9, pp. 1309–1321, 2001.
- [84] Li, J. and Gray, R., Context-based multi-scale classification of document images using wavelet coefficient distributions, *IEEE Trans. Image Process.*, Vol. 9, No. 9, pp. 1604–1616, 2000.
- [85] Charalampidis, D. and Kasparis, T., Wavelet-based rotational invariant roughness features for texture classification and segmentation, *IEEE Trans. Image Process.*, Vol. 11, No. 8, pp. 825–837, 2002.
- [86] Chan, T. F. and Vese, L. A., Active contours without edges, *IEEE Trans. Image Process.*, Vol. 10, No. 2, pp. 266–277, 2001.
- [87] Yezzi, A., Tsai, A., and Willsky, A., A statistical approach to image segmentation for bimodal and trimodal imagery, *ICCV*, pp. 898–903, 1999.
- [88] Canny, J., A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, No. 6, pp. 679–698, 1986.
- [89] Aydin, T., Yemez, Y., Anarim, E., and Sankur, B., Multi-directional and multi-scale edge detection via M-band wavelet Transform, *IEEE Trans. Image Process.*, Vol. 5, No. 9, pp. 1370–1377, 1996.

- [90] Laine, A. F., Huda, W., Chen, D., and Harris, J. G., Local enhancement of masses using continuous scale representations, *J. Math. Imaging Vis.*, Vol. 7, No. 1, 1997.
- [91] Laine, A., and Zong, X., Border identification of echocardiograms via multi-scale edge detection and shape modeling, In: *IEEE International Conference on Image Processing*, Lausanne, Switzerland, pp. 287–290, 1996.
- [92] Koren, I., Laine, A. F., Fan, J., and Taylor, F. J., Edge detection in echocardiographic image sequences by 3-D multiscale analysis, *IEEE International Conference on Image Processing*, Vol. 1, No. 1, pp. 288–292, 1994.
- [93] Dima, A., Scholz, M., and Obermayer, K., Automatic segmentation and skeletonization of neurons from confocal microscopy images based on the 3-D wavelet transform, *IEEE Trans. Image Process.*, Vol. 11, No. 7, pp. 790–801, 2002.
- [94] Wilson, R., Calway, A., and Pearson, R., A generalized wavelet transform for Fourier analysis: The multiresolution Fourier transform and its application to image and audio signal analysis, *IEEE Trans. Inf. Theory*, Vol. 38, No. 2, pp. 674–690, 1992.
- [95] Yoshida, H., Katsuragawa, S., Amit, Y., and Doi, K., Wavelet snake for classification of nodules and false positives in digital chest radiographs, In: *IEEE EMBS Annual Conference*, Chicago, IL, pp. 509–512, 1997.
- [96] deRivaz, P. and Kingsbury, N., Fast Segmentation using level set curves of complex wavelet surfaces, In: *IEEE International Conference on Image Processing*, Vol. 3, pp. 29–32, 2000.
- [97] Wu, H., Liu, J., and Chui, C., A wavelet frame based image force model for active contouring algorithms, *IEEE Trans. Image Process.*, Vol. 9, No. 11, pp. 1983–1988, 2000.
- [98] Sun, H., Haynor, D., and Kim, Y., Semiautomatic video object segmentation using VSnakes, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 1, pp. 75–82, 2003.

- [99] Neves, S., daSilva, E., and Mendonca, G., Wavelet-watershed automatic infrared image segmentation method, *IEEE Electron. Lett.*, Vol. 39, No. 12, pp. 903–904, 2003.
- [100] Bello, M., A combined Markov random field and wave-packet transform-based approach for image segmentation, *IEEE Trans. Image Process.*, Vol. 3, No. 6, pp. 834–846, 1994.
- [101] Davatzikos, C., Tao, X., and Shen, D., Hierarchical active shape models using the wavelet transform, *IEEE Trans. Med. Imaging*, Vol. 22, No. 3, pp. 414–423, 2003.
- [102] Strickland, R. N. and Hahn, H. I., Wavelet transforms for detecting microcalcifications in mammograms, *IEEE Trans. Med. Imaging*, Vol. 15, No. 2, pp. 218–229, 1996.
- [103] Zhang, X. and Desai, M., Segmentation of bright targets using wavelets and adaptive thresholding, *IEEE Trans. Image Process.*, Vol. 10, No. 7, pp. 1020–1030, 2001.
- [104] Allen, R., Kamangar, F., and Stokely, E., Laplacian and orthogonal wavelet pyramid decompositions in coarse-to-fine registration, *IEEE Trans. Signal Process.*, Vol. 41, No. 12, pp. 3536–3541, 1993.
- [105] Unser, M., Thevenaz, P., Lee, C., and Ruttimann, U., Registration and statistical analysis of PET images using the wavelet transform, *IEEE Eng. Med. Biol.* (September/October), pp. 603–611, 1995.
- [106] McGuire, M. and Stone, H., Techniques for multiresolution image registration in the presence of occlusions, *IEEE Trans. Geosci. Remote Sensing*, Vol. 38, No. 3, pp. 1476–1479, 2000.
- [107] Zheng, Q. and Chellappa, R., A computational vision approach to image registration, *IEEE Trans. Image Process.*, Vol. 2, No. 3, pp. 311–325, 1993.
- [108] Moigne, J., Campbell, W., and Crompton, R., Automated parallel image registration technique based on the correlation of wavelet features, *IEEE Trans. Geosci. Remote Sensing*, Vol. 40, No. 8, pp. 1849–1864, 2002.

- [109] Dinov, I., Mega, M., Thompson, P., Woods, R., Sumners, D., Sowell, E., and Toga, A., Quantitative comparison and analysis of brain image registration using frequency-adaptive wavelet shrinkage, *IEEE Trans. Inf. Technol. Biomed.*, Vol. 6, No. 1, pp. 73–85, 2002.
- [110] Unser, M. and Blu, T., Mathematical properties of the JPEG2000 wavelet filters, *IEEE Trans. Image Process.*, Vol. 12, No. 9, pp. 1080–1090, 2003.

Chapter 7

Improving the Initialization, Convergence, and Memory Utilization for Deformable Models

Gilson A. Giraldi¹, Paulo S. Rodrigues¹, Leandro S. Marturelli¹, and Rodrigo L. S. Silva¹

7.1 Introduction

In this chapter our aim is twofold. Firstly, we point out some limitations of deformable models for medical images and analyze recent works to overcome these limitations. Next, we offer new perspectives in the area, which are part of our current research in this field.

Deformable models, which include the popular *snake models* [42] and deformable surfaces [19, 48], are well-known techniques for tracking, boundary extraction, and segmentation in 2D/3D images.

Basically, these models can be classified into three categories: parametric, geodesic snakes, and implicit models. The relationships between these models have been demonstrated in several works in the literature [57, 75].

Parametric deformable models consist of a curve (or surface) which can dynamically conform to object shapes in response to internal (elastic) forces and external forces (image and constraint ones) [6].

For geodesic snakes, the key idea is to construct the evolution of a contour as a geodesic computation. A special metric is proposed (based on the gradient of the image field) to let the minimal length curve correspond to the desired boundary. This approach allows one to address the parameterization

¹National Laboratory for Scientific Computing, Brazil

dependence of parametric snake models and can be extended to 3D through the theory of minimal surfaces [11, 57].

Implicit models, such as the formulation used in [46], consist of embedding the snake as the zero level set of a higher dimensional function and to solve the corresponding equation of motion. Such methodologies are best suited to the recovery of objects with unknown topologies.

Parametric deformable models are more intuitive than the implicit and geodesic ones. Their mathematical formulation makes it easier to integrate image data, initial estimated, desired contour properties and knowledge-based constraints, in a single extraction process [6].

However, parametric models also have their limitations. First, most of these methods can only handle objects with simple topology. The topology of the structures of interest must be known in advance since the mathematical model cannot deal with topological changes without adding extra machinery [21–47]. Second, parametric snakes are too sensitive to their initial conditions due to the nonconvexity of the energy functional and the contraction force which arises from the internal energy term [37, 79]. Several works have been done to address the mentioned limitations.

Topological restrictions can be addressed through a two-step approach: firstly, a method of identifying the necessity of a topological operation (split or merge) and secondly, a procedure of performing it. In [21] we found such a methodology that can split a closed snake into two closed parts. This is accomplished by first constructing a histogram of the image force norm along the snake to identify the appropriate region to cut it (region with weakest image field). Next, the method identifies two points in this region to be the end points of the segment which will cut the curve into two parts. The criterion to do this is based on the direction of an area force used to make the contour fit concave parts. This methodology has the disadvantages of not dealing with the contour merges and its extension to the 3D case is very difficult.

In [65] another approach is presented. It seeds particles on the surface of an object until their density on the surface is within some threshold value. Its components are a dynamical particle system and an efficient triangulation scheme which connects the particles into a continuous polygonal surface model consistent with the particles configuration. Particles are oriented; that is, each one has a position and a normal vector associated. The interparticle forces are used to encourage neighboring oriented particles to lie in each other's tangent planes, and therefore favor smooth surfaces. This technique has the advantage of dealing

easily with open and closed surfaces. The topology of the particle-based surface can be modified during the triangulation step. However, this has the disadvantages of being expensive ($O(N) \log N$) where N is the number of particles) and that it may be difficult or cumbersome to find good initial seed particle sites, especially automatically [50].

A more general approach to incorporate topological changes in the parametric snake models is the T-snakes model [47–50]. The method embeds the snake model within a framework defined by a simplicial domain decomposition, using classical results in the field of numerical continuation methods [1]. The resulting model has the power of an implicit one without the need for a higher dimensional formulation [46]. Besides, it can be efficiently extended to 3D, generating the T-surfaces model [49].

The sensitivity to the initialization is a very common problem for deformable models. The use of simulated annealing for minimization was proposed in [62]. Despite the global optimization properties, the use of this technique is limited to both its computational complexity and memory requirements.

Levine *et al.* [44] applied hierarchical filtering methods, as well as a continuation method based on a discrete scale-space representation. At first, a scale-space scheme is used at a coarse scale to get closer to the global energy minimum represented by the desired contour. In further steps, the optimal valley or contour is sought at increasingly finer scales.

These methods address the nonconvexity problem but not the adverse effects of the internal normal force. This force is a contraction force which makes the curve collapse into a point if the external field is not strong enough. In Cohen [18] and Gang *et al.* [79] this problem is addressed by the addition of another internal force term to reduce the adverse effects of the contraction force. In both works the number of parameters is increased if compared with the original model and there are some trade-offs between efficiency and performance.

Another way to remove the undesired contraction force of the original snake model is to use the concept of invariance, which is well known in the field of computer vision [26, 36]. This concept has been applied to closed contours, and consists in designing an internal smoothing energy, biased toward some prior shape, which has the property of being invariant to scale, rotation, and translation. In these models, the snake has no tendency to expand or contract, but it tends to acquire a natural shape.

An example of a technique, which applies invariance concepts, is the dual active contour (dual ACM) [37]. This approach basically consists of one contour

which expands from inside the target feature, and another one which contracts from the outside. The two contours are interlinked to provide a driving force to carry the contours out of local minima, which makes the solution less sensitive to the initial position.

The sensitivity to initialization of snakes can also be addressed by a two-stage approach: (1) The region of interest is limited; and (2) a global minimization technique is used to find the object boundary. Bamford and Lovell [4] describe such a method to segment cell nucleus based on a dynamic programming algorithm (Viterbi algorithm) to find the solution.

The use of dynamic programming (DP) for solving variational problems is discussed by Amini *et al.* [2]. Unlike the variational approach, DP ensures global optimality of the solution and does not require estimates of higher order derivatives, which improves the numerical stability. However, these techniques are limited by their storage requirements of $O(NM^2)$ and computational complexity of $O(NM^3)$, where N is the number of snaxels and M is the size of the neighborhood around each snaxel (given a *discrete search space* with NM points). These performance difficulties can be lowered with a method to reduce the search space. That is the main point addressed in [32,34].

In those works, we propose to reduce the search space through the dual-T-snakes model [30] by its ability to get closer to the desired boundaries. The result is two contours close to the border bounding the search space. Hence, a DP algorithm [2,4,38] can be used more efficiently.

The sensitivity to the initial contour position can also be addressed by a method which initializes automatically the snake closer to the boundaries [43]. An efficient methodology in this field would be worthwhile, not only to save time/calculation, but also to facilitate the specification of parameters, a known problem for snake models [31].

In [29,33] we propose a method to initialize deformable models, which is based on properties related to the topology and spatial scale of the objects in 2D or 3D scenes. We assume some topological and scale properties for the objects of interest. From these constraints we propose a method which first defines a triangulation of the image domain. After that, we take a subsampling of the image field over the grid nodes. This field is thresholded, generating a binary one, an “object characteristic function,” from which a rough approximation of the boundary geometry is extracted. This method was extended to 3D in [63].

Neural networks and Hough transforms have also been applied for initialization of deformable models [14, 74].

An other possibility to address the sensitivity to initialization is the gradient vector flow, which is a scheme based on a vector diffusion–reaction equation. It was introduced in [77] and can be used to obtain a more efficient image force field [78].

Deformable models can be extended to 3D, generating deformable surface models. Besides the described problems, a new one arises when considering these models: memory utilization.

In general, deformable surface models make use of only the data information along the surface when evolving the model toward the object boundary [48, 49]. However, state-of-the-art implementations of these models in general do not account for this fact and fetch the whole volume from disk at the initialization. Such a procedure brings limitations for large size image volumes, mainly if we consider that, in general, deformable models need not only the image intensity but also the image gradient [42, 49].

Nowadays, image volumes with 512^3 sampling points can be acquired in CT scanners. Besides, other scanning techniques were developed allowing the acquisition of a huge amount of 3D color image volumes (www.nlm.nih.gov/research/visible/visible_human.html). In these cases, the data set information (image intensity and gradient) can be too large to fit in main memory, even if we take the usual cut policy: In a first stage, select a subvolume (a bounding box) that contains the structure of interest, and then segment it. When the size of the data that must be accessed is larger than the size of main memory, some form of virtual memory is simply required, which leads to performance problems [20].

The analysis of large data sets is a known problem in the context of scientific visualization [15, 24, 71]. Out-of-core techniques have been developed for scalar and vector fields visualization and new proposals are still in progress. Among these methods, out-of-core isosurface extraction techniques are closely related with our work, as we shall see next.

These methods partition the data set into clusters that are stored in disk blocks, and build a data structure to index the blocks for information retrieval (*preprocessing step*). At run-time, the data structure is read to main memory and traverse to find out the data blocks that must be read to main memory to perform the isosurface generation. The most commonly used data structures, for

scientific visualization applications, are the octrees [64, 71] and a k - d -tree-based technique called *meta-cell* [15].

In [27, 28] we show that the meta-cell technique is the most suitable data structure to perform out-of-core implementations of segmentation methods. We take advantage of the meta-cell method to present an out-of-core implementation of the segmentation approach proposed in [63]. This method is a straightforward extension of the initialization method that we proposed in [26, 29].

The core of the algorithm is an out-of-core T-surfaces method based on the meta-cell structure. To our knowledge, it is the first out-of-core algorithm for deformable surface model reported in the literature. Besides, other parametric deformable models as well as implicit models (level sets) and region growing methods can be out-of-core implemented by using the same meta-cell structure (see Section 7.10). It is important to highlight that the proposed structure is useful not only to efficiently swap data between memory and disk, but also to accelerate the segmentation process, as we shall demonstrate (Section 7.9).

To make this text self-contained, some background is offered in Section 7.2. We describe the deformable model methods that will be used in this chapter.

Next, the initialization techniques of interest are described (Section 7.3). We survey the most important works in this subject and show that their basic limitation is that the obtained contour may suffer self-intersections during its evolution. Thus, a deformable model that can deal with such a problem is necessary. T-snakes (or T-surfaces) is a possibility.

Thus, in Section 7.4 we describe an efficient method to initialize the T-surfaces model, which encompasses the basic elements of the segmentation approach presented on Section 7.5. Despite the capabilities of our segmentation approach, we may have problems due to memory limitations for large datasets and poor convergence for noisy images. These problems are considered in Sections 7.6 and 7.8, respectively.

Finally, discussions and perspectives for deformable models in medical images are offered (Section 7.10). Conclusions are given in Section 7.11.

7.2 Background in Deformable Models

In some sense, deformable models used in segmentation and shape recovery applications can be classified into two groups: *free form* and *shape models* [53].

In shape models prior knowledge of the global structure is included using a parameterized template of a specific structure. Free form deformable templates, like snakes, have no explicit global structures as the prior knowledge includes basically local continuity and smoothness constraints.

Considering as a functional energy minimization process, the snake model consists of an initial model which is carried to the desired object boundary by forces described by the Euler–Lagrange equations. In a different way, the snake evolution can be formulated by local deformations to reshape dynamically the initial model in a process which do not apply minimization techniques explicitly. The former is the formulation used by Kass *et al.* [42] in the original snake model. It will be described next.

7.2.1 Original Model

Geometrically, a snake is a parametric contour c , here assumed to be closed, embedded in a domain $D \subset \mathbb{R}^2$:

$$c : [0, 1] \rightarrow D \subset \mathbb{R}^2, \quad c(s) = (x(s), y(s)). \quad (7.1)$$

We can define a deformable model as a space of admissible deformations (contours) Ad and a functional E to be minimized [18]. This functional represents the energy of the model and has the form:

$$\begin{aligned} E : Ad &\rightarrow \mathbb{R}, \\ E(c) &= E_1(c(s)) + E_2(c(s)), \end{aligned} \quad (7.2)$$

where

$$E_1 = \int_{\Omega} \left(w_1 \|c'(s)\|^2 + w_2 \|c''(s)\|^2 \right) ds, \quad (7.3)$$

$$E_2 = \int_{\Omega} P(c(s)) ds \quad (7.4)$$

are the internal and external energy terms, respectively. In the internal energy expression, the parameter w_1 (tension) gives the snake the behavior of resisting the stretch and w_2 (rigidity) makes the snake less flexible and smoother. These parameters can be constants or dependent on s [44]. Each prime denotes a degree of differentiation with respect to the parameter s .

In the external energy E_2 , P is a potential related with the features we seek. For edge detection in a grayscale image a possible definition is [6]:

$$P = -\|\nabla I\|^2, \quad (7.5)$$

where I is the image intensity.

The process of minimizing the functional given in (7.2) can be viewed from a dynamic point of view by using the Lagrangian mechanics. This leads to dynamic deformable models that unify the description of shape and motion. In these models the deformable contour is viewed as a time-varying curve:

$$c(s, t) = (x(s, t), y(s, t)), \quad (7.6)$$

with a mass density μ and a damping density γ .

In this formulation, the Lagrange equations of motion for a snake with potential energy given by expression (7.2) have the form [44, 50]:

$$\mu \frac{\partial^2 c}{\partial t^2} + \gamma \frac{\partial c}{\partial t} + (w_1 c'(s))' + (w_2 c''(s))'' + \nabla P(c(s)) = 0, \quad (7.7)$$

where the first two terms represent the inertial and damping forces while the third and fourth terms give the forces related to the internal energy (Eq. (7.2)). The last term in Eq. (7.7) is the external force due to the external potential P in expression (7.5). Equilibrium is achieved when the internal and external forces balance and the contour comes to rest; which implies that:

$$\partial c / \partial t = \partial^2 c / \partial t^2 = 0. \quad (7.8)$$

In general, Eq. (7.7) does not have analytical solutions. Thus, numerical methods must be considered. Henceforth, in order to solve this equation, for an initial closed contour, we have to discretize the snake in space and time by using *finite differences* or *finite elements* methods, each of them with trade-offs between performance and numerical efficiency [19, 44]. We also have to use a *termination condition*, based on Eq. (7.8), to stop the numerical interactions [44].

It is important to observe that the space Ad in expression (7.2) does not include contours with more than one connected component. So the classical snake model does not incorporate topological changes of the contour c during its evolution given by Eq. (7.7). Besides, the contraction force generated by the third and fourth terms in this equation is shape dependent and makes the stabilization of the snake too dependent on the parameters w_1 and w_2 . While in theory it is possible to compute a pair of proper weights of the internal energy for each point, it is very difficult in practice [79].

For boundary extraction and segmentation tasks, in general we use a simplified version of Eq. (7.7) in which we take $\mu = 0$. Hence, the model has no inertial forces, which avoids oscillations near the equilibrium point [31].

Snake models can be extended to 3D, generating deformable surface models. The traditional mathematical description for these models is given next.

7.2.2 Deformable Surfaces

Let us consider the following balloon-like model for closed surfaces [19]:

$$v : \mathbb{R}^+ \times [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3, \quad v(t, r, s) = (v_1(t, r, s), v_2(t, r, s), v_3(t, r, s)),$$

$$\frac{\partial v}{\partial t} - \omega_{10} \frac{\partial^2 v}{\partial s^2} - \omega_{01} \frac{\partial^2 v}{\partial r^2} + 2\omega_{11} \frac{\partial^4 v}{\partial r^2 \partial s^2} + \omega_{20} \frac{\partial^4 v}{\partial s^4} + \omega_{02} \frac{\partial^4 v}{\partial r^4} = F(v) - kn(v), \quad (7.9)$$

Initial estimation : $v(0, r, s) = v_0(r, s)$,

where $n(v)$ is the normal (unitary) field over the surface v , F is the image force field (may be normalized), and k is a force scale factor. The parameters ω_{ij} control the smoothing and flexibility of the model.

By using the internal pressure force ($kn(v)$), the model behaves like a balloon, which is inflated, passing over regions in which the external force is too weak. Consequently, the model becomes less sensitive to initialization, which is an advantage over more traditional active models [6, 18].

If finite differences is used to numerically solve Eq. (7.9), the continuous surface $v(r, s)$ is discretized, generating a polygonal mesh. During the mesh evolution, self-intersections must be avoided.

This problem has been efficiently addressed in the context of *discrete deformable models*. Differently from the above formulation, in which the mesh arises due to a discretization of the continuous model (defined by Eq. (7.9)), discrete surface models start from a two-dimensional mesh. The mesh nodes are updated by a system of forces that resembles a discrete dynamical system. The T-surfaces model is such a system, which is fundamental for our work. It is summarized next.

7.2.3 T-Surfaces

The T-surfaces approach is composed of three components [49]: (1) a tetrahedral decomposition (CF-triangulation) of the image domain $D \subset \mathbb{R}^3$; (2) a particle

model of the deformable surface; and (3) a *characteristic function* χ defined on the grid nodes which distinguishes the interior ($\text{Int}(S)$) from the exterior ($\text{Ext}(S)$) of a surface S :

$$\chi : D \subset \mathbb{R}^3 \rightarrow \{0, 1\} \quad (7.10)$$

where $\chi(p) = 1$ if $p \in \text{Int}(S)$ and $\chi(p) = 0$, otherwise p is a node of the grid.

Following the classical nomenclature [1], a tetrahedron (also called a simplex) σ is a *transverse* one if the characteristic function χ in Eq. (7.10) changes its value in σ . Analogously, this follows for an edge.

In the framework composed of both the simplicial decomposition and the characteristic function, the reparameterization of a surface is done by [49]: (1) computing the intersection points of the surface with the grid; (2) finding the set of transverse tetrahedra (*combinatorial manifold*); (3) choosing an intersection point, for each transverse edge; and (4) connecting the selected points.

In this reparameterization process, the transverse simplices play a central role. Given such a simplex, we choose in each transverse edge an intersection point to generate the new surface patch. In general, we will obtain three or four transverse edges in each transverse tetrahedron (Fig. 7.1). The former gives a triangular patch and the latter defines two triangles. So, at the end of step (4), a triangular mesh is obtained. Each triangle is called a *triangular element* [49].

Taking a 2D example, let us consider the characteristic functions (χ_1 and χ_2) relative to the two contours pictured in Fig. 7.2. The functions are defined on the vertices of a CF-triangulation of the plane. The vertices marked are those where $\max\{\chi_1, \chi_2\} = 1$. Observe that they are enclosed by a merge of the contours. This merge can be approximated by a curve belonging to the region obtained by tracing the transverse triangles. The same would be true for more than two contours (and obviously for only one).

After the reparameterization process, a suitable evolution scheme must be applied. Dynamically, a T-surfaces can be seen as a closed elastic mesh [49].

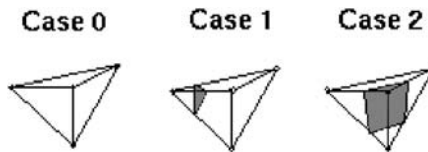


Figure 7.1: Basic types of intersections between a plane and a simplex in 3D.

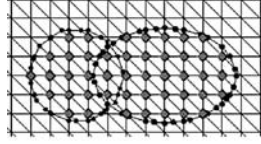


Figure 7.2: Two snakes colliding with the inside grid nodes and snaxels marked.

Each mesh node is called a *node element* and each pair of connected nodes v_i, v_j is called a *model element*.

The node elements are linked by springs, whose natural length we set to zero. Hence, a tensile force can be defined by:

$$\vec{\alpha}_i = \sum_j \vec{S}_{ij}, \quad \text{where } \vec{S}_{ij} = c \cdot r_{ij}, \quad (7.11)$$

c is a scale factor and $r_{ij} = \|v_i - v_j\|$ is the length of the corresponding model element. The model also has a normal force which can be weighted as follows [49]:

$$F_i = k(\text{sign}_i)n_i, \quad (7.12)$$

where n_i is the normal vector at node i , k is a scale factor, and $\text{sign}_i = +1$ if $I(v_i) > T$ and $\text{sign}_i = -1$ otherwise (T is a threshold of the image I). This force is used to push the model toward image edges until it is opposed by external image forces.

The forces defined by Eqs. (7.11) and (7.12) are internal forces. The external force is defined as a function of the image data, according to the interested features. Several different approaches have been adopted according to the application [34, 77]. In our case, it can be defined as follows:

$$\text{image} :: \text{force} :: f_i^t = -\gamma_i \nabla P, \quad P = \|\nabla I\|^2. \quad (7.13)$$

The evolution of the surface is controlled by the following dynamical system:

$$v_i^{(t+\Delta t)} = v_i^t + h_i \left(\vec{\alpha}_i^t + \vec{F}_i^t + \vec{f}_i^t \right), \quad (7.14)$$

where h_i is an evolution step.

During the T-surfaces evolution, some grid nodes become interior to a surface. Such nodes are called *burnt nodes* and its identification is required by the update of the characteristic function [49]. To deal with self-intersections, the T-surfaces model incorporates an entropy condition: *Once a node is burnt it*

stays burnt. A termination condition is set based on the number of deformation steps in which a simplex has remained a transverse one.

7.2.4 Level Sets

It will be useful to review some details of *level sets*, which is the implicit formulation presented in [46]. The main idea of this method is to represent the deformable surface (or curve) as a level set $\{x \in \mathbb{R}^3 | G(x) = 0\}$ of an embedding function:

$$G : \mathbb{R}^3 \times \mathbb{R}^+ \rightarrow \mathbb{R}, \quad (7.15)$$

such that the deformable surface (also called *front* in this formulation), at $t = 0$, is given by a surface S :

$$S(t = 0) = \{x \in \mathbb{R}^3 | G(x, t = 0) = 0\}. \quad (7.16)$$

The next step is to find an Eulerian formulation for the front evolution. Following Sethian [46], let us suppose that the front evolves in the normal direction with velocity $\vec{F}$, where $\vec{F}$ may be a function of the curvature, normal direction, etc.

We need an equation for the evolution of $G(x, t)$, considering that the surface S is the level set given by:

$$S(t) = \{x \in \mathbb{R}^3 | G(x, t) = 0\}. \quad (7.17)$$

Let us take a point $x(t)$, $t \in \mathbb{R}^+$, of the propagating front S . From its implicit definition given above, we have:

$$G(x(t), t) = 0. \quad (7.18)$$

Now, we can use the chain rule to compute the time derivative of this expression:

$$G_t + F |\nabla G| = 0, \quad (7.19)$$

where $F = \|\vec{F}\|$ is called the *speed function*. An initial condition $G(x, t = 0)$ is required. A straightforward (and expensive) technique to define this function is to compute a signed-distance function as follows:

$$G(x, t = 0) = \pm d, \quad (7.20)$$

where d is the distance from x to the surface $S(x, t = 0)$ and the signal indicates if the point is interior ($-$) or exterior ($+$) to the initial front.

Finite difference schemes, based on a uniform grid, can be used to solve Eq. (7.19). The same entropy condition of T-surfaces (*once a grid node is burnt it stays burnt*) is incorporated in order to drive the model to the desired solution (in fact, T-surfaces was inspired on the level sets model [50]).

In this higher dimensional formulation, topological changes can be efficiently implemented. Numerical schemes are stable, and the model is general in the sense that the same formulation holds for 2D and 3D, as well as for merge and splits. Besides, the surface geometry is easily computed. For example, the front normal and curvature are given by:

$$\vec{n} = \nabla G(x, t), \quad K = \nabla \cdot \left(\frac{\nabla G(x, t)}{\|\nabla G(x, t)\|} \right), \quad (7.21)$$

respectively, where the gradient and the divergent ($\nabla \cdot$) are computed with respect to x .

7.3 Initializing Traditional Deformable Models

In the area of deformable models, the definition of the initial estimation (see Eq. (7.9)) from which we can start the model evolution (the initialization step) is a difficult and important task. Problems associated with fitting the model to data could be reduced if a better start point for the search were available. In this section, we show a set of methods used to find the initial curve (or surface).

We start with methods that use image statistics and morphological techniques, and later we present modern approaches, such as neural nets.

7.3.1 Region-Based Approaches

The simplest way to initializing deformable models is through a preprocessing step in which the structures of interest are enhanced.

This can be done by image statistics extracted by image histograms or pattern recognition techniques [69] (see [39] for a recent review). These statistics can be represented by a mean μ and variance σ of the image field I or any other field

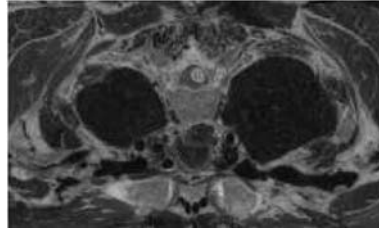


Figure 7.3: Original grayscale image of human's torso.

defined over the image domain (fuzzy fields [33, 76], for example). The aim is to find statistical representation of the objects, which means:

$$p \in O \implies |I(p) - \mu| \leq k\sigma, \quad (7.22)$$

where k is an used defined parameter [49].

In some applications, a threshold T could be enough to characterize the object(s). *Iterative* and *entropy* methods can be obtained by simple inspection [40].

For an illustrative example, Fig. 7.3 shows an image of a cross section slice of a human's torso, where we can see several interesting regions such as arteries, bones, and lungs (the two largest central black regions).

Suppose we are interested in extracting the boundary of the right lung. First of all, we should isolate, in each slice, the region of interest.

Applying Eq. (7.22) with a threshold, e.g. 30, we have as output the result pictured in Fig. 7.4(a). Thus, an isoline extraction method can be used to get a rough approximation of the target boundary. Figure 7.4(b) shows the obtained curve over the original data.

We can observe that the curve is not smooth, there are protrusions and concavities due to inhomogeneities of the image field. Besides, some regions of interest may be merged (or even slit) after binarization. Such difficulties

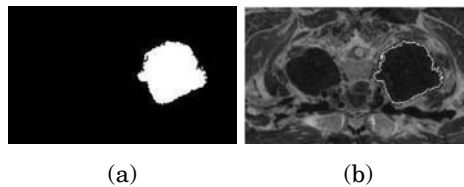


Figure 7.4: (a) Result of applying a threshold $T = 30$ over image of Fig. 7.3. (b) Initialization through isoline extraction.

arise even when the images are preprocessed with more robust segmentation approaches, such as image foresting transformation [23] or other fuzzy techniques [70, 76]. These problems make threshold-based methods not very adequate for deformable models initialization.

In the following section, we discuss an approach to improve the automatic detection of an initial curve.

7.3.2 Mathematical Morphology for Initialization

The use of mathematical morphology to initialize deformable models is a subject with few references in the literature [59, 76].

For the particular case of medical images, the general idea is to isolate objects of interest (such as lungs, arteries, heart, bones, etc.) in the scene and to work with them individually, avoiding neighboring interference of other objects, noise, spurious artifacts, or background.

Mathematical morphology is a known set of mathematical tools used in digital image processing area to perform linear transformations on the shapes of images's regions. There are two basic morphological operations: *erosion* and *dilation*. They will be defined next to make this text self-contained.

Let us take the image X and a template B , the *structuring element*. They will be represented as sets in two-dimensional Euclidean space. Let B_x denote the translation of B so that its origin is located at x . Then the erosion of X by B is defined as the set of all points x such that B_x is included in X , that is,

$$\text{erosion} : X \ominus B = \{x : B_x \subset X\}. \quad (7.23)$$

Similarly, the dilation of X by B is defined as the set of all points x such that B_x hits X , that is, they have a nonempty intersection:

$$\text{dilation} : X \oplus B = \{x : B_x \cap X \neq \emptyset\}. \quad (7.24)$$

These two operations are the base of all more complex transformations in mathematical morphology. For example, we can use an *opening* which consists of an erosion followed by a dilation of the result. This operation allows one to disconnect two different regions for treating them separately. The dual of opening is the close operation, which consists of an erosion over the dilation's result. The effect of closing an image is rightly the opposite of opening: It connects weak separated regions (see [40] for a review of other useful operations).

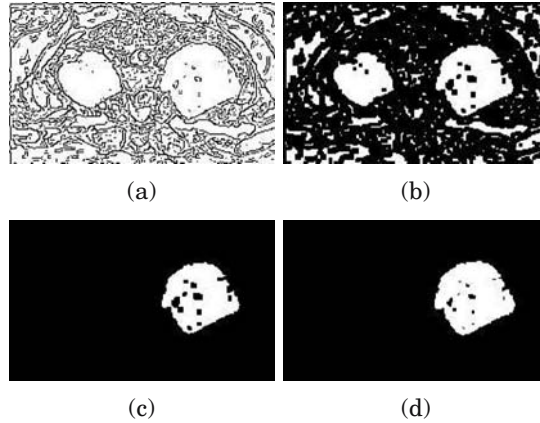


Figure 7.5: (a) Edge map after using the canny algorithm in the image of Fig. 7.3. (b) Erosion result over the Canny algorithm output. (c) Isolated region of interest. (d) Final result after dilation.

In this section, we are interested in applying *morphological chains* (sequence of a morphological operations) techniques to isolate specific regions in medical images. These extracted regions will be used for initializing deformable models.

We begin with a grayscale image such as in Fig. 7.3. Firstly, an edge detection filter is applied. The Canny edge detector was used [9], despite the fact that there are many other possibilities [13, 35, 40]. Figure 7.5(a) gives the result of applying the Canny methodology over the image in Fig. 7.3.

In Fig. 7.5(a) note that the two white predominant regions at the center of the image are the two lungs, which are the regions of interest. For convenience, this image was inverted with regard to its black–white pixels before initializing the morphological process. In this case, when applying the erosion operation (Eq. (7.23)) over the image in Fig. 7.5(a), we eliminate artifacts, weak edges, and separate weak connected regions. The net effect is to attenuate or eliminate high-frequency components. In the example of Fig. 7.5(a), we used a cross-structuring element. The result can be seen in Fig. 7.5(b).

Now, the two bigger regions are detached from the other ones, and we can separate and treat them individually. Figure 7.5(c) shows this result.

To restore the original size of the lung, we can apply the dilation operation (Eq. (7.24)). The result can be seen in Fig. 7.5(d).

Finally, an algorithm for isoline extraction gives the polygonal curve pictured in Fig. 7.5(d). This curve is an approximation of the desired boundary. It can be used as the initial curve for a deformable model.

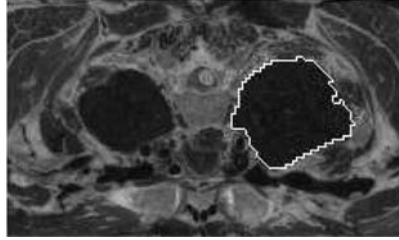


Figure 7.6: Original image with the outlined initial contour.

The obtained contour was plotted over the original image for matching (Fig. 7.6). If compared with Fig. 7.4(b) we observe an improvement in the obtained initialization.

7.3.3 Neural Nets

Neural networks have been used for instantiating deformable models for face detection [54] and handwritten digit recognition tasks [74] (see also [14] and references therein). To the best of our knowledge, there are no references using neural nets to initialize deformable models for medical images. However, the network system proposed in [25], which segments MR images of the thorax, may be closer to this proposal.

In this method each slice is a gray-level image composed of (256×256) pixels values and is accompanied by a corresponding (target) image containing just the outline of the region. Target images were obtained using a semiautomatic technique based on a region growing algorithm. The general idea is to use a multilayer perceptron (MLP), where each pixel of each slice is classified into a contour-boundary and non-contour-boundary one.

The inputs to the MLP are intensity values of pixels from a (7×7) window centered on the pixel to be classified. This window size was found to be the smallest that enabled the contour boundary to be distinguished from the other image's artifacts. The output is a single node trained to have an activation of 1.0 for an input window centered in the pixel of a contour boundary, and 0.0 otherwise. The network has a single hidden layer of 30 nodes.

The network was trained using error backpropagation [12, 55] with weight elimination [72] to improve the network's generalization ability. The training data should be constructed interactively: A proportion of misclassified examples should be added to the training set and used for retraining. The process

is initiated from small random selection of contour-boundary and non-contour-boundary examples and should be terminated when a reasonable classification (on a given slice) is achieved.

The MLP classified each pixel independently of the others, and therefore has no notion of a closed contour. Consequently, the contour boundaries it produces are often fragmented and noisy (false negatives and false positives, respectively). Then, with this initial set of points classified as contour boundaries, a deformable model is used to link the boundary segments together, while attempting to ignore noise.

In [25] the *elastic net* algorithm is used. This technique is based on the following equations:

$$\Delta u_{j,l}^{t+1} = \alpha \sum_{i=1}^N G_{ij}(p_{i,l} - u_{j,l}^t) + K\beta(u_{j+1,l}^t - 2u_{j,l}^t + u_{j-1,l}^t), \quad (7.25)$$

$$\Delta u_{j,l}^{t+1} = K\gamma(u_{j,l+1}^{t+1} - 2u_{j,l}^{t+1} + u_{j,l-1}^{t+1}), \quad (7.26)$$

where $\Delta u_{j,l}^{t+1}$ is an interslice smoothing force, K is a simulated annealing term, α , β , γ are predefined parameters, and G_{ij} is a normalized Gaussian that weights the action of the force that acts over the net point $u_{j,l}$ due to edge point $p_{i,l}$ (l is the slice index).

The deformable model initialization is performed by using a large circle encompassing the lung boundary in each slice. This process can be improved by using the training set.

As an example, let us consider the work [74] in handwritten digit recognition. In this reference, each digit is modeled by a cubic B-spline whose shape is determined by the positions of the control points in the object-based frame. The models have eight control points, except for the one model which has three, and the model for the number seven which has five control points. A model is transformed from the object-based frame to the image-based frame by an affine transformation which allows translation, rotation, dilation, elongation, and shearing. The model initialization is done by determining the corresponding parameters. Next, model deformations will be produced by perturbing the control points away from their initial locations.

There are ten classes of handwritten digits. A feedforward neural network is trained to predict the position of the control points in a normalized 16×16 gray-level image. The network uses a standard three-layer architecture. The outputs are the location of the control points in the normalized image. By inverting the

normalization process, the positions of the control points in the unnormalized image are determined. The affine transformation corresponding to these image can then be determined by running a special search procedure.

7.4 Initialization of T-Surfaces

All the methods described in Section 7.3 suffer from a common limitation: Self-intersections may happen during the evolution of the initial curve/surface.

Traditional deformable models [6, 19, 42], including the one defined by Eq. (7.9), cannot efficiently deal with self-intersections. It is due to the non-local testes dependency, which requires $O(N^2)$ in the worst case, where N is the number of mesh nodes (or snaxels, for 2D).

Recently, in [63] we have shown that such limitation can be addressed by using the T-snakes model because the reparameterization process of this model can naturally deal with self-intersections. It can also be addressed for 3D by using the T-surfaces.

Firstly, let us make some considerations about the T-snakes/T-surfaces.

The threshold T used in the normal force definition (7.12) plays an important role in the T-surfaces model [47, 49]. If not chosen properly, the T-surfaces can be frozen in a region far from the target(s) [33, 63].

The choice of T is more critical when two objects to be segmented are too close, as shown in Fig. 7.7. In this example, the marked grid nodes are those whose image intensity falls bellow the threshold T .

For T-snakes model to accurately segment the pictured objects, it has to burn the marked grid nodes. However, the normal force given by expression (7.12) changes its signal if the T-snakes gets closer. So, the force parameters

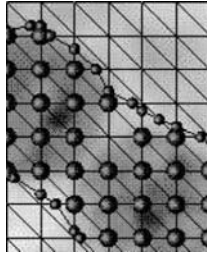


Figure 7.7: T-snake and grid nodes marked.

in expressions (7.11) and (7.12) have to be properly chosen to guarantee the advance over narrow regions. However, parameters choice remains an open problem in snake models [31]. This problem can be addressed by increasing the grid resolution as it controls the flexibility of T-surfaces. However, this increases the computational cost of the method.

To address the trade-off between model flexibility and the computational cost, in [22, 29] we propose to get a rough approximation of the target surfaces by isosurfaces generation methods. Then T-surfaces model is applied.

The topological capabilities of T-surfaces enable one to efficiently evolve the isosurfaces extracted. Thus, we combine the advantages of a closer initialization, through isosurfaces, and the advantages of using a topologically adaptable deformable model. These are the key ideas of our previous works [22, 29]. We give some details of them.

At first, a *local scale property* for the targets was supposed: Given an object O and a point $p \in O$, let r_p be the radius of a hyperball B_p which contains p and lies entirely inside the object. We assume that $r_p > 1$ for all $p \in O$. Hence, the minimum of these radii ($r_{\min}$) is selected.

Thus, we can use $r_{\min}$ to reduce the resolution of the image without losing the objects of interest. This idea is pictured in Fig. 7.8. In this simple example, we have a threshold which identifies the object ($T < 150$), and a CF triangulation whose grid resolution is 10×10 .

Now, we can define a simple function, called an *object characteristic function*, as follows:

$$\begin{aligned} \chi(p) &= 1, & \text{if } I(p) < T, \\ \chi(p) &= 0, & \text{otherwise,} \end{aligned} \quad (7.27)$$

where p is a node of the triangulation (marked grid nodes on Fig. 7.8(a)).

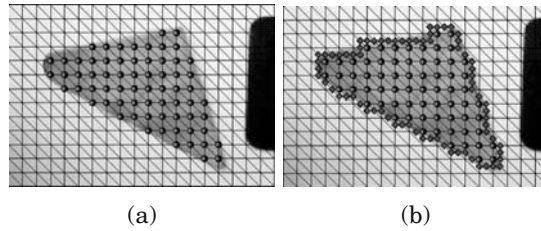


Figure 7.8: (a) Original image and characteristic function. (b) Boundary approximation.

We can do a step further, shown in Fig. 7.8(b), where we present a curve which belongs to the transverse triangles. Observe that this curve approximates the boundary we seek. This curve (or surface for 3D) can be obtained by isosurface extraction methods and can be used to efficiently initialize the T-surfaces model, as we already pointed out before.

If we take a grid resolution coarser than $r_{\min}$, the isosurface method might split the objects. Also, in [22, 29] it is supposed that the object boundaries are closed and connected. These topological restrictions imply that we do not need to search inside a generated connected component.

In [63] we discard the mentioned scale and topological constraints. As a consequence, the target topology may be corrupted. So, a careful approach will be required to deal with topological defects. An important point is the choice of the method to be used for isosurfaces generation. In [22, 63] we consider two kinds of isosurface generation methods: the marching ones and continuation ones.

In marching cubes, each surface-finding phase visits all cells of the volume, normally by varying coordinate values in a triple “for” loop [45]. As each cell that intersects the isosurface is found, the necessary polygon(s) to represent the portion of the isosurface within the cell is generated. There is no attempt to trace the surface into neighboring cells. Space subdivision schemes (such as Octree and k-d-tree) have been used to avoid the computational cost of visiting cells that the surface does not cut [17, 64].

Once the T-surfaces grid is a CF one, the tetra-cubes is especially interesting for this discussion [10]. As in the marching cubes, its search is linear: Each cell of the volume is visited and its simplices (tetrahedrons) are searched to find surfaces patches. Following marching cubes implementations, tetra-cubes uses auxiliary structures based on the fact that the topology of the intersections between a plane and a tetrahedron can be reduced to three basic configurations pictured in Fig. 7.1 (Section 7.2.3).

Unlike tetra-cubes, continuation algorithms attempt to trace the surface into neighboring simplices [1]. Thus, given a transverse simplex, the algorithm searches its neighbors to continue the surface reconstruction. The key idea is to generate the combinatorial manifold (set of transverse simplices) that holds the isosurface.

The following definition will be useful. Let us suppose two simplices σ_0, σ_1 , which have a common face and the vertices $v \in \sigma_0$ and $v' \in \sigma_1$ both opposite

the common face. The process of obtaining v' from v is called *pivoting*. Let us present the basic continuation algorithm [1].

PL generation algorithm:

```

Find a transverse triangle  $\sigma_0$ ;
 $\Sigma = \{\sigma_0\}$ ;  $V(\sigma_0) = \text{set of vertices of } \sigma_0$ ;
while  $V(\sigma) \neq \emptyset$  for some  $\sigma \in \Sigma$ 
.   get  $\sigma \in \Sigma$  such that  $V(\sigma) \neq \emptyset$ ;
.   get  $v \in V(\sigma)$ ;
.   obtain  $\sigma'$  from  $\sigma$  by pivoting  $v$  into  $v'$ 
.   if  $\sigma'$  is not transverse
.       then drop  $v$  from  $V(\sigma)$ ;
.   else
.       if  $\sigma' \in \Sigma$  then
.           drop  $v$  from  $V(\sigma)$ ,  $v'$  from  $V(\sigma')$ 
.       else
.            $\Sigma \leftarrow \Sigma + \sigma'$ ;
.            $V(\sigma') \leftarrow \text{set of vertices of } \sigma'$ ;
.           drop  $v$  from  $V(\sigma)$ ,  $v'$  from  $V(\sigma')$ 

```

Differently from tetra-cubes, once the generation of a component is started, the algorithm runs until it is completed. However, the algorithm needs a set of seed simplices to be able to generate all the components of an isosurface. This is an important point when comparing continuation and marching methods.

If we do not have guesses about seeds, every simplex should be visited. Thus, the computational complexity of both methods is the same ($O(N)$ where N is the number of simplices).

However, if we know in advance that the target boundary is connected, we do not need to search inside a connected component. Consequently, the computational cost is reduced if continuation methods are applied.

Based on this discussion about marching cubes and PL generation, we can conclude that, if we do not have the topological and scale restrictions given in Section 7.4, tetra-cubes is more appropriate to initialize the T-surfaces. In this case, it is not worthwhile to attempt to reconstruct the surface into neighboring simplices because all simplices should be visited to find surface patches.

However, for the T-surfaces reparameterization (steps (1)–(4) in Section 7.2.3), the situation is different. Now, each connected component is

evolved at a time. Thus a method which generates only the connected component being evolved—that is, the PL generation algorithm—is interesting.

7.5 Reconstruction Method

Following the above discussion, we proposed in [22, 63] a segmentation/surface reconstruction method that is based on the following steps: (1) extract region-based statistics; (2) coarser image resolution; (3) define the *object characteristic function*; (4) PL manifold extraction by the tetra-cubes; (5) if needed, increase the resolution, return to step (3); and (6) apply T-surfaces model.

It is important to highlight that T-surfaces model can deal naturally with the self-intersections that may happen during the evolution of the surfaces obtained by step (4). This is an important advantage of T-surfaces.

Among the surfaces extracted in step (4), there may be open surfaces which start and end in the image frontiers and small surfaces corresponding to artifacts or noise in the background. The former is discarded by a simple automatic inspection. To discard the latter, we need a set of predefined features (volume, surface area, etc.) and corresponding lower bounds. For instance, we can set the volume lower bound as $8(r)^3$, where r is the dimension of the grid cells.

Besides, some polygonal surfaces may contain more than one object of interest (see Fig. 7.9). Now, we can use upper bounds for the features. These upper bounds are application dependent (anatomical elements can be used).

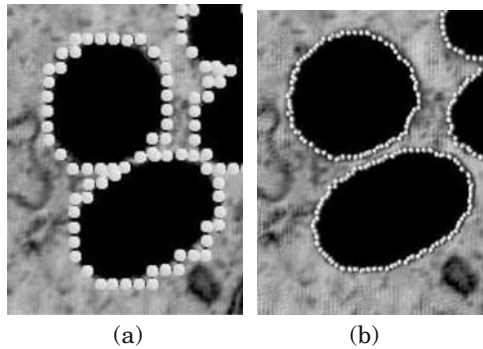


Figure 7.9: (a) PL manifolds for resolution 3×3 . (b) Result with the highest (image) resolution.

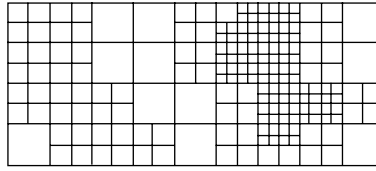


Figure 7.10: Representation of the multiresolution scheme.

The surfaces whose interior have volumes larger than the upper bound will be processed in a finer resolution. By doing this, we adopted the basic philosophy of some nonparametric multiresolution methods used in image segmentation based on pyramid and quadtree approaches [3, 8, 41]. The basic idea of these approaches is that as the resolution is decreasing, small background artifacts become less significant relative to the object(s) of interest. So, it can be easier to *detect* the objects in the lowest level and then propagate them back down the structure. In this process, it is possible to *delineate* the boundaries in a coarser resolution (step (4)) and to re-estimate them after increasing the resolution in step (5).

It is important to stress that the upper bound(s) is not an essential point for the method. Its role is only to avoid expending time computation in regions where the boundaries enclose only one object.

When the grid resolution of T-surfaces is increased, we just reparameterize the model over the finer grid and evolve the corresponding T-surfaces.

For uniform meshes, such as the one in Fig. 7.10, this multiresolution scheme can be implemented through *adaptive mesh refinement* data structures [5]. In these structures each node in the refinement level l splits into η^n nodes in level $l + 1$, where η is the refinement factor and n is the space dimension ($\eta = 2$ and $n = 3$ in our case). Such a scheme has also been explored in the context of *level sets* methods [61].

As an example, let us consider Fig. 7.9. In this image, the *outer* scale corresponding to the separation between the objects is finer than the object scales. Hence, the coarsest resolution could not *separate* all the objects. This happens for the bottom-left cells in Fig. 7.9(a). To correct this result, we increase the resolution only inside the extracted region to account for more details (Figure 7.9(b)).

We shall observe that T-surfaces makes use of only the data information along the surface when evolving the model toward the object boundary. Thus, we can

save memory space by reading to main memory only smaller chunks of the data set, instead of the whole volume, as is usually done by the implementations of deformable surface models. Such point is inside the context of out-of-core methods which are discussed next.

7.6 Out-of-Core for Improving Memory Utilization

There are few references of out-of-core approaches for segmentation purposes. The site (graphics.cs.ucdavis.edu/research/Slicer.html) describes a technique based on reordering the data according to a three-dimensional Lebesgue-space-filling-curve scheme to speed up data traversal in disk. The visualization toolkit uses cached, streaming (pulling regions of data in a continual flow through a pipeline) to transparently deal with large data sets [60]. Finally, and more important for our work, out-of-core isosurface extraction techniques have been implemented [16, 64] and can be used for segmentation purposes.

From the viewpoint of out-of-core isosurface generation, we need to efficiently perform the following operations: (a) group spatial data into clusters; (b) compute and store in disk cluster information (pointer to the corresponding block recorded in disk, etc.); and (c) optimize swap from disk to main memory. These operations require the utilization of efficient data structures. Experimental tests show that the branch-on-need octree (BONO) [64] and the meta-cell [16] framework provide efficient structures for out-of-core isosurface extraction. Next, we summarize and compare these methods.

Octrees are hierarchical tree structures of degree 8. If the volume's resolution is the same power of 2 in each direction; e.g., $2^d \times 2^d \times 2^d$, octrees offer the best ratio of the number of nodes to data points 1/7 [73]. Otherwise, an alternative, to be close to the optimum, is the branch-on-need octree (BONO) strategy [73]. Essentially, the octree is regarded as conceptually full, but the algorithm avoids allocating space for empty subtrees. With each node is associated a conceptual region and an actual region, as illustrated in Fig. 7.11. Besides, at each node the octree contains the maximum and minimum data values found in that node's subtree.

We shall observe that the same space partition could be obtained if we take the following procedure: Sort all data points by the x -values and partition them

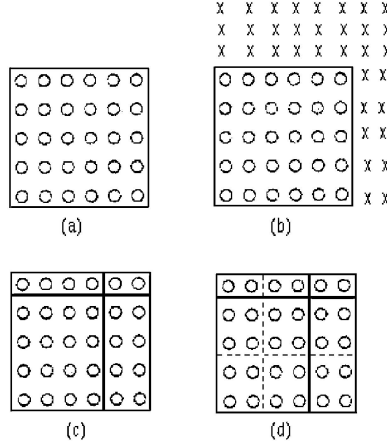


Figure 7.11: (a) Data set; (b) conceptual region; (c) level 1; and (d) final level.

into H consecutive chunks ($H = 3$ in Fig. 7.11). Then, for each such chunk, sort its data points by the y -values and partition them into H consecutive chunks. For 3D images we must repeat the procedure for the z -values.

That is precisely the *meta-cell* partition. Unlike octrees, meta-cell is not a hierarchical structure. The partition is defined through the parameter H . Besides, given a point (q_1, q_2, q_3) , inside the domain, the corresponding meta-cell is given by:

$$mcell = \lfloor q_i / C_i \rfloor, \quad i = 1, 2, 3, \quad (7.28)$$

where C_i is the number of data points of each chunk of the conceptual region, in the direction i . To each meta-cell is associated a set of meta-intervals (connected components among the intervals of the cells in that meta-cell). These meta-intervals are used to construct an interval tree, which will be used to optimize I/O operations. Given a set of N meta-intervals, let $e_1, e_2, \dots, e_{2n}$ be the sorted list of left and right endpoints of these intervals. Then, the interval tree is recursively defined as follows:

Interval tree construction: (i) If there is only one interval, then the current node r is a leaf containing that interval; (ii) else, the value $m = (e_n + e_{n+1})/2$ is stored in r as a key; the intervals that contain m are assigned to r as well as pointers to the subtrees $left(r)$ and $right(r)$. Go to step (i).

Now, let us take an overview of out-of-core isosurface extraction methods based on the above structures. The methodology presented in [64] extends the BONO for time-varying isosurface extraction. The proposed structure (*temporal*

branch-on-need (T-BON) octree) minimizes the impact of the I/O bottleneck by reading from disk only those portions of the search structure and data necessary to construct the current isosurface. The method works as follows.

A preprocessing step builds a BONO for each time step and properly stores it to disk. To avoid I/O performance problems at run-time, the algorithm packs nodes into disk blocks in order to read a number of nodes at once.

At run-time, the tree infrastructure is read from disk and recreated in memory. Isovalues queries are then accepted in the form $(timestep, isovalue)$. The algorithm initially fetches the root node of the octree corresponding to *timestep* from disk. If the extreme values are stored in the root node span *isovalue*, the algorithm next fetches all children of the root node from disk. This process repeats recursively until reaching the leaf nodes. Then, the algorithm computes disk blocks containing data points needed by that leaf and inserts those blocks into a list. Once all nodes required to construct the current isosurface have been brought into memory, the algorithm traverses the block list and reads the required data blocks sequentially from disk.

The meta-cell technique proposed by Chiang *et al.* [16] works through a similar philosophy. Given an *isovalue*, the query pipeline follows the next steps: (1) query the interval tree to find all meta-cells whose meta-intervals contain the *isovalue* (*active meta-cells*); (b) sort the reported meta-cell IDs properly to allow sequential disk reads; and (c) for active meta-cell, read it from disk to main memory and compute the corresponding isosurface patches.

An important difference between the meta-cell technique and T-BON is that, unlike T-BON, meta-cell uses two distinct structures: one for the scalar field information (interval tree) and another for the space partition. The link between these structures is given by the interval tree leaves information (meta-intervals and pointers to corresponding meta-cells). Such split in the way meta-cell technique deals with domain partition and the scalar field gives more flexibility to meta-cell if compared with T-BON.

For instance, the query “given a point (x, y, z) , find its image intensity,” useful when segmenting with deformable models, is implemented more easily through meta-cell (see expression (7.28)) than with BONO. Besides, image data sets are represented on regular grids which means that we do not need hierarchical structures to take account for regions with higher density of points. These are the reasons why meta-cell is more suitable for out-of-core image segmentation than BONO. Next, we will explore this fact.

7.7 Out-of-Core Segmentation Approach

In this section we present the out-of-core version of the segmentation framework described in Section 7.5.

That algorithm is interesting for this work because of two aspects. First, it uses the T-surfaces model which uses auxiliary and very memory consuming data structures (hash table to keep transverse simplices, T-surfaces mesh, etc.). Thus, a suitable out-of-core implementation would improve algorithm performance as well as make it possible to segment the data sets which would not fit in memory. Second, it needs both the queries found in segmentation algorithms: (a) Given a reference value q , find all image points p such that $I(p) = q$ and (b) given a point p , find the image intensity $I(p)$.

The meta-cell technique used has the following elements.

Meta-cell partition: The meta-cell size is application dependent. Basically, it depends on the data set size, disk block size, and the amount of memory available. For isosurface extraction, we can obtain a quantitative bound by following [16] and taking the dimensional argument that an active meta-cell with C cells has, most of times, $C^{2/3}$ active cells (or voxels). Therefore, we read $C^{1/3}$ layers of cells for each layer of isosurface. Thus, if the isosurface cuts K cells and if B is the number of cells fitting in one disk block, we expect to read $C^{1/3} \cdot (K/B)$ disk blocks to complete the isosurface. Henceforth, we can increase meta-cells sizes while keeping the effect of the factor $C^{1/3}$ negligible.

Interval tree: Volumetric images have some particular features that must be considered. Intensity values range from 0 to 255 and the data set is represented by a regular grid. This information can be used to find an upper bound for the interval tree size. Let us consider the worst case, for which the meta intervals are of the form: $I_0 = [0, 0]$; $I_1 = [2, 2]$; $\dots$; $I_{127} = [254, 254]$. Thus, in the worst case, we have 128 meta-intervals for each meta-cell. Each meta-interval uses two bytes in memory. For a $2^9 \times 2^9 \times 2^9$ data set, if we take meta-cells with $2^4 \times 2^4 \times 2^4$ data points, we find $2^{15} = 32$ kB meta-cells. Thus, we will need an amount of $2 \times 128 \times 32$ kB = 8.0 MB, which is not restrictive for usual workstations. Besides, in general, interval tree sizes are much smaller than this bound (see Section 7.9). Thus, we do not pack tree nodes as in [16].

Data cache: To avoid memory swap, we must control the memory allocation at run-time. This can be done through a data cache, which can store a predefined

number M of meta-cells. When the cache fills, the least recently used (LRU) meta-cell will be replaced [64].

Query algorithm: (a) Given an isovalue q , find all meta-intervals (and the corresponding meta-cell IDs) containing q , by querying the I/O interval tree defined in Section 7.6. (b) Given a point $q = (q_1, q_2, q_3)$, find the corresponding meta-cell ID through the expression (7.28).

Besides, we need some auxiliary structures. The characteristic function (χ) is a zero field at the beginning. There is a *processing_list* which is dynamically constructed through a procedure called *insert_neighbors*:

insert_neighbors(p): For each neighbor q of a node element p , verify if q has not been evolved by Eq. (7.14) and if $q \notin \text{processing\_list}$. In this case, insert q in *processing_list*.

The key idea behind the *processing_list* construction is to update node elements according to a *breadth-first-search* (BFS) *algorithm*; that is, we consider neighbors of a node as predecessors in the search. With such a procedure, we can minimize I/O operations due to the following property: starting at a seed, the algorithm visits all the neighbors; then it visits all the neighbors of neighbors, etc. until it runs out of neighbors to visit (see Fig. 7.12).

Thus, the *least recently used* meta-cell must be replaced when data cache fills because most probably the portion of T-surfaces that intersects that meta-cell has been completely updated. Certainly, we can generate the isosurfaces in step (2) according to a breadth-first-search continuation algorithm. However, we chose to incorporate this procedure in the T-surfaces method to get more generality for the out-of-core segmentation method.

Next, we outline the algorithm. We call *seed* a node element for which neighbors belong to the same meta-cell. Also, we suppose that the object of interest has intensity pattern inside the range $[I_1, I_2]$.

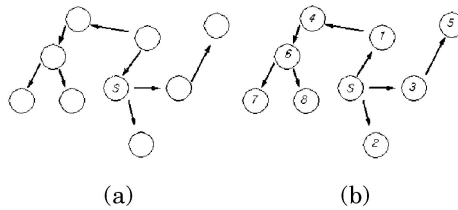


Figure 7.12: (a) Example of BFS algorithm in graphs. (b) Possible order of visiting nodes after BFS with seed S .

Out-of-Core Segmentation Algorithm:

```

(1) Compute Object_Characteristic_Function
.Traverse interval tree to find the list  $L$  of active meta-cells;
.While  $L$  is not NULL
.  Read  $M$  active meta-cells to main memory.
.  Take a metacell. Given a grid node  $p \in \text{metacell}$ :
if  $I(p) \in [I_1, I_2]$  then  $\chi(p) = 1$ 
(2) Extract isosurfaces.
(3) If needed, increase grid resolution. Go to step (1)
(4) Find a seed and insert it into processing_list
(5) Begin T-Surfaces model;
.While the processing_list is not empty:
.  Pop a point  $p$  from processing_list
.  Find the corresponding meta-cell( $p$ )
.  If meta-cell( $p$ ) is not in memory, read it
.  Find  $I(p)$  and  $\nabla I(p)$ 
.  Update  $p$  according to Eq. (7.14)
.  Call insert_neighbors(p)
.Update function  $\chi$ 
.Reparameterization of T-Surfaces (Section 7.2.3)
.If the termination condition is not reached, go to (4).

```

We shall observe that when the grid resolution of T-surfaces is (locally) increased in step (3), the list L of active meta-cells remains unchanged and the procedure to define the Object_Characteristic_Function does not change. Also, we must observe that the isosurfaces are taken over the object characteristic function field. Thus, there are no I/O operations in step (2).

7.8 Convergence of Deformable Models and Diffusion Methods

Despite the capabilities of the segmentation approach in Section 7.5, the projection of T-surfaces can lower the precision of the final result. Following [49], when T-surfaces stops, we can discard the grid and evolve the model without it avoiding errors due to the projections.

However, for noisy images the convergence of deformable models to the boundaries is poor due to the nonconvexity of the image energy. This problem can be addressed through diffusion techniques [18, 44, 52].

In image processing, the utilization of diffusion schemes is a common practice. Gaussian blurring is the most widely known. Other approaches are the anisotropic diffusion [52] and the gradient vector flow [77].

From the viewpoint of deformable models, these methods can be used to improve the convergence to the desired boundary. In the following, we summarize these methods and conjecture their unification.

Anisotropic diffusion is defined by the following general equation:

$$\frac{\partial I(x, y, t)}{\partial t} = \operatorname{div}(c(x, y, t) \nabla I), \quad (7.29)$$

where I is a gray-level image [52].

In this method, the blurring on parts with high gradient can be made much smaller than in the rest of the image. To show this property, we follow Perona *et al.* [52]. Firstly, we suppose that the edge points are oriented in the x direction. Thus, Eq. (7.29) becomes:

$$\frac{\partial I(x, y, t)}{\partial t} = \frac{\partial}{\partial x}(c(x, y, t) I_x(x, y, t)). \quad (7.30)$$

If c is a function of the image gradient: $c(x, y, t) = g(I_x(x, y, t))$, we can define $\phi(I_x) \equiv g(I_x) \cdot I_x$ and then rewrite Eq. (7.29) as:

$$I_t = \frac{\partial I}{\partial t} = \frac{\partial}{\partial x}(\phi(I_x)) = \phi'(I_x) \cdot I_{xx}. \quad (7.31)$$

We are interested in the time variation of the slope: $\frac{\partial I_x}{\partial t}$. If $c(x, y, t) > 0$ we can change the order of differentiation and with a simple algebra demonstrate that:

$$\frac{\partial I_x}{\partial t} = \frac{\partial I_t}{\partial x} = \phi'' \cdot I_{xx}^2 + \phi' \cdot I_{xxx}.$$

At edge points we have $I_{xx} = 0$ and $I_{xxx} \ll 0$ as these points are local maxima of the image gradient intensity. Thus, there is a neighborhood of the edge point in which the derivative $\partial I_x / \partial t$ has sign opposite to $\phi'(I_x)$. If $\phi'(I_x) > 0$ the slope of the edge point decrease in time. Otherwise it increases, that means, border becomes sharper. So, the diffusion scheme given by Eq. (7.29) allows to blur small discontinuities and to enhance the stronger ones. In this work, we have

used ϕ as follows:

$$\phi = \left(\frac{\nabla I}{(1 + [\|\nabla I\|/K]^2)} \right), \quad (7.32)$$

as shall see next.

In the above scheme, I is a scalar field. For vector fields, a useful diffusion scheme is the gradient vector flow (GVF). It was introduced in [77] and can be defined through the following equation [78]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla \cdot (g \nabla u) + h(u - \nabla f), \\ u(x, 0) &= \nabla f \end{aligned} \quad (7.33)$$

where f is a function of the image gradient (for example, P in Eq. (7.13)), and $g(x)$, $h(x)$ are non-negative functions defined on the image domain.

The field obtained by solving the above equation is a smooth version of the original one which tends to be extended very far away from the object boundaries. When used as an external force for deformable models, it makes the methods less sensitive to initialization [77] and improves their convergence to the object boundaries.

As the result of steps (1)–(6) in Section 7.5 is in general close to the target, we could apply this method to push the model toward the boundary when the grid is turned off. However, for noisy images, some kind of diffusion (smoothing) must be used before applying GVF. Gaussian diffusion has been used [77] but precision may be lost due to the nonselective blurring [52].

The anisotropic diffusion scheme presented above is an alternative smoothing method that can be used. Such observation points forward the possibility of integrating anisotropic diffusion and the GVF in a unified framework. A straightforward way of doing this is allowing g and h to be dependent upon the vector field u . The key idea would be to combine the selective smoothing of anisotropic diffusion with the diffusion of the initial field obtained by GVF. Besides, we expect to get a more stable numerical scheme for noisy images.

Diffusion methods can be extended for color images. In [56, 57] such a theory is developed. In what follows we summarize some results in this subject.

Firstly, the definition of edges for multivalued images is presented [57]. Let $\Phi(u_1, u_2, u_3) : D \subset \mathbb{R}^3 \rightarrow \mathbb{R}^m$ be a multivalued image. The difference of image values at two points $P = (u_1, u_2, u_3)$ and $Q = (u_1 + du_1, u_2 + du_2, u_3 + du_3)$ is

given by $d\Phi$:

$$d\Phi = \sum_{i=1}^{i=3} \frac{\partial \Phi}{\partial u^i} du^i \Rightarrow d\Phi^2 = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} \left\langle \frac{\partial \Phi}{\partial u^i}, \frac{\partial \Phi}{\partial u^j} \right\rangle du^i du^j, \quad (7.34)$$

where $d\Phi^2$ is the square Euclidean norm of $d\Phi$. The matrix composed of the coefficients $g_{ij} = \langle \frac{\partial \Phi}{\partial u^i}, \frac{\partial \Phi}{\partial u^j} \rangle$ is symmetric, and the extremes of the quadratic form $d\Phi^2$ are obtained in the directions of the eigenvectors (θ_+, θ_-) of the metric tensor $[g_{ij}]$, and the values attained there are the corresponding maximum/minimum eigenvalues (λ_+, λ_-) . Hence, a potential function can be defined as [57]:

$$f(\lambda_+, \lambda_-) = \lambda_+ - \lambda_-, \quad (7.35)$$

which recovers the usual edge definition for gray-level images: $(\lambda_+ = \|\nabla I\|^2, \lambda_- = 0 \text{ if } m = 1)$.

Similarly to the gray-level case, noise should be removed before the edge map computation. This can be done as follows [56, 57]. Given the directions $\theta_{\pm}$, we can derive the corresponding anisotropic diffusion by observing that diffusion occurs normal to the direction of maximal change θ_+ , which is given by θ_- . Thus, we obtain:

$$\frac{\partial \Phi}{\partial t} = \frac{\partial^2 \Phi}{\partial \theta_-}, \quad (7.36)$$

which means:

$$\frac{\partial \Phi_1}{\partial t} = \frac{\partial^2 \Phi_1}{\partial \theta_-}, \dots, \frac{\partial \Phi_m}{\partial t} = \frac{\partial^2 \Phi_m}{\partial \theta_-}. \quad (7.37)$$

In order to obtain control over local diffusion, a factor g_{color} is added:

$$\frac{\partial \Phi}{\partial t} = g_{\text{color}}(\lambda_+, \lambda_-) \frac{\partial^2 \Phi}{\partial \theta_-}, \quad (7.38)$$

where g_{color} can be a decreasing function of the difference $(\lambda_+ - \lambda_-)$.

This work does not separate the vector into its direction (chromaticity) and magnitude (brightness).

In [67], Tang *et al.* pointed out that, although good results have been reported, chromaticity is not always well preserved and color artifacts are frequently observed when using such a method. They proposed another diffusion scheme to address this problem. The method is based on separating the color image Φ into chromaticity and brightness, and then processing each one of these components

with proper diffusion flows. By doing this, the following multiscale representation is proposed for 2D images, which can be straightforwardly extended to 3D. Let $B : D \subset \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ and $C : D \subset \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow S^{m-1}$, the image brightness and chromaticity, respectively ((S^{m-1}) being the $(m-1)$ -dimensional unit sphere), such that:

$$B(u_1, u_2, 0) = \sqrt{\sum_{i=1}^m \Phi_i(u_1, u_2)}, \quad (7.39)$$

$$C(u_1, u_2, 0) = \frac{1}{B(u_1, u_2, 0)} \Phi(u_1, u_2), \quad (7.40)$$

and, at time t , the former will be given by the following anisotropic diffusion flow:

$$\frac{\partial B}{\partial t} = \frac{(B_{u_1 u_1} B_{u_2}^2 - 2B_{u_1} B_{u_2} B_{u_1 u_2} + B_{u_2 u_2} B_{u_1}^2)^{1/3}}{1 + \|\nabla B\|}, \quad (7.41)$$

which is motivated by the affine-invariant denoising method proposed in [51, 58]. The above flow can be interpreted by observing that the level sets of the brightness function have curvature K that can be written as (see expression (7.21) also):

$$K = \frac{B_{u_1 u_1} B_{u_2}^2 - 2B_{u_1} B_{u_2} B_{u_1 u_2} + B_{u_2 u_2} B_{u_1}^2}{\|\nabla B\|^3}. \quad (7.42)$$

Thus, the desired effect is to get an affine-invariant diffusion without smoothing the brightness field across edges (see [51, 57] for more details).

The chromaticity is the solution of the variational problem given by:

$$\min_{C : \mathbb{R}^2 \rightarrow S^{m-1}} \iint_D \|\nabla C\|^p du_1 du_2, \quad (7.43)$$

where $p \geq 1$ and $\|\nabla C\|$ is:

$$\|\nabla C\| = \left[\sum_{i=1}^m \left(\left(\frac{\partial C_i}{\partial u_1} \right)^2 + \left(\frac{\partial C_i}{\partial u_2} \right)^2 \right) \right]^{1/2}. \quad (7.44)$$

The scheme for the chromaticity comes from the theory of harmonic maps in liquid crystals [66]. The optimization problem can be solved by Euler-Lagrange equations or even in the content of weak solution. In [67] some results are reported for 2D images and open questions related to the mathematical formulation are presented.

7.9 Experimental Results

In this section we present a set of results obtained with the methods presented in Sections 7.5–7.8. The main application context is medical images.

7.9.1 Anisotropic Diffusion

Now, we demonstrated the utility of image diffusion methods in our work. We take a synthetic $150 \times 150 \times 150$ image volume composed of a sphere with a radius of 30 and an ellipsoid with axes 45, 60, and 30 inside a uniform noise specified by the image intensity range 0–150.

Figure 7.13 shows the result for steps (1)–(4) in Section 7.5, applied to this volume after Gaussian diffusion (Fig. 7.13(a)), and anisotropic diffusion

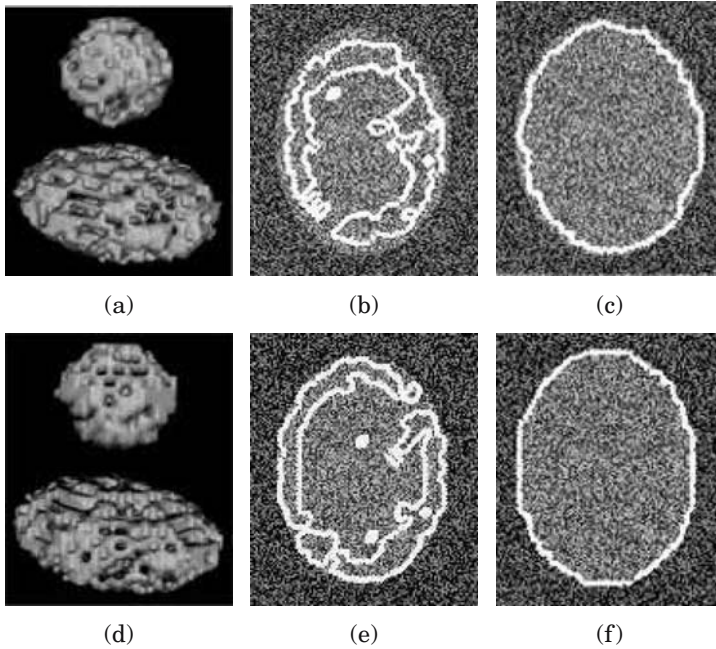


Figure 7.13: (a) Result for steps (1)–(4) with Gaussian diffusion. (b) Cross sections of (a) for slice 40. (c) Cross section of final solution for slice 40. (d) Result for steps (1)–(4) with anisotropic diffusion. (e) Cross sections of (d) for slice 40. (f) Cross section of final solution when using anisotropic diffusion (slice 40).

(Fig. 7.13(d)) defined by the equation:

$$\frac{\partial I}{\partial t} = \operatorname{div} \left(\frac{\nabla I}{(1 + [\|\nabla I\|/K]^2)} \right), \quad (7.45)$$

where the threshold K can be determined by a histogram of the gradient magnitude. It was set to $K = 300$ in this example. The number of interactions of the numerical scheme used [52] to solve this equation was 4.

Figures 7.13(b) and (e) show the cross section corresponding to the slice 40. We observe that with anisotropic diffusion (Fig. 7.13(e)), the result is closer to the boundary than with the Gaussian one (Fig. 7.13(b)).

Also, the final result is more precise when preprocessing with anisotropic diffusion (Fig. 7.13(f)). This is expected because, according to Section 7.8, Eq. (7.45) enables the blurring of small discontinuities (gradient magnitude below K) as well as enhancement of edges (gradient magnitude above K).

Another point becomes clear in this example: The topological abilities of T-surfaces enable the correction of the defects observed in the surface extracted through steps (1)–(4). We observed that, after few interactions, the method gives two closed components. Thus, the reconstruction becomes better.

The T-surface parameters used are: $c = 0.65$, $k = 1.32$, and $\gamma = 0.01$. The grid resolution is $5 \times 5 \times 5$, freezing point is set to 15, and threshold $T \in (120, 134)$ in Eq. (7.12). The number of deformation steps for T-surfaces was 17. The model evolution can be visualized in <http://virtual01.lncc.br/rodrigo/tese/ellipse.html>.

7.9.2 Artery Reconstruction

This section demonstrates the advantages of applying T-surfaces plus isosurface methods. Firstly, we segment an artery from an $80 \times 86 \times 72$ image volume obtained from the Visible Human project. This is an interesting example because the intensity pattern inside the artery is not homogeneous.

Figure 7.14(a) shows the result of steps (1)–(4) when using $T \in (28, 32)$ to define the object characteristic function (Eq. (7.27)). The extracted topology is too different from that of the target. However, when applying T-surfaces the obtained geometry is improved.

Figure 7.14(b) shows the result after the first step of evolution. The merges among components improve the result. After four interactions of the

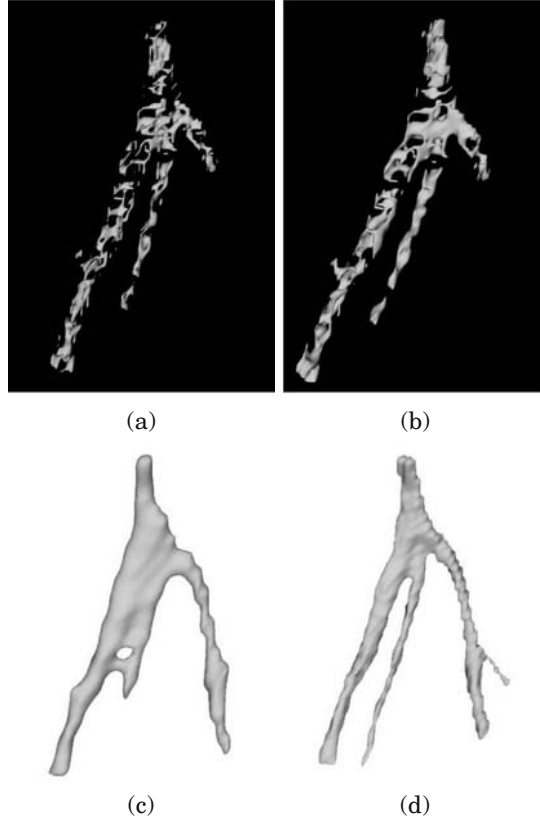


Figure 7.14: (a) Result of steps (1)–(4) with grid $3 \times 3 \times 3$. (b) T-surfaces evolution (step 1). (c) Solution for initial grid. (d) Final solution for grid $1 \times 1 \times 1$.

T-surfaces algorithm, the extracted geometry becomes closer to that of the target (Fig. 7.14(c)).

However, the topology remains different. The problem in this case is that the used grid resolution is too coarse if compared with the separation between branches of the structure. Thus, the flexibility of the model was not enough to correctly perform the surface reconstruction.

The solution is to increase the resolution and to take the partial result of Fig. 7.14(c) to initialize the model in the finer resolution. In this case, the correct result is obtained only with the finest grid ($1 \times 1 \times 1$). Figure 7.14(d) shows the desired result obtained after nine interactions. We also observe that new portions of the branches were reconstructed due to the increase of T-surfaces flexibility obtained through the finer grid. We should emphasize that an advantage of

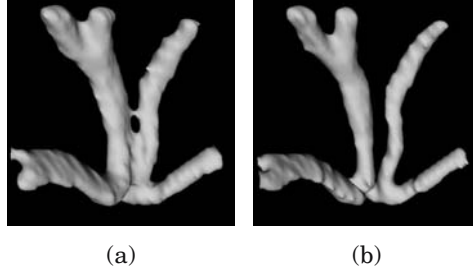


Figure 7.15: (a) Example showing an incorrect result. (b) Anisotropic diffusion in a preprocessing phase improving final result.

the multiresolution approach is that at the lower resolution, small background artifacts become less significant relative to the object(s) of interest. Besides, it avoids the computational cost of using a finer grid resolution to get closer to the target (see Section 7.4).

The T-surfaces parameters are $\gamma = 0.01$, $k = 1.222$, and $c = 0.750$. The total number of evolution is 13. The number of triangular elements is 10 104 for the highest resolution and the clock time was of the order of 3 min.

Sometimes, even the finest resolution may not be enough to get the correct result. Figure 7.15(a) pictures such an example.

In this case, we segment an artery from a $155 \times 170 \times 165$ image volume obtained from the visible human project. The T-surfaces parameters are: $c = 0.75$, $k = 1.12$, $\gamma = 0.3$, grid resolution is $4 \times 4 \times 4$, and freezing point is set to 10. The result of steps (1)–(6) is pictured in Fig. 7.15(a).

Among the proposals to address this problem (relax the threshold, mathematical morphology [59], etc.), we tested anisotropic diffusion [52]. The properties of this method (Section 7.8) enable smoothing within regions in preference to smoothing across boundaries. Figure 7.15(b) shows the correct result obtained when preprocessing the image with anisotropic diffusion and then applying steps (1)–(6).

7.9.3 Out-of-Core Segmentation

In this section, we attempt to experimentally demonstrate our out-of-core segmentation technique. We consider three gray-level data sets and a 3D color image (Table 7.1).

Table 7.1: Statistics for preprocessing: number of meta-cells (no. of MC), times for meta-cell generation (MC generation), gradient computation (gradient), interval tree construction (IT), size of each meta-cell (MC size), and size of the interval tree (IT size)

Data set	Artery	Artery2	Kidney	ColorV
Size (MB)	3.37	20.97	4.57	63.08
No. of MC	125	1000	7600	125
MC generation (sec)	3	25	5	58
Gradient (sec)	16	88	24	1740
IT (sec)	0.5	0.5	0.5	1.5
Total (sec)	20	114	30	1801
MC size (kB)	343.04	285.696	8.2944	2718.72
IT size (kB)	38.62	379.13	938.95	176.01

As we already said, T-surfaces uses auxiliary and very memory consuming data structures. We certainly can design optimizations. However, by now, we had to use a machine with enough memory to manage the T-surfaces structures. The machine used is Pentium III, 863 MHz with 512 MB of RAM and 768 MB of swap space.

There are three main steps to be considered: preprocessing, isosurfaces generation, and T-surfaces evolution. Preprocessing encompasses the gradient computation and meta-cell generation. Meta-cell generation is basically divided into two steps: (a) mapping data points into meta-cells and writing data information to the corresponding meta-cells; and (b) finding meta-intervals and computing the interval tree. As can be seen in Table 7.1, preprocessing step can be expensive due to the gradient computation. Also, we observe from this table that the interval tree size (last row) is very much smaller than the bound computed in Section 7.7 (8 MB).

Isosurfaces generation encompasses steps (1) and (2) of the algorithm in Section 7.7. Table 7.2 reports some performance statistics for this step. In this case, we use a data cache of 15 MB.

It is important to observe that, in general, the smaller the meta-cell size, the faster the isosurface search. This fact is verified in Table 7.2 in which we vary the number of meta-cells used for the kidney data set. For instance, when using 7600 meta-cells, the algorithm can fetch all the sets of active meta-cells from disk. Thus, there are no extra I/O operations during step (1) of the segmentation

Table 7.2: Statistics for isosurface generation on the kidney data set. This table reports the number of meta-cells (no. of MC), number of active meta-cells (activeMC), interval tree (IT) information, and total time for isosurface generation (IsoTime). The data cache size used is 15 MB

No. of MC	7600	1000	288	125
ActiveMC	1140	474	256	125
IT size (kB)	938.95	203.56	61.23	21.25
IT time (sec)	1	1	1	1
IsoTime (sec)	13	15	15	20

algorithm. Also, the meta-cell technique minimizes the effect of the I/O bottle-neck by reading from disk only those portions of the data necessary for step (1). Besides, the time for an interval tree query was approximately 1 sec (“IT time” in Table 7.2). As a consequence, if compared with the traditional implementation, we observe a performance improvement of 2 sec when using 7600 meta-cells.

The final step, the T-surfaces evolution, is globally reported in Table 7.3 for the kidney data set, maintaining the same partitions of Table 7.2. The quantity “no. of I/O” reported in this table counts the number of times that the algorithm reads a meta-cell from disk.

Again, the smaller the meta-cell size, the faster the whole process. Despite the high number of I/O operations reported in row 2 of Table 7.3, we must highlight that the total time for T-surfaces evolution without using the meta-cell was 623 sec, against 600 sec for the worst case reported in Table 7.3. For the best case, we observe a performance improvement of 120 sec, which is an important result. The final surface (Fig. 7.16(c)) has 34 624 triangular elements.

Table 7.3: T-surfaces in the kidney data set. This table reports the number of meta-cells (no. of MC), of number I/O operations (no. of I/O), number of meta-cells that have been cut (CutMC), and the total clock time for evolution (time). The data cache size is 15 MB and the number of interactions is 16

No. of MC	7600	1000	288	125
No. of I/O	1244	4780	1818	1458
CutMC	1074	325	125	70
Time (sec)	503	570	584	600

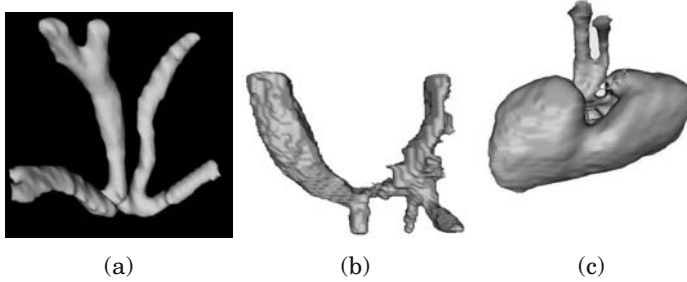


Figure 7.16: Extracted surfaces for: (a) artery data set; (b) artery2; and (c) kidney data set.

The number of I/O operations is a problem that we must address in future works. If we compare the “no. of I/O” with the number of meta-cells that the T-surfaces cuts during evolution (cutMC, in row 3), we observe that we should look for more efficient schemes for memory management.

The parameters used in the T-surfaces for the above experiments are: grid $4 \times 4 \times 4$, freezing point = 10, $\gamma = 0.01$, $k = 1.222$, $c = 0.750$. The intensity pattern of the targets is given by the following ranges: [10, 22] for data set, [195, 255] for kidney, and [15, 30] for artery2. Figure 7.16 shows the extracted surfaces.

The data set artery2 is a gray-level version of a volume obtained from the Visible Human project. The ColorV data set, mentioned in Table 7.1, is the same volume, but in its original color (RGB). We apply our method for this volume, just using one threshold for each color channel [7] and using the color edge definition given in Section 7.8.

The Visual Human project encompasses a huge color data set of human body. For 125 meta-cells, we found R, G, and B interval trees with 64.75 kB, 65.67 kB and 45.59 kB, respectively, given the total size of 176.01 kB reported in Table 7.1. The preprocessing time is much higher now (29 min) due to the number of operations required to compute the gradient.

7.10 Discussion and Perspectives

When considering automatic initialization for deformable models, some aspects must be taken into account. The target topology may be corrupted due to inhomogeneities of the image field or gradient discontinuities. Besides, the obtained

curves/surfaces are in general not smooth, presenting defects such as protrusions, concavities, or even holes (for surfaces) due to image irregularities.

These problems can be addressed through an efficient presegmentation. For instance, when reconstructing the geometry of the human cerebral cortex, Prince *et al.* [76] used a fuzzy segmentation method (*Adaptive Fuzzy C-Means*) to obtain the following elements: a segmented field which provides a fuzzy membership function for each tissue class; the mean intensity of each class; and the inhomogeneity of the image, modeled as a smoothly varying gain field (see [76] and references therein). The result can be used to steer the isosurface extraction process as well as the deformable model, which is initialized by the obtained isosurface. We have used a similar approach as described in [33].

Moreover, the image forces may not be strong enough to push the model toward the object boundary. Even the balloon model in Eq. (7.9) cannot deal with such a problem because it is difficult to predict if the target is inside or outside the isosurface (see Fig. 7.6). So, it makes harder to accurately define the normal force field. The GVF (Section 7.8) can be used to generate an image force field that improves the convergence of the model toward the boundary. GVF is sensitive to noise and artifacts but we can achieve good results for presegmented images [77, 78].

Now, we will compare our segmentation approach (Section 7.5) to that proposed in [47]. In that reference, a set of small spherical T-snakes is uniformly distributed over the image. These curves progressively expand/merge to recover the geometry of interest. The same can be done for 3D.

Our approach can be considered an improvement of that one described in [47]. Our basic argument is that we should use the threshold to get *seeds* closer to the objects of interest. Thus, we avoid expanding time evolving surfaces far from the target geometry. Besides, we have observed an improvement in the performance of the segmentation process if compared with the traditional initialization of T-surfaces (an implicit defined surface inside the object) [49].

Our method is adaptive in the sense that we can increase the T-surfaces grid resolution where it is necessary. As the T-surfaces grid controls the density of the polygonal surfaces obtained, the number of triangular elements gets larger inside these regions. That increase in density is not due to boundary details but because the outer scale corresponding to the separation between the objects is too fine (as in Fig. 7.9). This is a disadvantage of our approach.

Such a problem would be avoided if we could define significant areas along the surfaces and then apply the refinement only in the regions around them. However, it is difficult to automatically perform this task.

As a consequence, polygonal meshes generated by the T-surface method may not be efficient for some further applications. For instance, for finite element purposes, small triangles must be removed. Consequently, filtering mesh procedures must be applied in order to improve the surface. Mesh smoothing and denoising filtering methods, such as those proposed in [68], could also be useful in this postprocessing step.

We tested the precision of our approach when segmenting a sphere immersed on a uniform noise specified by the image intensity range $[0, 150]$. We found a mean error of 1.58 (pixels) with standard deviation of 2.49 for a $5 \times 5 \times 5$ grid resolution, which we consider acceptable in this case.

This error is due to the projection of T-surfaces as well as the image noise. Following [49, 50], when T-surfaces stops, we can discard the grid and evolve the model without it, avoiding errors due to the projections. However, for noisy images, the convergence of deformable models to the boundaries is poor due to the nonconvexity of the image energy [31].

Anisotropic diffusion applied to 3D images can improve the result, as already demonstrated in Sections 7.8 and 7.9.1. The gradient vector flow (see Section 7.8) can also be applied when the grid is turned off.

Now, let us consider the following question: Would it be possible to implement the reconstruction method through level sets? The relevance of it will be clear in what follows.

The initialization of the model through expression (7.20) is computationally expensive and not efficient if we have more than one front to initialize [75].

The *narrow-band* technique is much more appropriate for this case. The key idea of this technique comes from the observation that the front can be moved by updating the level set function at a small set of points in the neighborhood of the zero set instead of updating it at all the points in the domain (see [46, 61] for details).

To implement this scheme, we need to pre-set a distance Δd to define the narrow band. The front can move inside the narrow band until it collides with the narrow-band frontiers. Then, the function G should be reinitialized by treating the current zero set configuration as the initial one.

Also, this method can be made cheaper by observing that the grid points that do not belong to the narrow band can be treated as sign holders [46], following the same idea of the characteristic function of T-surfaces (Section 7.2.3). Thus, the result of steps (1)–(5) in Section 7.5 can be used to initialize the level sets model if the narrow-band extension technique is applied.

The proposed out-of-core method for segmentation must be analyzed against usual procedures to deal with memory limitations when segmenting a 3D image.

General-purpose methods, such as streaming of Visualization Toolkit and virtual memory of operating systems, have demonstrated less efficiency for scientific visualization applications [24] than the meta-cell technique. The results in Section 7.9 show that the same happens for 3D image segmentation.

Among the special-purpose methods, the simplest strategy would be to subdivide the region of interest in a set of subregions and then segment the structure in each one at a time. Besides being a very tedious procedure, some additional work must be done to put the extracted components together in order to complete the reconstruction.

Another possibility would be to segment 2D slices, extract the corresponding contours, and then reconstruct the surface through the obtained curves. This is particularly difficult for artery segmentation, a case of interest, due to their tree structures and branching characteristics.

On the other hand, having once segmented slice by slice, each 2D image could be binarized (1 inside the extracted contours and 0 otherwise). The obtained binary field could fit in the main memory and then the reconstruction performed through an isosurface extraction method. However, the obtained polygon mesh may not be smooth. The application of mesh smoothing procedures [68] may not be efficient if the data set information is not taken into account. But, if it does not fit into the computer memory, we return to the original problem.

The preprocessing step is very simple for the meta-cell technique applied to image volumes because the data set is regular. The algorithm presented in graphics.cs.ucdavis.edu/research/Slicer.html has a longer preprocessing step. New experiments must be performed to compare both approaches.

Potentially, the most common deformable surface models (energy-minimizing, dynamic deformable surfaces and balloons [48]) can be made out-of-core by using the meta-cell technique. Basically, it can be performed by maintaining the traditional algorithms, by choosing explicit methods to solve the

evolution equations (e.g., expression (7.14)), and using the *processing_list* to guarantee locality during evolution.

Other interesting perspectives for our work are out-of-core implementations of other techniques such as region growing (for segmentation) and level sets (for surface reconstruction).

To show this, let us consider a simple region growing algorithm, which takes a seed point p , and find out the connected set: $\{q \in Image; |I(q) - I(p)| \leq \epsilon\}$. At run-time, we traverse the interval tree and find the active meta-cells. Then, we fill the data cache and perform usual region growing operations [40], but calling *insert_neighbors* for each point p incorporated to the region.

Besides, level sets can be made out-of-core by using the *narrow-band* technique described above. In this case, it is just a matter of observing that the level sets algorithm would only need the image information inside the narrow band. Henceforth, an out-of-core implementation can be provided.

7.11 Conclusions

Deformable models offer an attractive approach for geometry recovery and tracking because these models are able to represent complex and broad shapes variability, particularly in the context of medical imaging.

Despite their capabilities, traditional deformable models suffer from the strong sensitivity to the initial contour position and topological limitations.

Among the possibilities to address these problems, we follow the research line that uses a two-step approach: Firstly, a rough approximation of the boundary is taken. Secondly, the obtained geometry is improved by a topologically adaptable deformable model. The reconstruction method presented in Section 7.5 is a result of our research in this direction.

We have used the T-surfaces model but it is pointed out that level sets could also be used. When T-surfaces stops, we can discard the grid and evolve the model without it to avoid errors due to the projections. Now, GVF can be useful to improve the convergence toward the boundary.

Also, when using deformable surfaces, memory limitations can lower the performance of segmentation applications for large 3D images. Few works have been done to address this problem.

We show that the meta-cell technique is the most suitable data structure to perform out-of-core implementations of segmentation methods. Next, we take advantage of the meta-cell method to present an out-of-core implementation of the segmentation approach presented in Section 7.5.

The experimental results presented in Section 7.9 demonstrate the potential of the segmentation approach in Section 7.5 when augmented with diffusion and out-of-core techniques. This is emphasized with the discussion and perspectives in Section 7.10.

Questions

1. *The static formulation of the original snake model is given by the minimization of the energy functional*

$$E : Ad \rightarrow \mathbb{R},$$

$$E(c) = E_1(c(s)) + E_2(c(s)),$$

defined in Section 7.2.1. Supposing that $c \in C^4$, show that the Euler–Lagrange equations become:

$$-(w_1 c'(s))' + (w_2 c''(s))'' + \nabla P(c(s)) = 0.$$

2. *Discuss the effect of the parameters w_1 and w_2 over the original snake model in exercise 1 by using the following equations for a curve c :*

$$\frac{dc}{d\alpha} = \vec{T}, \quad \frac{d^2c}{d\alpha^2} = K\vec{N},$$

where α is the arc length, K is the curvature, and $\vec{T}$ and $\vec{N}$ are the unitary tangent and normal vectors, respectively.

3. *Show that the original snake model is not invariant under affine transformations given by the general form:*

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

4. *Discuss the role of the characteristic function for the T-surfaces model.*

5. Let us consider a characteristic function in $f : \mathbb{R}^2 \rightarrow \{0, 1\}$ defined over a CF triangulation of $\mathbb{R}^2$. In this case, given a triangle, it can be verified (do it as an exercise) that it has exactly two transverse edges or it does not have transverse edges. Based on this property, write a pseudocode for an algorithm to generate the polygonal curves, after computing the intersections with the triangulation (see Section 7.2.3).
6. Would it be possible to design a T-surfaces model based on a cellular decomposition of the image domain? What would be the advantages over the traditional T-surfaces?
7. Choose a gray scale image, binarize it applying several values of thresholds. Later, with the same initial image, apply the following sequence of operations and compare the results: Canny's edge detector of thresholds 30 and 80; invert the result; apply over the result the erosion operation with a cross structuring element. Observe the isolated regions with other values of thresholds of your choice.
8. Choose a binary image, apply the following sequence of operations and describe the net effect (B is the structuring element of your choice):

$$(a) X_B = (X \ominus B) \oplus B$$

$$(b) X_B = (X \oplus B) \ominus B$$

$$(c) X_B = X / (X \ominus B)$$

(d) $X_B = (X \ominus B_{ob}) / (X \oplus B_{bk})$, where B_{ob} is the set formed from pixels in B that should belong to the object, and B_{bk} is the set formed from pixels in B that should belong to the background.

9. Considering the implicit representation of a curve, $G(x, y) = 0$, show that the normal $\vec{n}$ and the curvature K can be computed by:

$$\vec{n} = \nabla G(x, y), \quad K = \nabla \cdot \left(\frac{\nabla G(x, y)}{\|\nabla G(x, y)\|} \right),$$

respectively, where the gradient and the divergent ($\nabla \cdot$) are computed with respect to the spatial coordinates (x, y) .

10. Take the anisotropic diffusion scheme (see Section 7.8):

$$\frac{\partial \phi}{\partial t} = \operatorname{div} \left(\frac{\nabla I}{(1 + [\|\nabla I\| / T]^2)} \right).$$

Show that if $\|\nabla I\| < T$, the edges are blurring and if $\|\nabla I\| > T$ they become sharper.

11. Let us suppose h and g as constants in the GVF model given by the equation:

$$\frac{\partial u}{\partial t} = g \Delta u + h(\nabla f - u).$$

Consider the stationary solution and take the Fourier transform of the corresponding stationary equation to analyze the GVF in the frequency space.

Bibliography

- [1] Allgower, E. L. and Georg, K., *Numerical Continuation Methods: An Introduction*, Springer-Verlag, Berlin, 1990.
- [2] Amini, A. A., Weymouth, T. E., and Jain, R. C., Using dynamic programming for solving variational problems in vision, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 12, No. 9, pp. 855–867, 1990.
- [3] Bamford, P. and Lovell, B. C., A water immersion algorithm for cytological image segmentation. In: *Proceedings APRS Image Segmentation Workshop*, Sydney, Australia, December 13, 1996, pp. 75–79.
- [4] Bamford, P. and Lovell, B., A two-stage scene segmentation scheme for the automatic collection of cervical cell images, In: *Proceedings of TENCON '97*, Brisbane, Australia, December 1997.
- [5] Berger, M. J. and Oliger, J., Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.*, Vol. 54, pp. 484–512, 1984.
- [6] Black, A. and Yuille, A., editors. *Active Vision*, MIT Press, Cambridge, MA, 1993.
- [7] Bruce, J., Balch, T., and Veloso, M., Fast and inexpensive color image segmentation for interactive robots, In: *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, Vol. 3. October 2000, pp. 2061–2066.
- [8] Burt, P. J., Tsai-Hong, H., and Rosenfeld, A., Segmentation and estimation of image region properties through cooperative hierarchical computation, *IEEE Trans. Syst. Man, Cybern.*, Vol. SMC-11, No. 12, pp. 802–809, 1981.
- [9] Canny, J., A computational approach to edge detection, *Pattern Anal. Mach. Intell.*, Vol. 8, No. 6, pp. 689–698, 1986.
- [10] Carneiro, B. P., Silva, C. T., and Kaufman, A. E., Tetra-cubes: An algorithm to generate 3d isosurfaces based upon tetrahedra. In: *International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'96)*, 1996.

- [11] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic active contours, *Int. J. Comput. Vis.*, Vol. 22, No. 1, pp. 61–79, 1997.
- [12] Chauvin, Y. and Rumelhart, D., *Backpropagation: Theory, Architectures and Applications*, Lawrence Erlbaum, Hillsdale, NJ, 1995.
- [13] Cheng, H. D., Jiang, X. H., Sun, Y., and Wang, J. L., Color image segmentation: Advances and prospects, *Pattern Recogni.* Vol. 34, pp. 2259–2281, 2001.
- [14] Cheung, K. W., Yeung, D. Y., and Chin, R. T., On deformable models for visual pattern recognition, *Pattern Recognit.*, Vol. 35, pp. 1507–1526, 2002.
- [15] Chiang, Y.-J., Farias, R., Silva, C., and Wei, B., A unified infrastructure for parallel out-of-core isosurface and volume rendering of unstructured grids, In: *IEEE Parallel and Large-Data Vis. and Graph.* 2001, 2001.
- [16] Chiang, Y.-J., Silva, C., and Schroeder, W. J., Interactive out-of-core isosurface extraction. In: *IEEE Visualization*, pp. 67–174, 1998.
- [17] Chiang, Y.-J., Silva, C., and Schroeder, W., Interactive out-of-core isosurface extraction. In: *IEEE Visualization '98*, pp. 167–174, 1998.
- [18] Cohen, L. D., On active contour models and balloons, *CVGIP: Image Understanding*, Vol. 53, No. 2, pp. 211–218, 1991.
- [19] Cohen, L. D. and Cohen, I., Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE PAMI* Vol. 15, No. 11, pp. 1131–1147, 1993.
- [20] Cox, Michael and Ellsworth, David, Application-controlled demand paging for out-of-core visualization, In: *IEEE Visualization*, pp. 235–244, 1997.
- [21] Durikovic, R., Kaneda, K., and Yamashita, H., Dynamic contour: A texture approach and contour operations, *Vis. Comput.*, Vol. 11, pp. 277–289, 1995.
- [22] Giraldi, G., Silva, R., Strauss, E., Jimenez, W., and Oliveira, A., A surface extraction approach based on multi-resolution methods and t-surfaces

- framework. Technical Report, National Laboratory for Scientific Computing, <ftp://ftp.lncc.br/pub/report/rep02/rep1002.ps.Z>, 2002.
- [23] Falco, A. X., da Cunha, B. S., and Lotufo, R. A., Design of connected operators using the image foresting transform, *SPIE Med. Imaging*, Vol. 4322, pp. 468–479, 2001.
- [24] Farias, R. and Silva, C., Out-of-core rendering of large unstructured grids, *IEEE Comput. Graph. Appl.*, Vol. 21, No. 4 pp. 42–50, 2001.
- [25] Gilson, S. J., Middleton, I., and Damper, R. I., Neural techniques for outlining the lungs from MR images of the thorax, In: *Proceedings of Symposium of Software Computing World Automation Congress*, May 1998.
- [26] Oliveira, A. A. F. and Giraldi, G. A., Invariant snakes and initialization of deformable models. *Int. J. Image Graphics*, Vol. 4, No. 3, 363–384, 2004.
- [27] Giraldi, G., Farias, R., Schaefer, L., and Silva, R., Improving memory utilization and convergence of deformable models, Technical Report, National Laboratory for Scientific Computing, <ftp://ftp.lncc.br/pub/report/rep03/rep1303.ps.gz>, 2003.
- [28] Schaefer, L., Farias, R., Silva, R., and Giraldi, G. A. Out-of-core segmentation by deformable models, In: *WILE 2003—International Workshop on Fuzzy Logic and Applications*, Italy October 2003, pp. 9–11, Springer-Verlag, Italy.
- [29] Giraldi, G., Strauss, E., Apolinario, A., and Oliveira, A. F., An initialization method for deformable models. In: *5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*, 2001.
- [30] Giraldi, G. A., Gonçalves, L. M., and Oliveira, A. F., Dual topologically adaptable snakes. In: *Proceedings of the Fifth Joint Conference on Information Sciences (JCIS'2000, Vol. 2)—Third International Conference on Computer Vision, Pattern Recognition, and Image Processing*, pp. 103–106, 2000.
- [31] Giraldi, G. A. and Oliveira, A. F., Convexity analysis of snake models based on Hamiltonian formulation, Technical Report, Universidade

- Federal do Rio de Janeiro, Dep. Eng. Sistemas e Computação, <http://www.cos.ufrj.br/relatorios/reltec99/>, 1999.
- [32] Giraldi, G. A., Strauss, E., and Oliveira, A. F., A boundary extraction method based on dual-t-snakes and dynamic programming. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000), 2000.
 - [33] Giraldi, G. A., Strauss, E., and Oliveira, A. F., An initialization method for active contour models. In: Proceedings of the 2000 International Conference on Imaging Science, Systems, and Technology (CISST'2000), 2000.
 - [34] Giraldi, G. A., Strauss, E., and Oliveira, A. F., Dual-t-snakes model for medical imaging segmentation, *Pattern Recognit. Lett.*, Vol. 24, No. 7, pp. 993–1003, 2003.
 - [35] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing*, Addison Wesley, Reading, MA, 1992.
 - [36] Gool, L. V., Moons, T., Powrls, E., and Oosterlinck, A., Vision and lie's approach to invariance, *Image Vis. Comput.*, Vol. 13, No. 4, PP. 259–277, 1995.
 - [37] Gunn, S. R. and Nixon, M. S., A robust snake implementation; a dual active contour, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 19, No. 1, pp. 63–68, 1997.
 - [38] Gunn, Steve R., Dual active contour models for image feature extraction, PhD Thesis, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science, May 1996.
 - [39] Jain, A. K., Duin, R., and Mao, J., Statistical pattern recognition: A review, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 22, No. 1, pp. 4–37, 2000.
 - [40] Jain, Anil K., *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.
 - [41] Jolion, J. M. and Montanvert, A., The adaptive pyramid: A framework for 2d image analysis, *CVGIP: Image Understanding*, Vol. 55, No. 3, pp. 339–348, 1992.

- [42] Kass, M., Witkin, A., and Terzopoulos, D., Snakes: Active contour models, *Int. J. Comput. Vis.*, Vol. 1, No. 4, pp. 321–331, 1988.
- [43] Lai, Kok F. and Chin, R. T., Deformable contours: Modeling and extraction, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 11, pp. 1084–1090, 1995.
- [44] Leymarie, F. and Levine, M. D., Tracking deformable objects in the plane using an active contour model, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 15, No. 6, pp. 617–634, 1993.
- [45] Lorensen, W. E. and Cline, H. E. Marching cubes: A high resolution 3d surface construction algorithm, *Comput. Graph.*, Vol. 21, No. 4, pp. 163–169, July 1987.
- [46] Malladi, R., Sethian, J. A., and Vemuri, B. C., Shape modeling with front propagation: A level set approach, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 2, pp. 158–175, 1995.
- [47] McInerney, T. and Terzopoulos, D., Topologically adaptable snakes. In: *Proc. Of the Fifth Int. Conf. On Computer Vision (ICCV'95)*, Cambridge, MA, USA, pp. 840–845, 1995.
- [48] McInerney, T. and Terzopoulos, D., Deformable models in medical image analysis: A survey, *Med. Image Anal.*, Vol. 1, No. 2, pp. 91–108, 1996.
- [49] McInerney, D. and Terzopoulos, D., Topology adaptive deformable surfaces for medical image volume segmentation, *IEEE Trans. Med. Imaging*, Vol. 18, No. 10, pp. 840–850, 1999.
- [50] McInerney, T. J., Topologically adaptable deformable models for medical image analysis, PhD Thesis, Department of Computer Science, University of Toronto, 1997.
- [51] Olver, P. J., Sapiro, G., and Tannenbaum, A., Affine invariant detection: Edges, active contours, and segments, In: *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, IEEE Computer Society, San Francisco, USA, 1996, pp. 520.
- [52] Perona, P. and Malik, J., Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 12, No. 7, pp. 629–639, 1990.

- [53] Carstensen, J. M. and Fisker, R., On parameter estimation in deformable models, In: Fourteenth International Conference on Pattern Recognition, August 16–20, pp. 762–766, 1998.
- [54] Rowley, Henry, Baluja, Shumeet, and Kanade, Takeo, Neural network-based face detection. In: Computer Vision and Pattern Recognition '96, June 1996.
- [55] Rumelhart, D. E., Hinton, G., and Williams, R., Learning representations by back-propagation errors, *Nature*, Vol. 323, pp. 533–536, 1986.
- [56] Sapiro, G., Color snakes. Technical Report, Hewlett-Packard Labs, guille@hpl.hp.com, 1995.
- [57] Sapiro, G., Color snakes. *Comput. Vis. Image Underst.*, Vol. 68, No. 2, pp. 247–253, 1997.
- [58] Sapiro, G. and Tannenbaum, A., Affine invariant scale-space, *Int. J. Comput. Vis.*, Vol. 11, No. 1, pp. 25–44, 1993.
- [59] Sarti, A., Ortiz, C., Lockett, S., and Malladi, R., A unified geometric model for 3d confocal image analysis in cytology. In *Proc. International Symposium on Computer Graphics, Image Processing, and Vision (SIBGRAPI'98)*, pp. 69–76, 1998.
- [60] Schroeder, W., Martin, K., and Lorensen, B., *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*, Prentice-Hall PTR, Englewood Cliffs, NJ, 1998.
- [61] Sethian, J. A., *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*, Cambridge University Press, Cambridge 1996.
- [62] Storvik, G., A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 16, No. 10, pp. 976–986, 1994.
- [63] Strauss, E., Jimenez, W., Giraldi, G. A., Silva, R., and Oliveira, A. F., A semi-automatic surface reconstruction framework based on t-surfaces and isosurface extraction methods. In: *International Symposium on*

Computer Graphics, Image Processing and Vision (SIBGRAP'2002), 2002.

- [64] Sutton, P. and Hansen, C., Accelerated isosurface extraction in time-varying fields, *IEEE Trans. Vis. Comput. Graph.*, Vol. 6, No. 2, pp. 98–107, 2000.
- [65] Szelinski, R., Tonnensen, D., and Terzopoulos, D., Modeling surfaces of arbitrary topology with dynamic particles. In: *Proc. Conf. Computer Vision and Pattern Recognition(CVPR'93)*, New York, NY, Los Alamitos CA, pp. 82–87, IEEE Computer Society Press, Los Alamitos, CA, 1993.
- [66] Tang, B., Sapiro, G., and Caselles, V., Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case, *Int. J. Comput. Vis.*, Vol. 36, No. 2 pp. 149–161, 2000.
- [67] Tang, B., Sapiro, G., and Caselles, V., Color image enhancement via chromaticity diffusion, *IEEE Trans. Image Proc.*, Vol. 10, pp. 701–707, 2001.
- [68] Taubin, G., Linear anisotropic mesh filtering, Technical Report, IBM T.J. Watson Research Center, Report RC-22213 10/18/2001, 2001.
- [69] Tou, J., *Pattern Recognition Principles*, Addison-Wesley, Reading, MA, 1977.
- [70] Udupa, J. K. and Samarasekera, S., Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation, *Graph. Models, Image Process.*, Vol. 58, No. 3, pp. 246–261, 1996.
- [71] Ueng, Shyh-Kuang, Sikorski, C., and Kwan-Liu, Ma, Out-of-core streamline visualization on large unstructured meshes, *IEEE Trans. Vis. Comput. Graph.*, Vol. 3, No. 4, pp. 370–380, 1997.
- [72] Weigend, A., Rumelhart, D., and Huberman, B., Generalization by weight-elimination with application to forecasting, *Adv. Neural Inf. Process.* Vol. 3, pp. 875–882, 1991.
- [73] Wilhelms, J. and Gelder, A. V., Octrees for faster isosurface generation, *ACM SIGGRAPH Comput. Graph.*, Vol. 24, No. 5, pp. 57–62, 1990.

- [74] Williams, C. K. I., Revow, M., and Hinton, G. E., Instantiating deformable models with a neural net, *Comput. Vis. Image Underst.*, Vol. 68, No.1, pp. 120–126, 1997.
- [75] Niessen, W.J., ter Haar Romery, B.M., and Viergever, M. A., Geodesic deformable models for medical image analysis, *IEEE Trans. Med. Imaging*, Vol. 17, No. 4, pp. 634–641, 1998.
- [76] Xu, C., Pham, D., Rettmann, M., Yu, D., and Prince, J., Reconstruction of the human cerebral cortex from magnetic resonance images, *IEEE Trans. Med. Imaging*, Vol. 18, No. 6, pp. 467–480, 1999.
- [77] Xu, C. and Prince, J. L., Snakes, shapes, and gradient vector flow, *IEEE Trans. Image Process.*, Vol. 7, No. 3, pp. 359–369, 1998.
- [78] Xu, C. and Prince, J. L., Global optimality of gradient vector flow, In: *Proc. of the Conference on Information Sciences and Systems*, Princeton University, March 2000.
- [79] Xu, Gang, Segawa, E., and Tsuji, S., Robust active contours with insensitive parameters, *Pattern Recognit.*, Vol. 27, No. 7, pp. 879–884, 1994.

Chapter 8

Level Set Segmentation of Biological Volume Datasets

David Breen¹, Ross Whitaker², Ken Museth³, and Leonid Zhukov⁴

8.1 Introduction

This chapter addresses the common problem of building meaningful 3D models of complex structures from noisy datasets generated from 3D imaging devices. In certain circumstances such data can be visualized *directly* [1–4]. While direct techniques can provide useful insights into volume data, they are insufficient for many problems. For instance, direct volume rendering techniques typically do not remove occluding structures, i.e., they do not allow one to “peel back” the various layers of the data to expose the inner structures that might be of interest. They also do not generate the models needed for quantitative study/analysis of the visualized structures. Furthermore, direct visualization techniques typically do not perform well when applied directly to noisy data, unless one filters the data first. Techniques for filtering noisy data are abundant in the literature, but there is a fundamental limitation—filtering that reduces noise tends to distort the shapes of the objects in the data. The challenge is to find methods which present the best trade-off between fidelity and noise.

Level set segmentation relies on a surface-fitting strategy, which is effective for dealing with both small-scale noise and smoother intensity fluctuations in

¹Department of Computer Science, Drexel University, Philadelphia, PA 19104, USA

²School of Computing, University of Utah, Salt Lake City, UT 84112, USA

³Department of Science and Technology, Linköping University, 601 74 Norrkoeping, Sweden

⁴Department of Computer Science, California Institute of Technology, Pasadena, CA 91125, USA

volume data. The level set segmentation method, which is well documented in the literature [5–8], creates a new volume from the input data by solving an initial value partial differential equation (PDE) with user-defined feature-extracting terms. Given the local/global nature of these terms, proper initialization of the level set algorithm is extremely important. Thus, level set deformations alone are not sufficient, they must be combined with powerful initialization techniques in order to produce successful segmentations. Our level set segmentation approach consists of defining a set of suitable preprocessing techniques for initialization and selecting/tuning different feature-extracting terms in the level set algorithm. We demonstrate that combining several preprocessing steps, data analysis and level set deformations produce a powerful toolkit that can be applied, under the guidance of a user, to segment a wide variety of volumetric data.

There are more sophisticated strategies for isolating meaningful 3D structures in volume data. Indeed, the so-called *segmentation problem* constitutes a significant fraction of the literature in image processing, computer vision, and medical image analysis. For instance, statistical approaches [9–12] typically attempt to identify tissue types, voxel by voxel, using a collection of measurements at each voxel. Such strategies are best suited to problems where the data is inherently multivalued or where there is sufficient prior knowledge [13] about the shape or intensity characteristics of the relevant anatomy. Alternatively, anatomical structures can be isolated by grouping voxels based on local image properties. Traditionally, image processing has relied on collections of edges, i.e. high-contrast boundaries, to distinguish regions of different types [14–16]. Furthermore deformable models, incorporating different degrees of domain-specific knowledge, can be *fitted* to the 3D input data [17, 18].

This chapter describes a level set segmentation framework, as well as the preprocessing and data analysis techniques needed to segment a diverse set of biological volume datasets. Several standard volume processing algorithms have been incorporated into framework for segmenting conventional datasets generated from MRI, CT, and TEM scans. A technique based on moving least-squares has been developed for segmenting multiple nonuniform scans of a single object. New scalar measures have been defined for extracting structures from diffusion tensor MRI scans. Finally, a direct approach to the segmentation of incomplete tomographic data using density parameter estimation is described. These techniques, combined with level set surface deformations, allow us to segment many different types of biological volume datasets.

8.2 Level Set Surface Models

When considering deformable models for segmenting 3D volume data, one is faced with a choice from a variety of surface representations, including triangle meshes [19, 20], superquadrics [21–23], and many others [18, 24–29]. Another option is an implicit level set model, which specifies the surface as a *level set* of a scalar volumetric function, $\phi : U \mapsto \mathbb{R}$, where $U \subset \mathbb{R}^3$ is the range of the surface model. Thus, a surface S is

$$S = \{\mathbf{s} | \phi(\mathbf{s}) = k\}, \quad (8.1)$$

with an isovalue k . In other words, S is the set of points $\mathbf{s}$ in $\mathbb{R}^3$ that composes the k th isosurface of ϕ . The embedding ϕ can be specified as a regular sampling on a rectilinear grid.

Our overall scheme for segmentation is largely based on the ideas of Osher *et al.* [30] that model propagating surfaces with (time-varying) curvature-dependent speeds. The surfaces are viewed as a specific level set of a higher dimensional function ϕ —hence the name level set methods. These methods provide the mathematical and numerical mechanisms for computing surface deformations as isovalues of ϕ by solving a partial differential equation on the 3D grid. That is, the level set formulation provides a set of numerical methods that describes how to manipulate the grayscale values in a volume, so that the isosurfaces of ϕ move in a prescribed manner (shown in Fig. 8.1). This chapter does not present a comprehensive review of level set methods, but merely introduces the basic concepts and demonstrates how they may be applied to

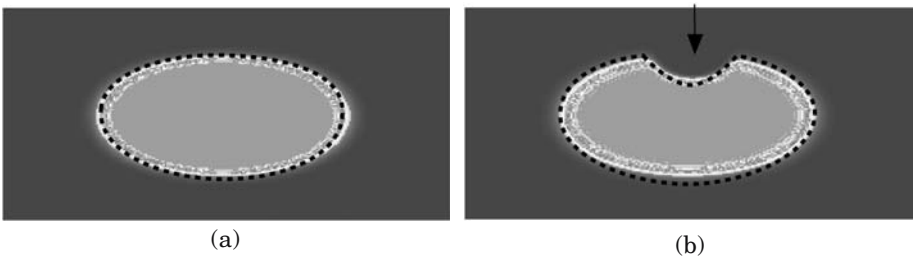


Figure 8.1: (a) Level set models represent curves and surfaces implicitly using grayscale images. For example, an ellipse is represented as the level set of an image shown here. (b) To change the shape of the ellipse we modify the grayscale values of the image by solving a PDE.

the problem of volume segmentation. For more details on level set methods see [7, 31].

There are two different approaches to defining a deformable surface from a level set of a volumetric function as described in Eq. (8.1). Either one can think of $\phi(\mathbf{s})$ as a *static* function and change the isovalue $k(t)$ or alternatively fix k and let the volumetric function *dynamically* change in time, i.e. $\phi(\mathbf{s}, t)$. Thus, we can mathematically express the static and dynamic models respectively as

$$\phi(\mathbf{s}) = k(t), \quad (8.2a)$$

$$\phi(\mathbf{s}, t) = k. \quad (8.2b)$$

To transform these definitions into partial differential equations which can be solved by standard numerical techniques, we differentiate both sides of Eq. (8.2) with respect to time t , and apply the chain rule:

$$\nabla\phi(\mathbf{s}) \frac{d\mathbf{s}}{dt} = \frac{dk(t)}{dt}, \quad (8.3a)$$

$$\frac{\partial\phi(\mathbf{s}, t)}{\partial t} + \nabla\phi(\mathbf{s}, t) \cdot \frac{d\mathbf{s}}{dt} = 0. \quad (8.3b)$$

The static equation (8.3a) defines a boundary value problem for the time-independent volumetric function ϕ . This static level set approach has been solved [32, 33] using “Fast Marching Methods.” However, it inherently has some serious limitations following the simple definition in Eq. (8.2a). Since ϕ is a function (i.e. single-valued), isosurfaces cannot self-intersect over time, i.e. shapes defined in the static model are strictly expanding or contracting over time. However, the dynamic level set approach of eq. (8.3b) is much more flexible and shall serve as the basis of the segmentation scheme in this chapter. Equation (8.3b) is sometimes referred to as a “Hamilton–Jacobi-type” equation and defines an initial value problem for the time-dependent ϕ . Throughout the remainder of this chapter we shall, for simplicity, refer to this dynamical approach as the level set method, and not consider the static alternative.

Thus, to summarize the essence of the (dynamic) level set approach, let $d\mathbf{s}/dt$ be the movement of a point on a surface as it deforms, such that it can be expressed in terms of the position of $\mathbf{s} \in U$ and the geometry of the surface at that point, which is, in turn, a differential expression of the implicit function, ϕ .

This gives a partial differential equation on ϕ : $\mathbf{s} \equiv \mathbf{s}(t)$

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{d\mathbf{s}}{dt} = \|\nabla \phi\| \mathcal{F}(\mathbf{s}, \mathbf{n}, \phi, D\phi, D^2\phi, \dots), \quad (8.4a)$$

$$\mathcal{F}() \equiv \mathbf{n} \cdot \frac{d\mathbf{s}}{dt}, \quad (8.4b)$$

where $\mathcal{F}()$ is a user-created “speed” term that defines the speed of the level set at point $\mathbf{s}$ in the direction of the local surface normal $\mathbf{n}$ at $\mathbf{s}$. $\mathcal{F}()$ may depend on a variety of local and global measures including the order- n derivatives of ϕ , $D^n\phi$, evaluated at $\mathbf{s}$, as well as other functions of $\mathbf{s}$, $\mathbf{n}$, ϕ , and external data. Because this relationship applies to every level set of ϕ , i.e. all values of k , this equation can be applied to all of U , and therefore the movements of *all* the level set surfaces embedded in ϕ can be calculated from Eq. (8.4).

The level set representation has a number of practical and theoretical advantages over conventional surface models, especially in the context of deformation and segmentation. First, level set models are topologically flexible, they easily represent complicated surface shapes that can form holes, split to form multiple objects, or merge with other objects to form a single structure. These models can incorporate many (millions) degrees of freedom, and therefore they can accommodate complex shapes such as the dendrite in Fig. 8.7. Indeed, the shapes formed by the level sets of ϕ are restricted only by the resolution of the sampling. Thus, there is no need to reparameterize the model as it undergoes significant changes in shape.

The solutions to the partial differential equations described above are computed using finite differences on a discrete grid. The use of a grid and discrete time steps raises a number of numerical and computational issues that are important to the implementation. However, it is outside of the scope of this chapter to give a detailed mathematical description of such a numerical implementation. Rather we shall provide a summary in a later section and refer to the actual source code which is publicly available⁵.

Equation (8.4) can be solved using finite forward differences if one uses the up-wind scheme, proposed by Osher *et al.* [30], to compute the spatial derivatives. This up-wind scheme produces the motion of level set models over the entire range of the embedding, i.e., for all values of k in Eq. (8.2). However, this

⁵The level set software used to produce the morphing results in this chapter is available for public use in the VISPACK libraries at <http://www.cs.utah.edu/~whitaker/vispack>.

method requires updating *every voxel in the volume for each iteration*, which means that the computation time increases as a function of the volume, rather than the surface area, of the model. Because segmentation requires only a single model, the calculation of solutions over the entire range of isovalues is an unnecessary computational burden.

This problem can be avoided by the use of *narrow-band* methods, which compute solutions only in a narrow band of voxels that surround the level set of interest [34, 35]. In a previous work [36] we described an alternative numerical algorithm, called the sparse-field method, that computes the geometry of only a small subset of points in the range and requires a fraction of the computation time required by previous algorithms. We have shown two advantages to this method. The first is a significant improvement in computation times. The second is increased accuracy when fitting models to forcing functions that are defined to subvoxel accuracy.

8.3 Segmentation Framework

The level set segmentation process has two major stages, initialization and level set surface deformation, as shown in Fig. 8.2. Each stage is equally important for generating a correct segmentation. Within our framework a variety of core operations are available in each stage. A user must “mix-and-match” these operations in order to produce the desired result [37]. Later sections describe specialized operations for solving specific segmentation problems that build upon and extend the framework.

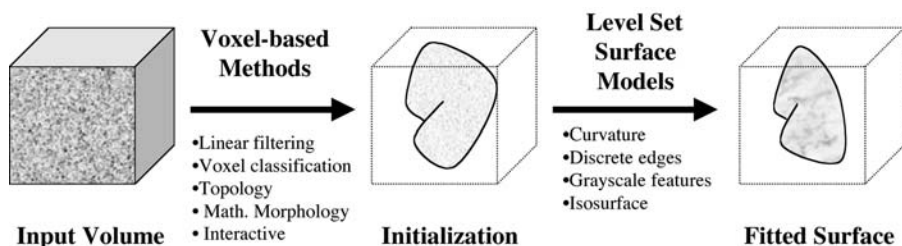


Figure 8.2: Level set segmentation stages—initialization and surface deformation.

8.3.1 Initialization

Because level set models move using gradient descent, they seek *local solutions*, and therefore the results are strongly dependent on the initialization, i.e., the starting position of the surface. Thus, one controls the nature of the solution by specifying an initial model from which the surface deformation process proceeds. We have implemented both computational (i.e. “semi-automated”) and manual/interactive initialization schemes that may be combined to produce reasonable initial estimates directly from the input data.

Linear filtering: We can filter the input data with a low-pass filter (e.g. Gaussian kernel) to blur the data and thereby reduce noise. This tends to distort shapes, but the initialization need only be approximate.

Voxel classification: We can classify pixels based on the filtered values of the input data. For grayscale images, such as those used in this chapter, the classification is equivalent to high and low thresholding operations. These operations are usually accurate to only voxel resolution (see [12] for alternatives), but the deformation process will achieve subvoxel results.

Topological/logical operations: This is the set of basic voxel operations that takes into account position and connectivity. It includes unions or intersections of voxel sets to create better initializations. These logical operations can also incorporate user-defined primitives. Topological operations consist of connected-component analyses (e.g. flood fill) to remove small pieces or holes from objects.

Morphological filtering: This includes binary and grayscale morphological operators on the initial voxel set. For the results in the chapter we implement openings and closings using *morphological propagators* [38,39] implemented with level set surface models. This involves defining offset surfaces of ϕ by expanding/contracting a surface according to the following PDE,

$$\frac{\partial \phi}{\partial t} = \pm |\nabla \phi|, \quad (8.5)$$

up to a certain time t . The value of t controls the offset distance from the original surface of $\phi(t = 0)$. A dilation of size α , D_α , corresponds to the solution of Eq. (8.5) at $t = \alpha$ using the positive sign, and likewise erosion, E_α , uses the negative sign. One can now define a morphological opening operator

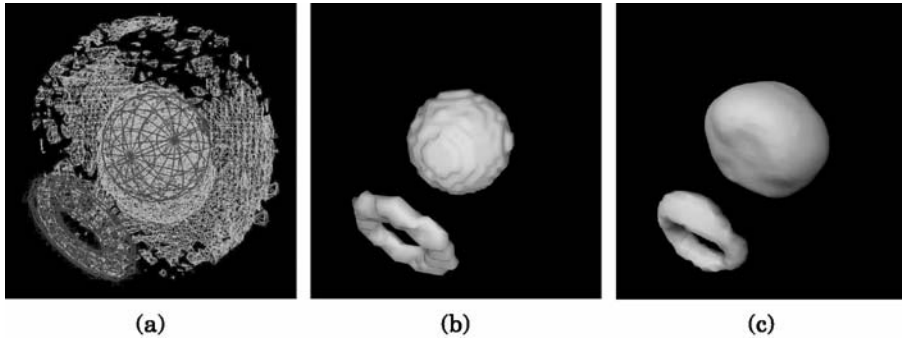


Figure 8.3: (a) Interactively positioning a CSG model relative to a Marching Cubes mesh. (b) Isosurface of a binary scan conversion of the initialization CSG model. (c) Final internal embryo structures.

O_α by first applying an erosion followed by a dilation of ϕ , i.e. $O_\alpha\phi = D_\alpha \circ E_\alpha\phi$, which removes small pieces or thin appendages. A closing is defined as $C_\alpha\phi = E_\alpha \circ D_\alpha\phi$, and closes small gaps or holes within objects. Both operations have the qualitative effect of low-pass filtering the isosurfaces in ϕ —an opening by removing material and a closing by adding material. Both operations tend to distort the shapes of the surfaces on which they operate, which is acceptable for the initialization because it will be followed by a surface deformation.

User-specified: For some applications it is desirable and easier for the user to interactively specify the initial model. Here, the user creates a Constructive Solid Geometry (CSG) model which defines the shape of the initial surface. In Fig. 8.3(a) the CSG model in blue is interactively positioned relative to a Marching Cubes mesh extracted from the original dataset. The CSG model is scan-converted into a binary volume, with voxels simply marked as inside (1) or outside (0), using standard CSG evaluation techniques [40]. An isosurface of the initialization volume dataset generated from the torus and sphere is presented in Fig. 8.3(b). This volume dataset is then deformed to produce the final result seen in Fig. 8.3(c).

8.3.2 Level Set Surface Deformation

The initialization should position the model near the desired solution while retaining certain properties such as smoothness, connectivity, etc. Given a rough initial estimate, the surface deformation process moves the surface model to-

ward specific features in the data. One must choose those properties of the input data to which the model will be attracted and what role the shape of the model will have in the deformation process. Typically, the deformation process combines a data term with a smoothing term, which prevents the solution from fitting too closely to noise-corrupted data. There are a variety of surface-motion terms that can be used in succession or simultaneously, in a linear combination to form $\mathcal{F}(\mathbf{x})$ in Eq. (8.4).

Curvature: This is the smoothing term. For the work presented here we use the mean curvature of the isosurface H to produce

$$\mathcal{F}_{\text{curv}}(\mathbf{x}) = H = \left(\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right). \quad (8.6)$$

The mean curvature is also the normal variation of the surface area (i.e., minimal surface area). There are a variety of options for second-order smoothing terms [41], and the question of efficient, effective higher-order smoothing terms is the subject of ongoing research [7, 31, 42]. For the work in this chapter, we combine mean curvature with one of the following three terms, weighting it by a factor β , which is tuned to each specific application.

Edges: Conventional edge detectors from the image processing literature produce sets of “edge” voxels that are associated with areas of high contrast. For this work we use a gradient magnitude threshold combined with nonmaximal suppression, which is a 3D generalization of the method of Canny [16]. The edge operator typically requires a scale parameter and a gradient threshold. For the scale, we use small, Gaussian kernels with standard deviation $\sigma = [0.5, 1.0]$ voxel units. The threshold depends on the contrast of the volume. The distance transform on this edge map produces a volume that has minima at those edges. The gradient of this volume produces a field that attracts the model to these edges. The edges are limited to voxel resolution because of the mechanism by which they are detected. Although this fitting is not sub-voxel accurate, it has the advantage that it can pull models toward edges from significant distances, and thus inaccurate initial estimates can be brought into close alignment with high-contrast regions, i.e. edges, in the input data. If $\mathcal{E}$ is the set of edges, and $D_{\mathcal{E}}(\mathbf{x})$ is the distance transform to those edges, then the movement of the surface model is given by

$$\mathcal{F}_{\text{edge}}(\mathbf{x}) = \mathbf{n} \cdot \nabla D_{\mathcal{E}}(\mathbf{x}). \quad (8.7)$$

Grayscale features—gradient magnitude: Surface models can also be attracted to certain grayscale features in the input data. For instance, the gradient magnitude indicates areas of high contrast in volumes. By following the gradient of such grayscale features, surface models are drawn to minimum or maximum values of that feature. Typically, grayscale features, such as the gradient magnitude, are computed with a scale operator, e.g., a derivative-of-Gaussian kernel. If models are properly initialized, they can move according to the *gradient of the gradient magnitude* and settle onto the edges of an object at a resolution that is finer than the original volume.

If $G(\mathbf{x})$ is some grayscale feature, for instance $G(\mathbf{x}) = |\nabla I(\mathbf{x})|$, where $I(\mathbf{x})$ is the input data (appropriately filtered—we use Gaussian kernels with $\sigma \approx 0.5$), then

$$\mathcal{F}_{\text{grad}}(\mathbf{x}) = \mathbf{n} \cdot (\pm \nabla G(\mathbf{x})), \quad (8.8)$$

where a positive sign moves surface toward maxima and the negative sign toward minima.

Isosurface: Surface models can also expand or contract to conform to isosurfaces in the input data. To a first order approximation, the distance from a point $\mathbf{x} \in U$ to the k -level surface of I is given by $(I(\mathbf{x}) - k) / |\nabla I|$. If we let $g(\alpha)$ be a fuzzy threshold, e.g., $g(\alpha) = \alpha / \sqrt{1 + \alpha^2}$, then

$$\mathcal{F}_{\text{iso}}(\mathbf{x}) = g\left(\frac{I(\mathbf{x}) - k}{|\nabla I|}\right) \quad (8.9)$$

causes the surfaces of ϕ to expand or contract to match the k isosurface of I . This term combined with curvature or one of the other fitting terms can create “quasi-isosurfaces” that also include other considerations, such as smoothness or edge strength.

8.3.3 Framework Results

Figure 8.4 presents one slice from an MRI scan of a mouse embryo, and an isosurface model of its liver extracted from the unprocessed dataset. Figure 8.5 presents 3D renderings of the sequence of steps performed on the mouse MRI data to segment the liver. The first step is the initialization, which includes smoothing the input data, thresholding followed by a flood fill to remove isolated holes, and finally applying morphological operators to remove small gaps and protrusions on the surface. The second (surface deformation) step

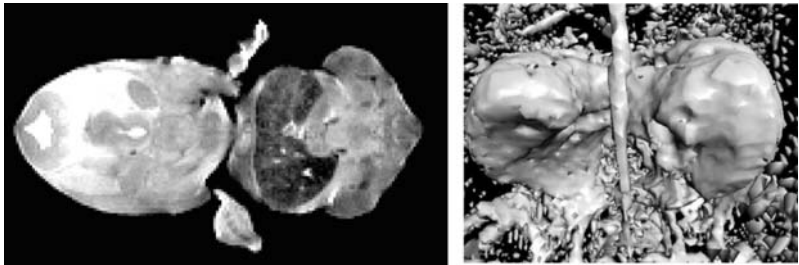


Figure 8.4: (Left) one slice of a $256 \times 128 \times 128$ MR scan of a mouse embryo. The central dark structure is its liver. (Right) a dual-threshold surface rendering highlights the segmentation problem.

first involves fitting to discrete edges and then to the gradient magnitude. This produces a significant improvement over the result in Fig. 8.4. Figure 8.8(a) presents several other structures that were segmented from the mouse embryo dataset. The skin (gray) and the liver (blue) were isolated using computational initialization. The brain ventricles (red) and the eyes (green) were segmented with interactive initialization.

The same set of initialization and surface deformation steps may be combined to extract a model of a spiny dendrite from the transmission electron microscopy (TEM) scan presented in Fig. 8.6(a). An isosurface extracted from the scan is presented in Fig. 8.6(b). Figure 8.7 shows the results of the proposed method compared to the results of a manual segmentation, which took approximately 10 hours of slice-by-slice hand contouring. The manual method suffers from slice-wise artifacts, and, because of the size and complexity of the dataset, the manual segmentation is unable to capture the level of detail that we obtain with the surface-fitting results. Manual segmentation can, however, form connections that are not well supported by the data in order to complete

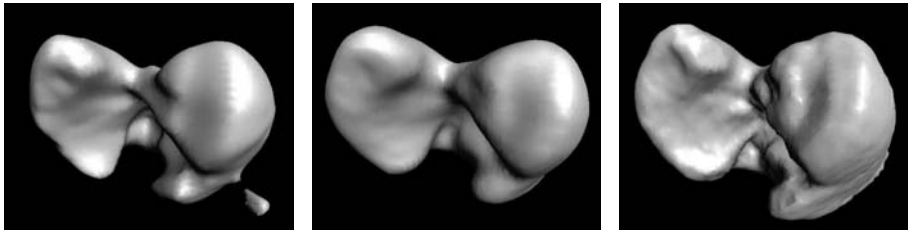


Figure 8.5: (Left) the initialization of a mouse liver dataset using morphology to remove small pieces and holes. (Center) surface fitting to discrete edges. (Right) the final fit to maxima of gradient magnitude.

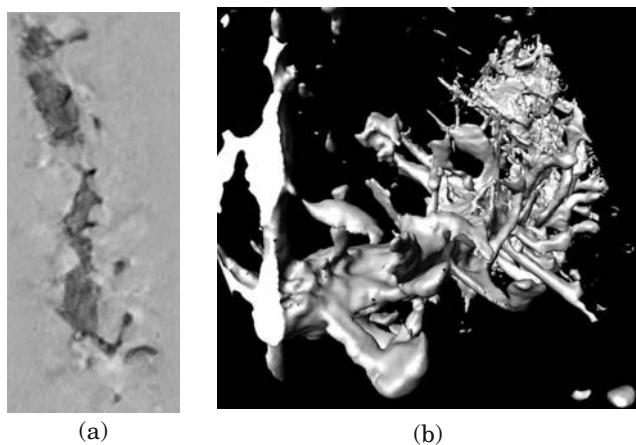


Figure 8.6: (a) One slice of a $154 \times 586 \times 270$ TEM scan of a spiny dendrite shows low contrast and high noise content in a relatively complex dataset. (b) An isosurface rendering, with prefiltering, shows how noise and inhomogeneities in density interfere with visualizing the 3D structure of the dendrite.

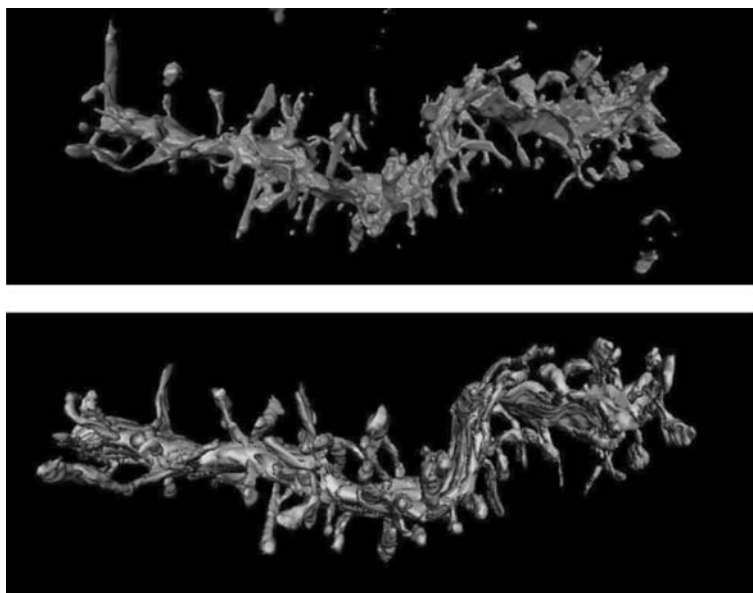


Figure 8.7: (Top) rendering of a dendrite segmented using our proposed method. (Bottom) rendering of a manual segmentation of the same dendrite.

the “spines” that cover this dendrite. These types of “judgments” that humans make when they perform such tasks by hand are a mixed blessing. Humans can use high-level knowledge about the problem to fill in where the data is weak, but the expectations of a trained operator can interfere with seeing unexpected or unusual features in the data.

Figure 8.8(c) presents models from four samples of an MR series of a developing frog embryo. The top left image (hour 9) shows the first evident structure, the blastocoel, in blue, surrounded by the outside casing of the embryo in gray.

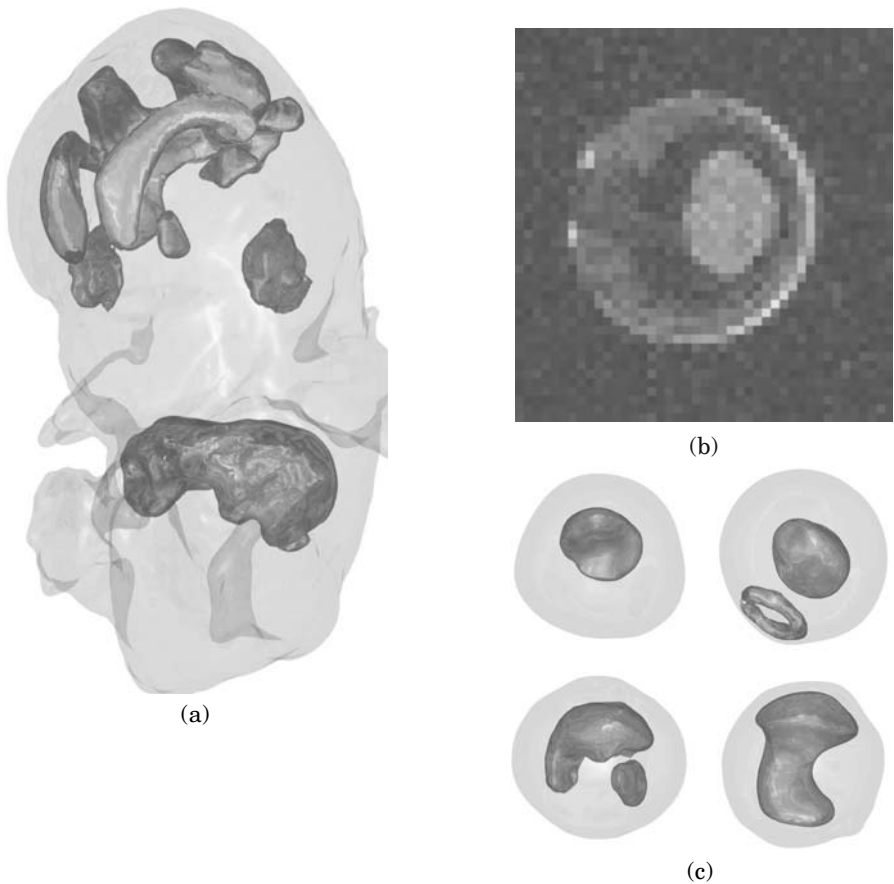


Figure 8.8: (a) Final mouse embryo model with skin (gray), liver (blue), brain ventricles (red), and eyes (green). (b) Hour 16 dataset. (c) Geometric structures extracted from MRI scans of a developing frog embryo, with blastocoel (blue), blastoporal lip (red), and archenteron (green). Hour 9 (top left), hour 16 (top right), hour 20 (bottom left), and hour 30 (bottom right).

Table 8.1: Parameters for processing example datasets

Dataset	Initialization	Surface Fitting
Dendrite	1. Gaussian blur $\sigma = 0.5$ 2. Threshold: $I < 127$ 3. Fill isolated holes 4. Morphology: $O_{0.5} \circ C_{1.5}$	1. Edge fitting: $\sigma = 0.75$, threshold = 6, $\beta = 0.1$ 2. Gradient magnitude fitting: $\sigma = 0.5$, $\beta = 1.0$
Mouse	1. Gaussian blur $\sigma = 0.5$ 2. Threshold: $I > 3$, $I < 60$ 3. Fill isolated holes 4. Morphology: $O_{2.0} \circ C_{3.0}$	1. Edge fitting: $\sigma = 0.75$, threshold = 20, $\beta = 2$ 2. Gradient magnitude fitting: $\sigma = 0.5$, $\beta = 16.0$
Frog	1. Interactive	1. Gradient magnitude fitting: $\sigma = 1.25$, $\beta = 1.0$

The top right image (hour 16) demonstrates the expansion of the blastocoel and the development of the blastoporal lip in red. In the bottom left image (hour 20) the blastoporal lip has collapsed, the blastocoel has contracted, and the archenteron in green has developed. In the bottom right image (hour 30) the blastocoel has collapsed and only the archenteron is present. For this dataset it was difficult to isolate structures only based on their voxel values. We therefore used our interactive techniques to isolate (during initialization) most of the structures in the frog embryo samples.

Table 8.1 describes for each dataset the specific techniques and parameters we used for the results in this section. These parameters were obtained by first making a sensible guess based on the contrasts and sizes of features in the data and then using trial and error to obtain acceptable results. Each dataset was processed between four and eight times to achieve these results. More tuning could improve things further, and once these parameters are set, they work moderately well for similar modalities with similar subjects. The method is iterative, but the update times are proportional to the surface area. On an SGI 180 MHz MIPS 10000 machine, the smaller mouse MR dataset required approximately 10 min of CPU time, and the dendrite dataset ran for approximately 45 min. Most of this time was spent in the initialization (which requires several complete passes through the data) and in the edge detection. The frog embryo datasets needed only a few minutes of processing time, because they did not require computational initialization and are significantly smaller than the other example datasets.

8.4 Segmentation From Multiple Nonuniform Volume Datasets

Many of today's volumetric datasets are generated by medical MR, CT, and other scanners. A typical 3D scan has a relatively high resolution in the scanning X - Y plane, but much lower resolution in the axial Z direction. The difference in resolution between the in-plane and out-of-plane samplings can easily range between a factor of 5 and 10, see Fig. 8.9. This occurs both because of physical constraints on the thickness of the tissue to be excited during scanning (MR), total tissue irradiation (CT), and scanning time restrictions. Even when time is not an issue, most scanners are by design incapable of sampling with high resolution in the out-of-plane direction, producing anisotropic "brick-like" voxels.

The nonuniform sampling of an object or a patient can create certain problems. The inadequate resolution in the Z direction implies that small or thin structures will not be properly sampled, making it difficult to capture them during surface reconstruction and object segmentation. One way to address this problem is to scan the same object from multiple directions, with the hope that the small structures will be adequately sampled in one of the scans. Generating several scans of the same object then raises the question of how to properly combine the information contained in these multiple datasets. Simply merging the individual scans does not necessarily assemble enough samples to produce a high resolution volumetric model. To address this problem we have developed a method for deforming a level set model using velocity information derived from multiple volume datasets with nonuniform resolution in order to produce a single high-resolution 3D model [43]. The method locally approximates the values of the multiple datasets by fitting a distance-weighted polynomial using moving least-squares (MLS) [44, 45]. Directional 3D edge information that may be used during the surface deformation stage is readily derived from MLS, and integrated within our segmentation framework.

The proposed method has several beneficial properties. Instead of merging all of the input volumes by global resampling (interpolation), we locally approximate the derivatives of the intensity values by MLS. This local versus global approach is feasible because the level set surface deformation only requires edge information in a narrow band around the surface. Consequently, the

MLS calculation is only performed in a small region of the volume, rather than throughout the whole volume, making the computational cost proportional to the object surface area [36]. As opposed to many interpolation schemes, the MLS method is stable with respect to noise and imperfect registrations [46]. Our implementation also allows for small intensity attenuation artifacts between the multiple scans thereby providing gain-correction. The distance-based weighting employed in our method ensures that the contributions from each scan are properly merged into the final result. If a slice of data from one scan is closer to a point of interest on the model, the information from this scan will contribute more heavily to determining the location of the point.

To the best of our knowledge there is no previous work on creating deformable models directly from multiple volume datasets. While there has been previous work on 3D level set segmentation and reconstruction [5, 6, 8, 41, 47], it has not been based on multiple volume datasets. However, 3D models have been generated from multiple range maps [29, 36, 48, 49], but the 2D nature of these approaches is significantly different from the 3D problem being addressed here. The most relevant related projects involve merging multiple volumes to produce a single high-resolution volume dataset [50, 51], and extracting edge information from a single nonuniform volume [52]. Our work does not attempt to produce a high-resolution merging of the input data. Instead, our contribution stands apart from previous work because it deforms a model based on local edge information derived from multiple nonuniform volume datasets.

We have demonstrated the effectiveness of our approach on three multi-scan datasets. The first two examples are derived from a single high-resolution volume dataset that has been subsampled in the X , Y , and Z directions. Since these nonuniform scans are extracted from a single dataset, they are therefore perfectly aligned. The first scan is derived from a high-resolution MR scan of a 12-day-old mouse embryo, which has already had its outer skin isolated with a previous segmentation process. The second example is generated from a laser scan reconstruction of a figurine. The third example consists of multiple MR scans of a zucchini that have been imperfectly aligned by hand. The first two examples show that our method is able to perform level set segmentation from multiple nonuniform scans of an object, picking up and merging features only found in one of the scans. The second example demonstrates that our method generates satisfactory results, even when there are misalignments in the registration.

8.4.1 Method Description

We have formulated our approach to 3D reconstruction of geometric models from multiple nonuniform volumetric datasets within our level set segmentation framework. Recall that speed function $\mathcal{F}()$ describes the velocity at each point on the evolving surface in the direction of the local surface normal. All of the information needed to deform a surface is encapsulated in the speed function, providing a simple, unified approach to evolving the surface. In this section we define speed functions that allow us to solve the multiple-data segmentation problem. The key to constructing suitable speed terms is 3D directional edge information derived from the multiple datasets. This problem is solved using a moving least-squares scheme that extracts edge information by locally fitting sample points to high-order polynomials.

8.4.1.1 Level Set Speed Function for Segmentation

Many different speed functions have been proposed over the years for segmentation of a single volume dataset [5, 6, 8, 41]. Typically such speed functions consist of a (3D) image-based feature attraction term and a smoothing term which serves as a regularization term that lowers the curvature and suppresses noise in the input data. From computer vision it is well known that features, i.e. significant changes in the intensity function, are conveniently described by an edge detector [53]. There exists a very large body of work devoted to the problem of designing optimal edge detectors for 2D images [14, 16], most of which are readily generalized to 3D. For this project we found it convenient to use speed functions with a 3D directional edge term that moves the level set toward the maximum of the gradient magnitude. This gives a term equivalent to Eq. (8.8),

$$\mathcal{F}_{\text{grad}}(\mathbf{x}, \mathbf{n}, \phi) = \alpha \mathbf{n} \cdot \nabla \|\nabla V_g\|, \quad (8.10)$$

where α is a scaling factor for the *image-based* feature attraction term $\nabla \|\nabla V_g\|$ and $\mathbf{n}$ is the normal to the level set surface at $\mathbf{x}$. V_g symbolizes some global uniform merging of the multiple nonuniform input volumes. This feature term is effectively a 3D directional edge detector of V_g . However, there are two problems associated with using this speed function exclusively. The first is that we cannot expect to compute reliable 3D directional edge information in all regions of space simply because of the nature of the nonuniform input volumes. In

other words, V_g cannot be interpolated reliably in regions of space where there are no nearby sample points. Hence the level set surface will not experience any image-based forces in these regions. The solution is to use a regularization term that imposes constraints on the mean curvature of the deforming level set surface. We include the smoothing term from Eq. (8.6) and scale it with parameter β , in order to smooth the regions where no edge information exists as well as suppress noise in the remaining regions, thereby preventing excessive aliasing.

Normally the feature attraction term, $\nabla\|\nabla V_g\|$, creates only a narrow range of influence. In other words, this feature attraction term will only reliably move the portion of the level set surface that is in close proximity to the actual edges in V_g . Thus, a good initialization of the level set surface is needed before solving the level set equation when using $\mathcal{F}_{\text{grad}}$ (Eq. (8.10)). A reasonable initialization of the level set surface may be obtained by computing the CSG union of the multiple input volumes, which are first trilinearly resampled to give a uniform sampling. However, if the input volumes are strongly nonuniform, i.e. they are severely undersampled in one or more directions, their union produces a poor initial model. To improve the initialization we attract the CSG union surface to the Canny edges [16] computed from V_g using the distance transform produced from those edges (see Eq. (8.7)). This approach allows us to move the initial surface from a long range, but only with pixel-level accuracy.

Canny edges are nondirectional edges defined from the zero-crossing of the second derivative of the image in the direction of the local normal. In 3D this is

$$\frac{\partial^2}{\partial \mathbf{n}_g^2} V_g = 0, \quad (8.11)$$

where $\mathbf{n}_g \equiv \nabla V_g / \|\nabla V_g\|$ is the local normal vector of V_g . Using the expression $\partial / \partial \mathbf{n}_g = \mathbf{n}_g \cdot \nabla$, we can rewrite Eq. (8.11) as

$$\frac{\partial^2}{\partial \mathbf{n}_g^2} V_g = \mathbf{n}_g \cdot \nabla [\mathbf{n}_g \cdot \nabla V_g] = \mathbf{n}_g \cdot \nabla \|\nabla V_g\|. \quad (8.12)$$

The next section focuses on the methods needed to reliably compute the vectors $\mathbf{n}_g$ and $\nabla\|\nabla V_g\|$. In preparation, the latter may be explicitly expressed in terms of the derivatives of the merged volume V_g ,

$$\nabla\|\nabla V_g\| = \frac{\nabla V_g \cdot \hat{H} V_g}{\|\nabla V_g\|}, \quad (8.13)$$

where we have defined the gradient vector and the Hessian matrix,

$$\widehat{\nabla} V_g = \left(\frac{\partial V_g}{\partial x}, \frac{\partial V_g}{\partial y}, \frac{\partial V_g}{\partial z} \right), \quad (8.14a)$$

$$\widehat{H} V_g = \begin{pmatrix} \frac{\partial^2 V_g}{\partial x^2} & \frac{\partial^2 V_g}{\partial y \partial x} & \frac{\partial^2 V_g}{\partial z \partial x} \\ \frac{\partial^2 V_g}{\partial x \partial y} & \frac{\partial^2 V_g}{\partial y^2} & \frac{\partial^2 V_g}{\partial z \partial y} \\ \frac{\partial^2 V_g}{\partial x \partial z} & \frac{\partial^2 V_g}{\partial y \partial z} & \frac{\partial^2 V_g}{\partial z^2} \end{pmatrix}. \quad (8.14b)$$

Thus, in closing we note that the level set propagation needed for segmentation only needs information about the first- and second-order partial derivatives of the input volumes, not the interpolated intensity values themselves.

8.4.1.2 Computing Partial Derivatives

As outlined above, the speed function $\mathcal{F}$ in the level set equation, Eq. (8.4), is based on edge information derived from the input volumes. This requires estimating first- and second-order partial derivatives from the multiple nonuniform input volumes. We do this by means of moving least-squares (MLS), which is an effective and well-established numerical technique for computing derivatives of functions whose values are known only on irregularly spaced points [44–46].

Let us assume we are given the input volumes $\widehat{V}_d$, $d = 1, 2, \dots, D$, which are volumetric samplings of an object on the nonuniform grids $\{\widehat{\mathbf{x}}_d\}$. We shall also assume that the local coordinate frames of $\{\widehat{\mathbf{x}}_d\}$ are scaled, rotated, and translated with respect to each other. Hence, we define a world coordinate frame (typically one of the local frames) in which we solve the level set equation. Now, let us define the world sample points $\{\mathbf{x}_d\}$ as

$$\mathbf{x}_d \equiv \mathbf{T}^{(d)}[\widehat{\mathbf{x}}_d], \quad (8.15)$$

where $\mathbf{T}^{(d)}$ is the coordinate transformation from a local frame d to the world frame. Next we locally approximate the intensity values from the input volumes $\widehat{V}_d$ with a 3D polynomial expansion. Thus, we define the N -order polynomials

$$V_N^{(d)}(\mathbf{x}) = C_{000}^{(d)} + \sum_{i+j+k=1}^N C_{ijk}^{(0)} x^i y^j z^k, \quad d = 1, 2, \dots, D, \quad (8.16)$$

where the coefficients C are unknown. Note that these local approximations

to the intensity values share coefficients $C_{ijk}^{(0)}$ of order higher than zero, i.e., all of the functions $V_N^{(d)}$, $d = 1, 2, \dots, D$ have the *same edges*. The fact that the zero-order term in Eq. (8.16) is input volume dependent means we allow for local constant offsets between the input volumes $\widehat{V}_d$. This effectively provides built-in gain correction in the scheme, since it can handle small intensity attenuation artifacts between the multiple scans.

Moving least-squares: To solve for the expansion coefficients C in Eq. (8.16) we define the moving least-squares functional

$$E(\mathbf{x}_0) = \sum_{d=1}^D \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[V_N^{(d)}(\mathbf{x}_d - \mathbf{x}_0) - V_d(\mathbf{x}_d) \right]^2, \quad (8.17)$$

where $\mathbf{x}_0$ is the expansion point from where we are seeking edge information, $V_d(\mathbf{x}_d) \equiv \widehat{V}_d(\widehat{\mathbf{x}}_d)$ and where

$$w_d(\mathbf{x}) \equiv \begin{cases} 1 - 2(\|\mathbf{x}\|/\Delta)^2 & \text{for } 0 \leq \|\mathbf{x}\| \leq \Delta/2 \\ 2(\|\mathbf{x}\|/\Delta - 1)^2 & \text{for } \Delta/2 < \|\mathbf{x}\| < \Delta \\ 0 & \text{for } \|\mathbf{x}\| \geq \Delta \end{cases} \quad (8.18)$$

is a “moving filter” that weights the contribution of different sampling points, $\mathbf{x}_d$, according to their Euclidean distance, $\|\mathbf{x}_d - \mathbf{x}_0\|$, to the expansion point, $\mathbf{x}_0$. Other expressions for this weighting function could, of course, be used, but Eq. (8.18) is fast to compute, has finite support (by the window parameter Δ), and its tangent is zero at the endpoints. After substitution of Eq. (8.16) into Eq. (8.35) we obtain the functional

$$\begin{aligned} E(\mathbf{x}_0) = & \sum_{d=1}^D \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[C_{000}^{(d)} - \widehat{V}_d(\mathbf{x}_d) \right. \\ & \left. + \sum_{i+j+k=1}^N C_{ijk}^{(0)} (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \right]^2. \end{aligned} \quad (8.19)$$

The minimization of this moving least-squares functional with respect to the expansion coefficients C requires the partial derivatives to vanish, i.e.,

$$\begin{aligned} \frac{\partial \widehat{E}(\mathbf{x}_0)}{\partial C_{000}^{(d)}} = 0 = & 2 \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[C_{000}^{(d)} - \widehat{V}_d(\mathbf{x}_d) \right. \\ & \left. + \sum_{i+j+k=1}^N C_{ijk}^{(0)} (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \right], \end{aligned} \quad (8.20a)$$

$$\begin{aligned}
\frac{\partial \widehat{E}(\mathbf{x}_0)}{\partial C_{lmn}^{(0)}} = 0 &= 2 \sum_{d=1}^D \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \left[C_{000}^{(d)} - \widehat{V}_d(\mathbf{x}_d) \right. \\
&+ \sum_{i+j+k=1}^N C_{ijk}^{(0)} (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k \left. \right] \\
&\times (x_d - x_0)^l (y_d - y_0)^m (z_d - z_0)^n.
\end{aligned} \tag{8.20b}$$

This defines a system of linear equations in the expansion coefficients $C_{ijk}^{(r)}$ that can be solved using standard techniques from numerical analysis, see Eqs. (8.21) and (8.23).

Equations (8.20a) and (8.20b) can then be conveniently expressed as

$$\sum_{\mathbf{q}} \mathbf{A}_{\mathbf{p},\mathbf{q}} \mathbf{c}_{\mathbf{q}} = \mathbf{b}_{\mathbf{p}}, \tag{8.21}$$

where $\mathbf{A}$ is a diagonal matrix, and $\mathbf{b}$, $\mathbf{c}$ are vectors. In this equation we have also introduced the compact index notations $\mathbf{p} \equiv (i, j, k, r)$ and $\mathbf{q} \equiv (l, m, n, s)$ defined as

$$\begin{aligned}
\mathbf{p} \in \{ &i, j, k, r \in \mathcal{N}^+ \mid i = j = k = 0, 1 \leq r \leq D \} \\
&\cup \{ i, j, k, r \in \mathcal{N}^+ \mid 1 \leq i+j+k \leq N, r = 0 \},
\end{aligned} \tag{8.22a}$$

$$\begin{aligned}
\mathbf{q} \in \{ &l, m, n, s \in \mathcal{N}^+ \mid l = m = n = 0, 1 \leq s \leq D \} \\
&\cup \{ l, m, n, s \in \mathcal{N}^+ \mid 1 \leq l+m+n \leq N, s = 0 \}.
\end{aligned} \tag{8.22b}$$

The diagonal matrix $\mathbf{A}$ and the vectors $\mathbf{b}$, $\mathbf{c}$ in Eq. (8.21) are defined as

$$\begin{aligned}
\mathbf{A}_{\mathbf{p},\mathbf{q}} \equiv \sum_d (\delta_{r,d} + \delta_{r,0}) (\delta_{s,d} + \delta_{s,0}) \sum_{\mathbf{x}_d} w_d(\mathbf{x}_d - \mathbf{x}_0) \\
\times (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k
\end{aligned} \tag{8.23a}$$

$$\begin{aligned}
&\times (x_d - x_0)^l (y_d - y_0)^m (z_d - z_0)^n, \\
\mathbf{b}_{\mathbf{p}} \equiv \sum_d (\delta_{r,d} + \delta_{r,0}) w_d(\mathbf{x}_d - \mathbf{x}_0) \widehat{V}_d(\mathbf{x}_d) \\
&\times (x_d - x_0)^i (y_d - y_0)^j (z_d - z_0)^k,
\end{aligned} \tag{8.23b}$$

$$\mathbf{c}_{\mathbf{p}} \equiv C_{ijk}^{(r)}. \tag{8.23c}$$

Next the matrix equation $\mathbf{A}\mathbf{c} = \mathbf{b}$ must be solved for the vector $\mathbf{c}$ of dimension $\binom{N+3}{3} + D - 1$, where N is the order of the expansion in Eq. (8.16) and D is the number of nonuniform input volumes. As is well known for many moving least-square problems, it is possible for the condition number of the matrix $\mathbf{A}$ to become very large. Any matrix is singular if its condition number is infinite

and can be defined as ill-conditioned if the reciprocal of its condition number approaches the computer's floating-point precision. This can occur if the problem is overdetermined (number of sample points, $\mathbf{x}_d$ greater than number of coefficients C) and underdetermined (ambiguous combinations of the coefficients C work equally well or equally bad). To avoid such numerical problems, a singular value decomposition (SVD) linear equation solver is recommended for use in combination with the moving least-squares method. The SVD solver identifies equations in the matrix $\mathbf{A}$ that are, within a specified tolerance, redundant (i.e., linear combinations of the remaining equations) and eliminates them thereby improving the condition number of the matrix. We refer the reader to [54] for a helpful discussion of SVD pertinent to linear least-squares problems.

Once we have the expansion coefficients $\mathbf{c}$, we can readily express the Hessian matrix and the gradient vector of the combined input volumes as

$$\nabla V = \left(C_{100}^{(0)}, C_{010}^{(0)}, C_{001}^{(0)} \right), \quad (8.24a)$$

$$\mathbf{H}V = \begin{pmatrix} 2C_{200}^{(0)} & C_{110}^{(0)} & C_{101}^{(0)} \\ C_{110}^{(0)} & 2C_{020}^{(0)} & C_{011}^{(0)} \\ C_{101}^{(0)} & C_{011}^{(0)} & 2C_{002}^{(0)} \end{pmatrix} \quad (8.24b)$$

evaluated at the moving expansion point $\mathbf{x}_0$. This in turn is used in Eq. (8.13) to compute the edge information needed to drive the level set surface.

8.4.1.3 Algorithm Overview

Algorithm 1 describes the main steps of our approach. The initialization routine, Algorithm 2, is called for all of the multiple nonuniform input volumes, V_d . Each nonuniform input dataset is uniformly resampled in a common coordinate frame (V_0 's) using trilinear interpolation. Edge information and the union, V_0 , of the V_d 's are then computed. Algorithm 1 calculates Canny and 3D directional edge information using moving least-squares in a narrow band in each of the resampled input volumes, V_d , and buffers this in V_{edge} and V_{grad} . Next Algorithm 1 computes the distance transform of the zero-crossings of the Canny edges and takes the gradient of this scalar volume to produce a vector field V_{edge} , which pulls the initial level set model to the Canny edges. Finally the level set model is attracted to the 3D directional edges of the multiple input volumes, V_{grad} , and a Marching Cubes mesh is extracted for visualization. The level set

solver, described in Algorithm 3, solves Eq. (8.4) using the “up-wind scheme” (not explicitly defined) and the sparse-field narrow-band method of [36], with V_0 as the initialization and V_{edge} and V_{grad} as the force field in the speed function.

Algorithm 1: $\text{MAIN}(V_1, \dots, V_D)$

comment: $V_1, \dots, V_D$ are nonuniform samplings of object V

global $V_{\text{edge}}, V_{\text{grad}}$

```

do {
   $V_0 \leftarrow$  uniform sampling of empty space
  for  $d \leftarrow 1$  to  $D$ 
    do  $V_0 \leftarrow V_0 \cup \text{INITIALIZATION}(V_d)$ 
   $V_{\text{edge}} \leftarrow \nabla [\text{distance transform}[\text{zero-crossing}[V_{\text{edge}}]]]$ 
   $V_0 \leftarrow \text{SOLVELEVELSETEQ}(V_0, V_{\text{edge}}, \alpha, 0)$ 
   $V_0 \leftarrow \text{SOLVELEVELSETEQ}(V_0, V_{\text{grad}}, \alpha, \beta)$ 
return (Marching Cubes mesh of  $V_0$ )

```

Algorithm 2: $\text{INITIALIZATION}(V_d)$

comment: Preprocessing to produce good LS initialization

```

do {
   $V_d \leftarrow$  Uniform trilinear resampling of  $V_d$ 
   $\Gamma_d \leftarrow$  Set of voxels in narrow band of isosurface of  $V_d$ 
  for each “unprocessed”  $\mathbf{x}_0 \in \Gamma_d$ 
    do {
      Solve moving least-squares problem at  $\mathbf{x}_0$ 
      do {
         $V_{\text{edge}}(\mathbf{x}_0) \leftarrow$  scalar Canny edge, cf. Equation (8.12)
         $V_{\text{grad}}(\mathbf{x}_0) \leftarrow$  3D directional edge, cf. Equation (8.13)
      }
    }
return ( $V_d$ )

```

Algorithm 3: $\text{SOLVELEVELSETEQ}(V_0, \mathbf{V}, \alpha, \beta)$

comment: Solve Equation (8.4) with initial condition $\phi(t=0) = V_0$

```

do {
   $\phi \leftarrow V_0$ 
  repeat
     $\Gamma \leftarrow$  Set of voxels in narrow band of isosurface of  $\phi$ 
     $\Delta t \leftarrow \gamma / \sup_{\mathbf{x} \in \Gamma} \|\mathbf{V}(\mathbf{x})\|$ ,  $\gamma \leq 1$ 
    for each  $\mathbf{x} \in \Gamma$ 
      do {
         $\mathbf{n} \leftarrow$  upwind scheme  $[-\nabla\phi(\mathbf{x})/\|\nabla\phi(\mathbf{x})\|]$ 
         $\dot{\phi}(\mathbf{x}) \leftarrow \|\nabla\phi(\mathbf{x})\|(\alpha \mathbf{V}(\mathbf{x}) \cdot \mathbf{n} + \beta \nabla \cdot \mathbf{n})$ 
         $\phi(\mathbf{x}) \leftarrow \phi(\mathbf{x}) + \dot{\phi}(\mathbf{x})\Delta t$ 
      }
    until  $\sup_{\mathbf{x} \in \Gamma} \|\dot{\phi}(\mathbf{x})\| \leq \epsilon$ 
return ( $\phi$ )

```

Table 8.2: Maximum in-plane to out-of-plane sampling ratios of nonuniform input datasets, and parameters for the two level set speed terms defined in Eqs. (8.6) and (8.10)

Model	Origin	Ratio	α	β
Griffin	Laser scan	6/10:1	1.0	0.5
Mouse	MR scan	10:1	1.0	0.5
Zucchini	MR scan	10:1	1.0	0.5

8.4.2 Multiple Volume Results

We have applied our segmentation method to several multiscan nonuniform datasets to produce high-resolution level set models. The parameters used for these segmentations are listed in Table 8.2. α and β are weights that the user adjusts to balance attraction to edges with curvature-based smoothing during the level set deformation process.

8.4.2.1 Griffin Dataset

The griffin dataset was created with a volumetric laser scan reconstruction algorithm [49]. This algorithm creates a high-resolution volumetric representation of an object by merging multiple depth maps produced via a laser scan. The original griffin dataset has a resolution of $312 \times 294 \times 144$. We have extracted two nonuniform datasets from this high-resolution representation by copying every sixth plane of data in the X direction and every tenth plane in the Y direction. The two derived nonuniform griffin datasets have the following resolution: $52 \times 294 \times 144$ and $312 \times 30 \times 144$. Isosurfaces have been extracted from these datasets, appropriately scaled in the low-resolution direction, and are presented in the first two images in Fig. 8.9 (top). Each low-resolution scan inadequately captures some important geometric feature of the griffin. We have performed a reconstruction from the undersampled nonuniform scans to produce the result in Fig. 8.9 (top). The method produces a high-resolution ($312 \times 294 \times 144$) level set model that contains all of the significant features of the original scan.

8.4.2.2 Mouse Embryo Dataset

The first three scans in Fig. 8.9 (bottom) are derived from a high-resolution MR scan of a mouse embryo. They are subsampled versions of a $256 \times 128 \times 128$

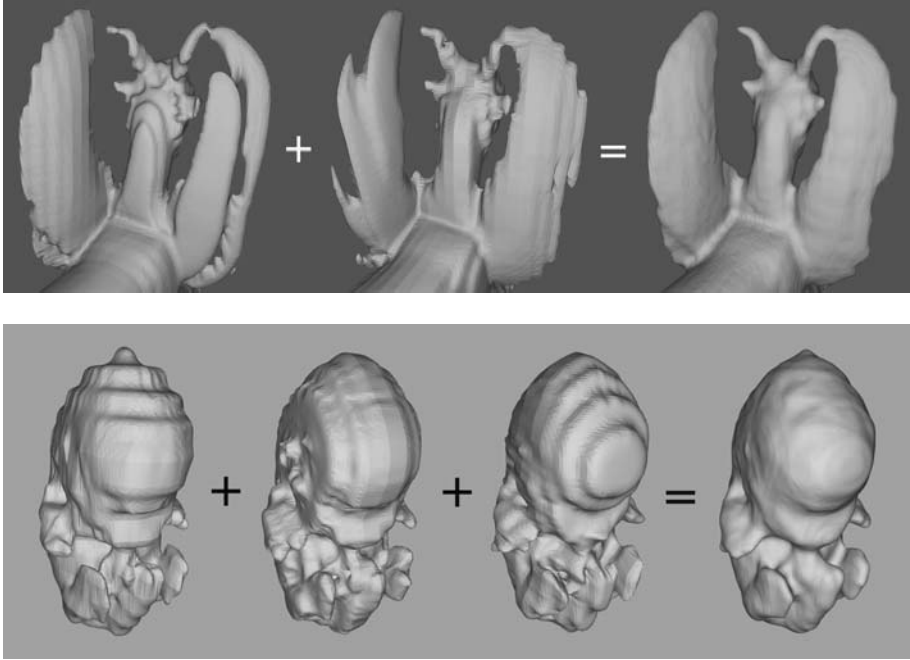


Figure 8.9: Nonuniform datasets merged to produce high-resolution level set models, (top) laser scan of a figurine and (bottom) MR scan of a mouse embryo.

volume dataset, and have the following resolutions: $26 \times 128 \times 128$, $256 \times 16 \times 128$, and $256 \times 128 \times 13$. The last image in Fig. 8.9 presents the result produced by our multiscan segmentation method. The information in the first three scans has been successfully used to create a level set model of the embryo with a resolution of $256 \times 128 \times 130$. The finer features of the mouse embryo, namely its hands and feet, have been reconstructed.

8.4.2.3 Zucchini Dataset

The final dataset consists of three individual MRI scans of an actual zucchini. The separate scans have been registered manually and are presented on the left side of Fig. 8.10, each with a different color. The resolutions of the individual scans are $28 \times 218 \times 188$, $244 \times 25 \times 188$, and $244 \times 218 \times 21$. This image highlights the rough alignment of the scans. The right side of Fig. 8.10 presents the result of our level set segmentation. It demonstrates that our approach is able to extract a reasonable model from multiple datasets that are imperfectly aligned.



Figure 8.10: Three low-resolution MR scans of a zucchini that have been individually colored and overlaid to demonstrate their imperfect alignment. The level set model on the right is derived from the three low-resolution scans.

8.5 Segmentation of DT-MRI Brain Data

Diffusion tensor magnetic resonance imaging [55, 56] (DT-MRI) is a technique used to measure the diffusion properties of water molecules in tissues. Anisotropic diffusion can be described by the equation

$$\frac{\partial C}{\partial t} = \nabla \cdot (d \nabla C), \quad (8.25)$$

where C is the concentration of water molecules and d is a diffusion coefficient, which is a symmetric second-order tensor

$$d = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix}. \quad (8.26)$$

Figure 8.11 presents a “slice” of the diffusion tensor volume data of human brain used in our study. Each subimage presents the scalar values of the associated diffusion tensor component for one slice of the dataset.

Tissue segmentation and classification based on DT-MRI offers several advantages over conventional MRI, since diffusion data contains additional physical information about the internal structure of the tissue being scanned. However, segmentation and visualization using diffusion data is not entirely straightforward. First of all, the diffusion matrix itself is not invariant with respect to rotations, and the elements that form the matrix will be different for different

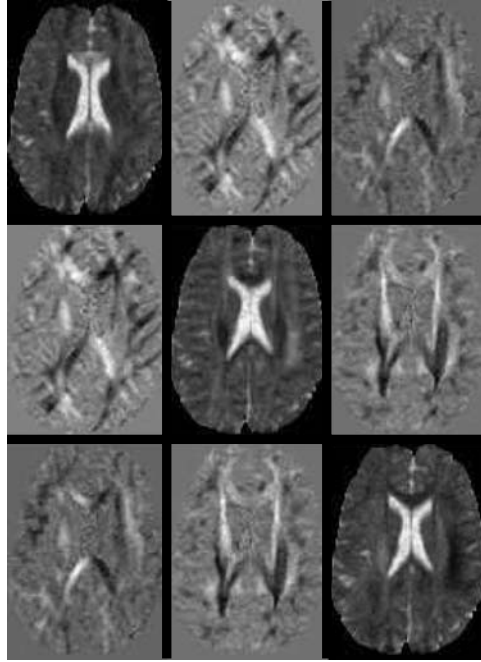


Figure 8.11: Slice of a tensor volume where every “element” of the image matrix corresponds to one component of the tensor D .

orientations of the sample or field gradient and therefore cannot themselves be used for classification purposes. Moreover, 3D visualization and segmentation techniques available today are predominantly designed for scalar and sometimes vector fields. Thus, there are three fundamental problems in tensor imaging: (a) finding an invariant representation of a tensor that is independent of a frame of reference, (b) constructing a mapping from the tensor field to a scalar or vector field, and (c) visualization and classification of tissue using the derived scalar fields.

The traditional approaches to diffusion tensor imaging involve converting the tensors into an eigenvalue/eigenvector representation, which is rotationally invariant. Every tensor may then be interpreted as an ellipsoid with principal axes oriented along the eigenvectors and radii equal to the corresponding eigenvalues. This ellipsoid describes the probabilistic distribution of a water molecule after a fixed diffusion time.

Using eigenvalues/eigenvectors, one can compute different anisotropy measures [55, 57–59] that map tensor data onto scalars and can be used for further

visualization and segmentation. Although eigenvalue/vector computation of the 3×3 matrix is not expensive, it must be repeatedly performed for every voxel in the volume. This calculation easily becomes a bottleneck for large datasets. For example, computing eigenvalues and eigenvectors for a 512^3 volume requires over 20 CPU min on a powerful workstation. Another problem associated with eigenvalue computation is stability—a small amount of noise will change not only the values but also the ordering of the eigenvalues [60]. Since many anisotropy measures depend on the ordering of the eigenvalues, the calculated direction of diffusion and classification of tissue will be significantly altered by the noise normally found in diffusion tensor datasets. Thus it is desirable to have an anisotropy measure which is rotationally invariant, does not require eigenvalue computations, and is stable with respect to noise. Tensor invariants with these characteristics were first proposed by Ulug *et al.* [61]. In Section 8.5.1 we formulate a new anisotropy measure for tensor field based on these invariants.

Visualization and model extraction from the invariant 3D scalar fields is the second issue addressed in this chapter. One of the popular approaches to tensor visualization represents a tensor field by drawing ellipsoids associated with the eigenvectors/values [62]. This method was developed for 2D slices and creates visual cluttering when used in 3D. Other standard CFD visualization techniques such as tensor-lines do not provide meaningful results for the MRI data due to rapidly changing directions and magnitudes of eigenvector/values and the amount of noise present in the data. Recently Kindlmann [63] developed a volume rendering approach to tensor field visualization using eigenvalue-based anisotropy measures to construct transfer functions and color maps that highlight some brain structures and diffusion patterns.

In our work we perform isosurfacing on the 3D scalar fields derived from our tensor invariants to visualize and segment the data [64]. An advantage of isosurfacing over other approaches is that it can provide the shape information needed for constructing geometric models, and computing internal volumes and external surface areas of the extracted regions. There has also been a number of recent publications [65, 66] devoted to brain fiber tracking. This is a different and more complex task than the one addressed in this chapter and requires data with a much higher resolution and better signal-to-noise ratio than the data used in our study.

8.5.1 Tensor Invariants

Tensor invariants (rotational invariants) are combinations of tensor elements that do not change after the rotation of the tensor's frame of reference, and thus do not depend on the orientation of the patient with respect to the scanner when performing DT imaging. The well-known invariants are the eigenvalues of the diffusion tensor (matrix) d , which are the roots of the corresponding characteristic equation

$$\lambda^3 - C_1 \cdot \lambda^2 + C_2 \cdot \lambda - C_3 = 0, \quad (8.27)$$

with coefficients

$$\begin{aligned} C_1 &= D_{xx} + D_{yy} + D_{zz} \\ C_2 &= D_{xx}D_{yy} - D_{xy}D_{yx} + D_{xx}D_{zz} - D_{xz}D_{zx} + D_{yy}D_{zz} - D_{yz}D_{zy} \\ C_3 &= D_{xx}(D_{yy}D_{zz} - D_{zy}D_{yz}) \\ &\quad - D_{xy}(D_{yx}D_{zz} - D_{zx}D_{yz}) + D_{xz}(D_{yx}D_{zy} - D_{zx}D_{yy}). \end{aligned} \quad (8.28)$$

Since the roots of Eq. (8.27) are rotational invariants, the coefficients C_1 , C_2 , and C_3 are also invariant. In the eigen-frame of reference they can be easily expressed through the eigenvalues

$$\begin{aligned} C_1 &= \lambda_1 + \lambda_2 + \lambda_3 \\ C_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 \\ C_3 &= \lambda_1\lambda_2\lambda_3 \end{aligned} \quad (8.29)$$

and are proportional to the sum of the radii, surface area, and the volume of the “diffusion” ellipsoid. Then instead of using $(\lambda_1, \lambda_2, \lambda_3)$ to describe the dataset, we can use (C_1, C_2, C_3) . Moreover, since C_i are the coefficients of the characteristic equation, they are less sensitive to noise than are the roots λ_i of the same equation.

Any combination of the above invariants is, in turn, an invariant. We consider the following dimensionless combination: C_1C_2/C_3 . In the eigenvector frame of reference, it becomes

$$\frac{C_1C_2}{C_3} = 3 + \frac{\lambda_2 + \lambda_3}{\lambda_1} + \frac{\lambda_1 + \lambda_3}{\lambda_2} + \frac{\lambda_1 + \lambda_2}{\lambda_3} \quad (8.30)$$

and we can define a new dimensionless anisotropy measure

$$C_a = \frac{1}{6} \left[\frac{C_1C_2}{C_3} - 3 \right]. \quad (8.31)$$

It is easy to show that for isotropic diffusion, when $\lambda_1 = \lambda_2 = \lambda_3$, the coefficient $C_a = 1$. In the anisotropic case, this measure is identical for both linear, directional diffusion ($\lambda_1 \gg \lambda_2 \approx \lambda_3$) and planar diffusion ($\lambda_1 \approx \lambda_2 \gg \lambda_3$) and is equal to

$$C_a^{\text{limit}} \approx \frac{1}{3} \left[1 + \frac{\lambda_1}{\lambda_3} + \frac{\lambda_3}{\lambda_1} \right]. \quad (8.32)$$

Thus C_a is always $\sim \lambda_{\text{max}}/\lambda_{\text{min}}$ and measures the magnitude of the diffusion anisotropy. We again want to emphasize that we use the eigenvalue representation here only to analyze the behavior of the coefficient C_a , but we use invariants (C_1, C_2, C_3) to compute it using Eqs. (8.28) and (8.31).

8.5.2 Geometric Modeling

Two options are usually available for viewing the scalar volume datasets, direct volume rendering [1, 4] and volume segmentation [67] combined with conventional surface rendering. The first option, direct volume rendering, is only capable of supplying images of the data. While this method may provide useful views of the data, it is well known that it is difficult to construct the exact transfer function that highlights the desired structures in the volume dataset [68]. Our approach instead focuses on extracting geometric models of the structures embedded in the volume datasets. The extracted models may be used for interactive viewing, but the segmentation of geometric models from the volume datasets provides a wealth of additional benefits and possibilities. The models may be used for quantitative analysis of the segmented structures, for example the calculation of surface area and volume, quantities that are important when studying how these structures change over time. The models may be used to provide the shape information necessary for anatomical studies and computational simulation, for example EEG/MEG modeling within the brain [69]. Creating separate geometric models for each structure allows for the straightforward study of the relationship between the structures, even though they come from different datasets. The models may also be used within a surgical planning/simulation/VR environment [70], providing the shape information needed for collision detection and force calculations. The geometric models may even be used for manufacturing real physical models of the structures [71]. It is clear that there are numerous reasons to develop techniques for extracting geometric models from diffusion tensor volume datasets.

The most widely used technique for extracting polygonal models from volume datasets is the Marching Cubes algorithm [72]. This technique creates a polygonal model that approximates the isosurface embedded in a scalar volume dataset for a particular isovalue. While the Marching Cubes algorithm is easy to understand and straightforward to implement, applying it directly to raw volume data from scanners can produce undesirable results, as seen in the first images in Figs. 8.13 and 8.16. The algorithm is susceptible to noise and can produce many unwanted triangles that mask the central structures in the data. In order to alleviate this problem, we utilize the tools in our level set framework to smooth the data and remove the noise-related artifacts.

8.5.3 Segmentation

In this section we demonstrate the application of our methods to the segmentation of DT-MRI data of the human head. We use a high-resolution dataset from a human volunteer which contains 60 slices each of 128×128 pixels resolution. The raw data is sampled on a regular uniform grid.

We begin by generating two scalar volume datasets based on the invariants described in Section 8.5.1. The first scalar volume dataset ($\mathcal{V}_1$) is formed by calculating the trace (C_1) of the tensor matrix for each voxel of the diffusion tensor volume. It provides a single number that characterizes the total diffusivity at each voxel within the sample. Higher values signify greater total diffusion irrespective of directionality in the region represented by a particular voxel. A slice from this volume can be seen in Fig. 8.12 (left). The second scalar volume

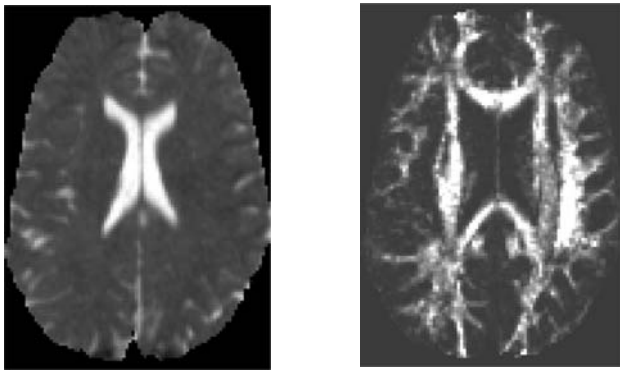


Figure 8.12: Isotropic C_1 (left) and anisotropic C_a (right) tensor invariants for the tensor slice shown in Fig. 8.11.

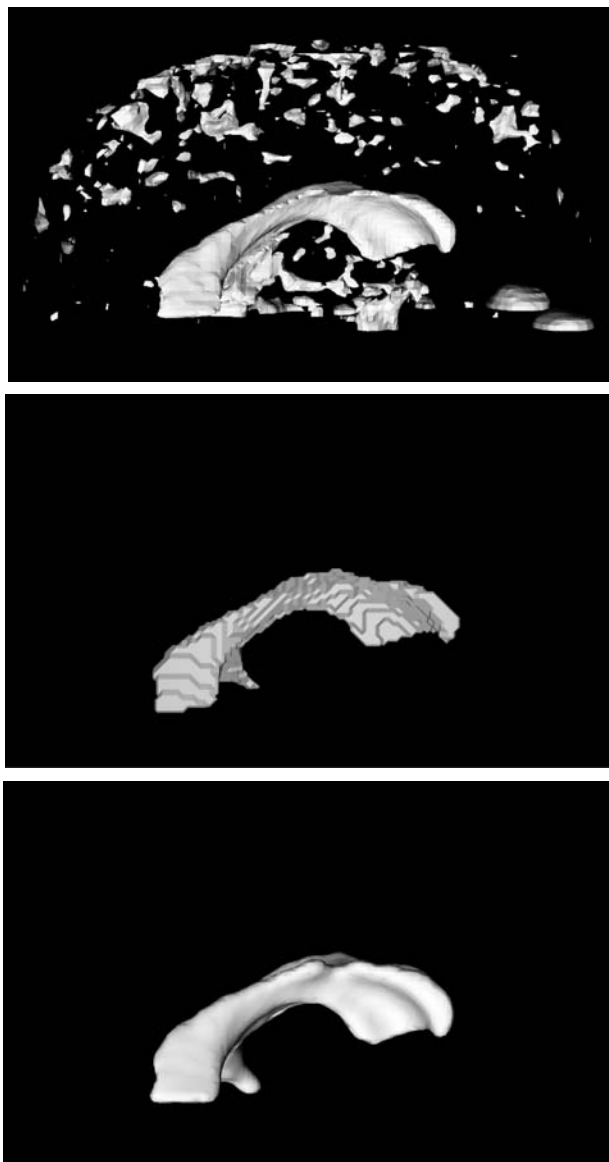


Figure 8.13: Segmentation from isotropic measure volume $\mathcal{V}_1$ for the first DT-MRI dataset. The first row is the Marching Cubes isosurface with iso-value 7.5. The second row is the result of flood-fill algorithm applied to the same volume and used for initialization. The third row is the final level set model.

dataset ($\mathcal{V}_2$) is formed by calculating (C_1, C_2, C_3) invariants for each voxel and combining them into C_a . It provides a measure of the magnitude of the anisotropy within the volume. Higher values identify regions of greater spatial anisotropy in the diffusion properties. A slice from the second scalar volume is presented in Fig. 8.12 (right). The measure C_a does not by definition distinguish between linear and planar anisotropy. This is sufficient for our current study since the brain does not contain measurable regions with planar diffusion anisotropy. We therefore only need two scalar volumes in order to segment the DT dataset.

We then utilize our level set framework to extract smoothed models from the two derived scalar volumes. First the input data is filtered with a low-pass Gaussian filter ($\sigma \approx 0.5$) to blur the data and thereby reduce noise. Next, the volume voxels are classified for inclusion/exclusion in the initialization based on the filtered values of the input data ($k \approx 7.0$ for $\mathcal{V}_1$ and $k \approx 1.3$ for $\mathcal{V}_2$). For grayscale images, such as those used in this chapter, the classification is equivalent to high and low thresholding operations. The last initialization step consists of performing a set of topological (e.g. flood fill) operations in order to remove small pieces or holes from objects. This is followed by a level set deformation that pulls the surface toward local maxima of the gradient magnitude and smooths it with a curvature-based motion. This moves the surface toward specific features in the data, while minimizing the influence of noise in the data.

Figures 8.13 and 8.14 present two models that we extracted from DT-MRI volume datasets using our techniques. Figure 8.13 contains segmentations from volume $\mathcal{V}_1$, the measure of total diffusivity. The top image shows a Marching Cubes isosurface using an isovalue of 7.5. In the bottom we have extracted just the ventricles from $\mathcal{V}_1$. This is accomplished by creating an initial model with a flood-fill operation inside the ventricle structure shown in the middle image. This identified the connected voxels with value of 7.0 or greater. The initial model was then refined and smoothed with a level set deformation, using a β value of 0.2.

Figure 8.14 again provides the comparison between direct isosurfacing and level set modeling, but on the volume $\mathcal{V}_2$. The image in the top-left corner is a Marching Cubes isosurface using an isovalue of 1.3. There is significant high-frequency noise and features in this dataset. The challenge here was to isolate coherent regions of high anisotropic diffusion. We applied our segmentation approach to the dataset and worked with neuroscientists from LA Childrens

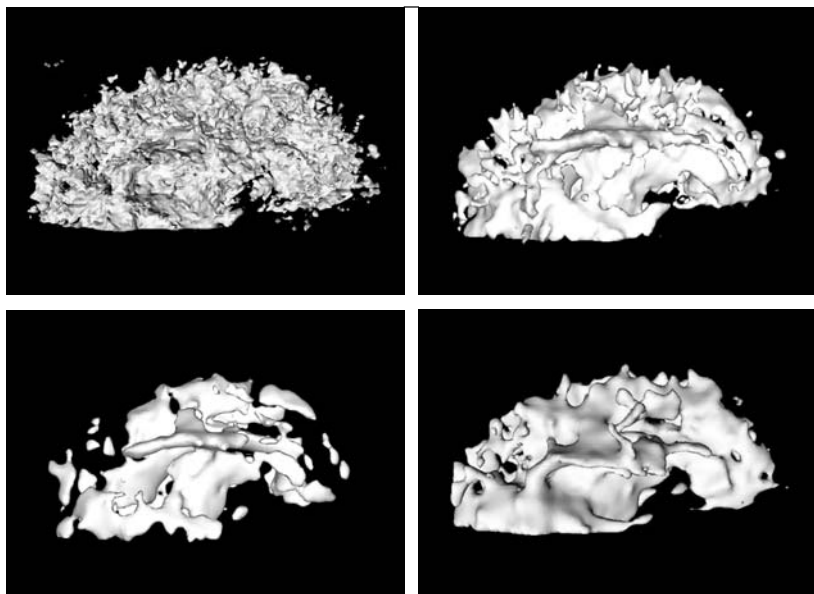


Figure 8.14: Model segmentation from volume $\mathcal{V}_2$. Top left image is an isosurface of value 1.3, used for initialization of the level set. Clockwise are the results of level set development with corresponding β values of 0.2, 0.4, and 0.5.

Hospital, City of Hope Hospital and Caltech to identify meaningful anatomical structures. We applied our approach using a variety of parameter values, and presented our results to them, asking them to pick the model that they felt best represented the structures of the brain. Figure 8.14 contains three models extracted from $\mathcal{V}_2$ at different values of smoothing parameter β used during segmentation. Since we were not looking for a single connected structure in this volume, we did not use a seeded flood-fill for initialization. Instead, we initialized the deformation process with an isosurface of value 1.3. This was followed by a level set deformation using a β value of 0.2. The result of this segmentation is presented on the bottom-left side of Fig. 8.14. The top-right side of this figure presents a model extracted from $\mathcal{V}_2$ using an initial isosurface of value 1.4 and a β value of 0.5. The result chosen as the “best” by our scientific/medical collaborators is presented on the bottom-right side of Fig. 8.14. This model is produced with an initial isosurface of 1.3 and a β value of 0.4. Our collaborators were able to identify structures of high diffusivity in this model, for example the corpus callosum, the internal capsule, the optical nerve tracks, and other white matter regions.

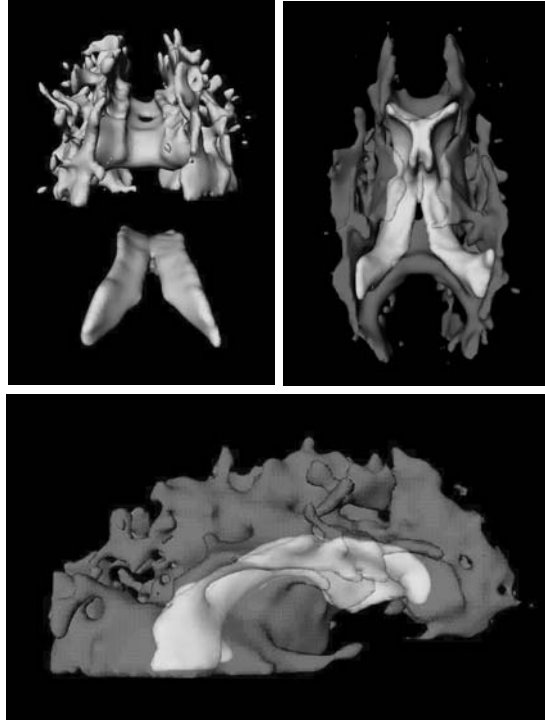


Figure 8.15: Combined model of ventricles and (semi-transparent) anisotropic regions: rear, exploded view (left), bottom view (right), side view (bottom). Note how model of ventricles extracted from isotropic measure dataset $\mathcal{V}_1$ fits into model extracted from anisotropic measure dataset $\mathcal{V}_2$.

We can also bring together the two models extracted from datasets $\mathcal{V}_1$ and $\mathcal{V}_2$ into a single image. They will have perfect alignment since they are derived from the same DT-MRI dataset. Figure 8.15 demonstrates that we are able to isolate different structures in the brain from a single DT-MRI scan and show their proper spatial interrelationship. For example, it can be seen that the corpus callosum lies directly on top of the ventricles, and that the white matter fans out from both sides of the ventricles.

Finally, to verify the validity of our approach we applied it to the second dataset from a different volunteer. This dataset has 20 slices of the 256×256 resolution. We generated the anisotropy measure volume $\mathcal{V}_2$ and performed the level set model extraction using the same isovalues and smoothing parameters as for $\mathcal{V}_2$. The results are shown in Fig. 8.16, and demonstrate the generality of our approach.

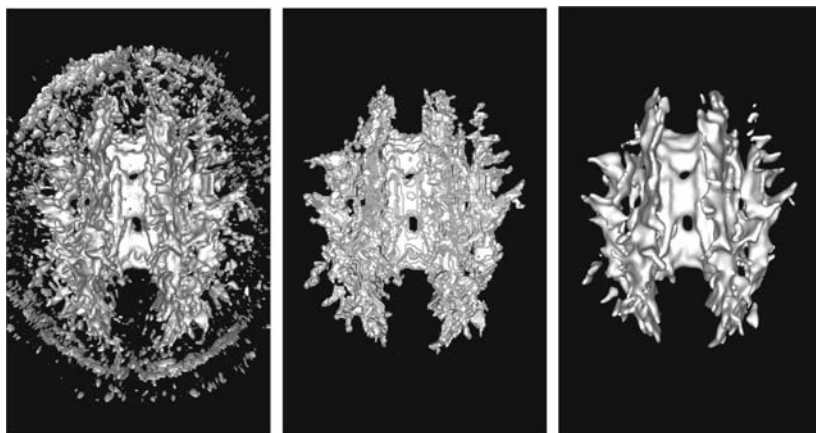


Figure 8.16: Segmentation using anisotropic measure $\mathcal{V}_2$ from the second DT-MRI dataset. (left) Marching Cubes isosurface with iso-value 1.3. (middle) Result of flood-fill algorithm applied to the volume and used for initialization. (right) Final level set model.

8.6 Direct Estimation of Surfaces in Tomographic Data

The radon transform is invertible (albeit, marginally so) when the measured data consists of a sufficient number of good quality, properly spaced projections [73]. However, for many applications the number of realizable projections is insufficient, and direct grayscale reconstructions are susceptible to artifacts. We will refer to such problems as *underconstrained* tomographic problems. Cases of underconstrained tomographic problems usually fall into one of two classes. The first class is where the measuring device produces a relatively dense set of projections (i.e. adequately spaced) that do not span a full 180° . In these cases, the sinogram contains regions without measured data. Considering the radon transform in the Fourier domain, these missing regions of the sinogram correspond to a transform with angular wedges (pie slices) that are null, making the transform noninvertible. We assume that these missing regions are large enough to preclude any straightforward interpolation in the frequency domain. The second class of incomplete tomographic problems are those that consist of an insufficient number of widely spaced projections. We assume that these sparse samples of the sinogram space are well distributed over a wide range of angles. For this discussion the precise spacing is not important. This problem

is characterized by very little data in the Fourier domain, and direct inversion approaches produce severe artifacts. Difficulties in reconstructing volumes from such incomplete tomographic datasets are often aggravated by noise in the measurements and misalignments among projections.

Under-constrained problems are typically solved using one or both of two different strategies. The first strategy is to choose from among feasible solutions (those that match the data) by imposing some additional criterion, such as finding the solution that minimizes an energy function. This additional criterion should be designed to capture some desirable property, such as minimum entropy. The second strategy is to parameterize the solution in a way that reduces the number of degrees of freedom. Normally, the model should contain few enough parameters so that the resulting parameter estimation problem is overconstrained. In such situations solutions are allowed to differ from the data in a way that accounts for noise in the measurements.

In this section we consider a special class of underconstrained tomographic problems that permits the use of a simplifying model. The class of problems we consider are those in which the imaging process is targeted toward tissues or organs that have been set apart from the other anatomy by some contrast agent. This agent could be an opaque dye, as in the case of transmission tomography, or an emissive metabolite, as in nuclear medicine. We assume that this agent produces relatively homogeneous patches that are bounded by areas of high contrast. This assumption is reasonable, for instance, in subtractive angiography or CT studies of the colon. The proposed approach, therefore, seeks to find the boundaries of different regions in a volume by estimating sets of closed surface models and their associated density parameters directly from the incomplete sinogram data [74]. Thus, the reconstruction problem is converted to a segmentation problem. Of course, we can never expect real tissues to exhibit truly homogeneous densities. However, we assert that when inhomogeneities are somewhat uncorrelated and of low contrast the proposed model is adequate to obtain acceptable reconstructions.

8.6.1 Related Work

Several areas of distinct areas of research in medical imaging, computer vision, and inverse problems impact this work. Numerous tomographic reconstruction methods are described in the literature [75, 76], and the method of choice depends on the quality of projection data. Filtered back projection (FBP), the

most widely used approach, works well in the case of the fully constrained reconstruction where one is given *enough high-quality projections over 180° angular range*. Statistical, iterative approaches such as maximum likelihood (ML) and maximum *a posteriori* (MAP) estimation have been proven to work well with *noisy* projection data, but do not systematically address the underconstrained reconstruction problem and generally rely on complete datasets. An exception is [77], which proposes an iterative algebraic approach that includes some assumptions about the homogeneity of the solution to compute a full grayscale reconstruction. Also, some hybrid approaches [78, 79] are specifically developed to deal with *limited-angle* tomography by extrapolating the missing sinogram data.

Other tomographics reconstruction techniques have been proposed, for example those that utilize discrete tomography strategies [73, 80–82], and deformable models [83–87]. The literature also describes many examples of level sets as curve and surface models for image segmentation [6, 7, 41, 88]. The authors have examined their usefulness for 3D segmentation of TEM *reconstructions* [37]. Several authors have proposed solving *inverse problems* using level sets [89–95], but are mostly limited to solving 2D problems.

We make several important contributions to this previous body of work; first we give a formal derivation of the motion of deformable surface models as the first variation of an error term that relates the projected model to the noisy tomographic data. This formulation does not assume any specific surface representation, and therefore applies to a wide range of tomographic, surface-fitting problems. Second we present a level set implementation of this formulation that computes incremental changes in the radon transform of the projected model only along the wave front, which makes it practical on large datasets. Third we examine the specific problem of initializing the deformable surface in the absence of complete sinogram data, and demonstrate, using real and synthetic data, the effectiveness of direct surface estimation for a specific class of tomographic problems which are underconstrained.

8.6.2 Mathematical Formulation

As an introduction, we begin with the derivation of surface estimation problem in two dimensions. The goal is to simultaneously estimate the interface between two materials and their densities, β_0 and β_1 . Thus we have a background with

density β_0 and collection of solid objects with density β_1 . We denote the (open) set of points in those regions as Ω , the closure of that set, the surface, as S .

The projection of a 2D signal $f(x, y)$ produces a sinogram given by the radon transform as

$$p(s, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(R_\theta \mathbf{x} - s) d\mathbf{x}, \quad (8.33)$$

where $R_\theta \mathbf{x} = x \cos(\theta) + y \sin(\theta)$ is a rotation and projection of a point $\mathbf{x} = (x, y)$ onto the imaging plane associated with θ . The 3D formulation is the same, except that the signal $f(x, y, z)$ produces a collection of images. We denote the projection of the model, which includes estimates of the objects and the background, as $\hat{p}(s, \theta)$. For this work we denote the angles associated with a discrete set of projections as $\theta_1, \dots, \theta_N$ and denote the domain of each projection as $S = s_1, \dots, s_M$. Our strategy is to find Ω , β_0 , and β_1 by maximizing the likelihood.

If we assume the projection measurements are corrupted by independent noise, the log likelihood of a collection of measurements for a specific shape and density estimate is the probability of those measurements conditional on the model,

$$\begin{aligned} \ln P(p(s_1, \theta_1), p(s_2, \theta_1), \dots, p(s_M, \theta_N) | S, \beta_0, \beta_1) \\ = \sum_i \sum_j \ln P(p(s_j, \theta_i) | S, \beta_0, \beta_1). \end{aligned} \quad (8.34)$$

We call the negative log likelihood the *error* and denote it E_{data} . Normally, the probability density of a measurement is parameterized by the ideal value, which gives

$$E_{\text{data}} = \sum_{i=1}^N \sum_{j=1}^M E(\hat{p}_{ij}, p_{ij}), \quad (8.35)$$

where $E(\hat{p}_{i,j}, p_{i,j}) = -\ln P(\hat{p}_{i,j}, p_{i,j})$ is the error associated with a particular point in the radon space, and $p_{i,j} = p(s_j, \theta_i)$. In the case of independent Gaussian noise, E is a quadratic, and the log likelihood results in a weighted least-squares in the radon space. For all of our results, we use a Gaussian noise model. Next we apply the object model, shown in Fig. 8.17, to the reconstruction of f . If we let $g(x, y)$ be a binary inside–outside function on Ω , then we have the following approximation to $f(x, y)$:

$$f(x, y) \approx \beta_0 + [\beta_1 - \beta_0]g(x, y). \quad (8.36)$$

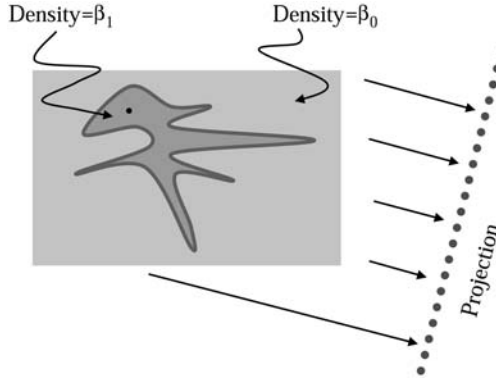


Figure 8.17: The model is the interface between two densities, which are projected onto the imaging plane to create $\hat{p}(s, \theta_i)$.

Applying the radon transform to the model and substituting for $\hat{p}$ gives

$$E_{\text{data}} = \sum_{i=1}^N \sum_{j=1}^M E \left(\beta_0 K(s_j, \theta_i) + [\beta_1 - \beta_0] \int_{\Omega} \delta(R_{\theta_i} \mathbf{x} - s_j) d\mathbf{x}, p_{ij} \right), \quad (8.37)$$

where $K(s_j, \theta_i)$ is the projection of the background—it depends on the geometry of the region over which the data is taken and is independent of the surface estimate. For some applications we know that $\beta_0 = 0$, and the term $\beta_0 K$ is zero. The integral over Ω results from integrating g over the entire domain.

The proposed strategy is to alternately (i.e. separately) update the shape of the surface model and the density parameters. For the surface shape, a gradient descent minimization approach describes the deformation of the surface, with respect to an evolution parameter t , as it progressively improves its fit to the

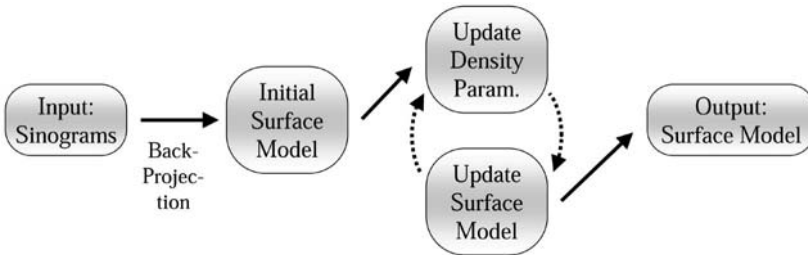


Figure 8.18: The reconstruction strategy starts with an initial surface estimate and iteratively modifies its shape and the associated density parameters to achieve a good fit to the input data.

sinogram data. The incremental change in the likelihood is

$$\frac{dE_{\text{data}}}{dt} = \int_S \sum_{i=1}^N \sum_{j=1}^M \frac{d}{dt} E(\hat{p}_{ij}, p_{i,j}) d\mathbf{x} = \int_S \sum_{i=1}^N \sum_{j=1}^M E'(\hat{p}_{ij}, p_{ij}) \frac{d\hat{p}_{ij}}{dt} d\mathbf{x}, \quad (8.38)$$

where $E' = \partial E / \partial \hat{p}$, which, for Gaussian noise, is simply the difference between $\hat{p}$ and p . Next we must formulate $d\hat{p}/dt$, which, by the transport equation, is

$$\begin{aligned} \frac{d\hat{p}_{ij}}{dt} &= [\beta_1 - \beta_0] \frac{d}{dt} \int_{\Omega} \delta(R_{\theta_i} \mathbf{x} - s_j) d\mathbf{x} \\ &= [\beta_1 - \beta_0] \int_S \delta(R_{\theta_i} \mathbf{x} - s_j) \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (8.39)$$

where $\mathbf{n}$ is an outward pointing surface normal and $\mathbf{v}(\mathbf{x})$ is the velocity of the surface at the point $\mathbf{x}$. The derivative of E_{data} with respect to surface motion is therefore

$$\frac{dE_{\text{data}}}{dt} = [\beta_1 - \beta_0] \int_S \sum_{i=1}^N \sum_{j=1}^M E'(\hat{p}_{i,j}, p_{ij}) \delta(R_{\theta_i} \mathbf{x} - s_j) \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}. \quad (8.40)$$

Note that the integral over $d\mathbf{x}$ and the δ functional serve merely to associate s_j in the i th scan with the appropriate $\mathbf{x}$ point. If the samples in each projection are sufficiently dense, we can approximate the sum over j as an integral over the image domain, and thus for every $\mathbf{x}$ on the surface there is a mapping back into the i th projection. We denote this point $s_i(\mathbf{x})$. This gives a closed-form expression for the derivative of the derivative of E_{data} in terms of the surface velocity,

$$\frac{dE_{\text{data}}}{dt} = [\beta_1 - \beta_0] \int_S \sum_{i=1}^N e_i(\mathbf{x}) \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}, \quad (8.41)$$

where $e_i(\mathbf{x}) = E'(\hat{p}(s_i(\mathbf{x}), \theta_i), p(s_i(\mathbf{x}), \theta_i))$ is the derivative of the error associated with the point $s_i(\mathbf{x})$ in the i th projection. The result shown in Eq. (8.41) does not make any specific assumptions about the surface shape or its representation. Thus, this equation could be mapped onto any set of shape parameters by inserting the derivative of a surface point with respect to those parameters. Of course one would have to compute the surface integral, and methods for solving such equations on parametric models (in the context of range data) are described in [96].

For this work we are interested in *free-form* deformations, where each point on the surface can move independently from the rest. If we let $\mathbf{x}_t$ represent the velocity of a point on the surface, the gradient descent surface free-form surface

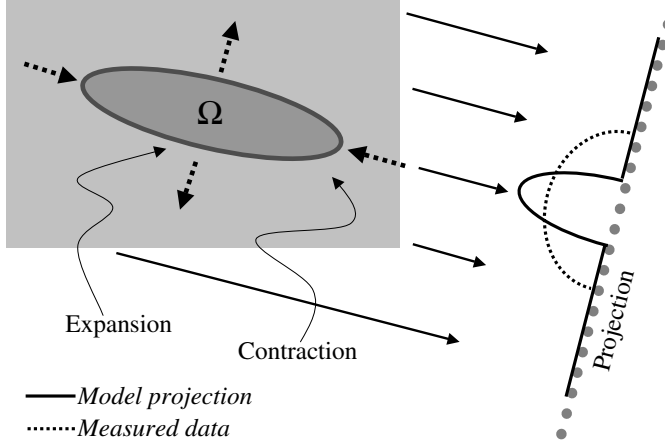


Figure 8.19: The model expands or contracts based on the difference in the sinograms between the projected model and the measured data.

motion is

$$\mathbf{x}_t = -\frac{dE_{\text{data}}}{d\mathbf{x}} = (\beta_0 - \beta_1) \sum_{i=1}^N e_i(\mathbf{x}) \mathbf{n}(\mathbf{x}). \quad (8.42)$$

Thus, at a point $\mathbf{x} \in \mathcal{S}$, the i th projection has the effect of causing the surface to expand or contract according to the difference between the projected model values and the measured data at the point $s_i(\mathbf{x})$, the projection of $\mathbf{x}$ (Fig. 8.19). The surface motion prescribed by a collection of projections is the sum of motions from the individual projections. In the case of continuous set of angles, the surface motion at a point is proportional to the sinusoidal line integral on the *error sinogram*, which is $e(s, \theta)$.

8.6.2.1 Density Parameter Estimation

The density parameters also affect the error term in Eq. (8.37). We propose to update the estimate of the surface model iteratively, and at each iteration we re-estimate the quantities β_0 and β_1 in such a way that the energy E_{data} is minimized. Treating Ω as fixed, Eq. (8.37) has two unknowns, β_0 and β_1 , which are computed from the following system:

$$\frac{\partial E_{\text{data}}}{\partial \beta_0} = 0, \quad \frac{\partial E_{\text{data}}}{\partial \beta_1} = 0. \quad (8.43)$$

In the case of a Gaussian noise model, (8.43) is a linear system. Because of variations in instrumentation, the contrast levels of images taken at different angles can vary. In such cases we estimate sets of such parameters, i.e., $\beta_0(\theta_i)$ and $\beta_1(\theta_i)$ for $i = 1, \dots, N$.

To extend the domain to higher dimensions, we have $\mathbf{x} \in \mathbb{R}^n$, and $S \subset \mathbb{R}^{n-1}$ and the mapping $\mathbf{s}_i : \mathbb{R}^n \mapsto S$ models the projective geometry of the imaging system (e.g. orthographic, cone beam, or fan beam). Otherwise, the formulation is the same as in 2D.

One important consideration is to model more complex models of density. If β_0 and β_1 are smooth, scalar functions defined over the space in which the surface model deforms and g is a binary function, the density model is

$$f(\mathbf{x}) = \beta_0(\mathbf{x}) + (\beta_1(\mathbf{x}) - \beta_0(\mathbf{x}))g(x, y). \quad (8.44)$$

The first variation of the boundary is simply

$$\frac{d\mathbf{x}}{dt} = [\beta_1(\mathbf{x}) - \beta_0(\mathbf{x})] \sum_{i=1}^N e_i(\mathbf{x})\mathbf{n}(\mathbf{x}). \quad (8.45)$$

Note that this formulation is different from that of Yu *et al.* [95], who address the problem of reconstruction from noisy tomographic data using a single density function f with a smoothing term that interacts with a set of deformable edge models Γ . The edges models are surfaces, represented using level sets. In that case the variational framework for deforming Γ requires differentiation of f across the edge, precisely where the proposed model exhibits (intentionally) a discontinuity.

8.6.2.2 Prior

The analysis above maximizes the likelihood. For a full MAP estimation, we include a prior term. Because we are working with the logarithm of the likelihood, the effect of the prior is additive:

$$\mathbf{x}_t = -\frac{dE_{\text{data}}}{d\mathbf{x}} - \frac{dE_{\text{prior}}}{d\mathbf{x}}. \quad (8.46)$$

Thus in addition to the noise model, we can incorporate some knowledge about the kinds of shapes that give rise to the measurements. With appropriately fashioned priors, we can push the solution toward desirable shapes or density values, or penalize certain shape properties, such as roughness or complexity. The

choice of prior is intimately related to the choice of surface representation and the specific application, but is independent of the formulation that describes the relationship between the estimate and the data, given in Eq. (8.37).

Because the data is noisy and incomplete it is useful to introduce a simple, low-level prior on the surface estimate. We therefore use a prior that penalizes surface area, which introduces a second-order smoothing term in the surface motion. That term introduces a free parameter C , which controls the relative influence of the smoothing term. The general question of how best to smooth surfaces remains an important, open question. However, if we restrict ourselves to curvature-based geometric flows, there are several reasonable options in the literature [7, 31, 97]. The following subsection, which describes the surface representation used for our application, gives a more precise description of our smoothing methods.

8.6.3 Surface Representation and Prior

Our goal is to build an algorithm that applies to a wide range of potentially complicated shapes with arbitrary topologies—topologies that could change as the shapes deform to fit the data. For this reason, we have implemented the free-form deformation given in Eq. (8.42) with an implicit level set representation.

Substituting the expression for $d\mathbf{x}/dt$ (from Eqs. (8.45) and (8.46)) into the $d\mathbf{s}/dt$ term of the level set equation (Eq. (8.4a)), and recalling that $\mathbf{n} = \nabla\phi/|\nabla\phi|$, gives

$$\frac{\partial\phi}{\partial t} = -|\nabla\phi| \left(\sum_{i=1}^M e_i(\mathbf{x}) + C\kappa(\mathbf{x}) \right), \quad (8.47)$$

where κ represents the effect of the prior, which is assumed to be in the normal direction.

The prior is introduced as a curvature-based smoothing on the level set surfaces. Thus, every level set moves according to a weighted combination of the principle curvatures, k_1 and k_2 , at each point. This point-wise motion is in the direction of the surface normal. For instance, the mean curvature, widely used for surface smoothing, is $H = (k_1 + k_2)/2$. Several authors have proposed using Gaussian curvature $K = k_1k_2$ or functions thereof [97]. Recently [98] proposed

using the minimum curvature, $M = \text{AbsMin}(k_1, k_2)$ for preserving thin, tubular structures, which otherwise have a tendency to *pinch off* under mean curvature smoothing.

In previous work [41], the authors have proposed a weighted sum of mean curvatures that emphasizes the minimum curvature, but incorporates a smooth transition between different surface regions, avoiding the discontinuities (in the derivative of motion) associated with a strict minimum. The *weighted curvature* is

$$W = \frac{k_1^2}{k_1^2 + k_2^2} k_2 + \frac{k_2^2}{k_1^2 + k_2^2} k_1 = \frac{2HK}{D^2}, \quad (8.48)$$

where $D = \sqrt{k_1^2 + k_2^2}$ is the deviation from flatness [99].

For an implicit surface, the shape matrix [100] is the derivative of the normal map projected onto the tangent plane of the surface. If we let the normal map be $\mathbf{n} = \nabla\phi/|\nabla\phi|$, the derivative of this is the 3×3 matrix

$$N = \begin{pmatrix} \frac{\partial \mathbf{n}}{\partial x} & \frac{\partial \mathbf{n}}{\partial y} & \frac{\partial \mathbf{n}}{\partial z} \end{pmatrix}^T. \quad (8.49)$$

The projection of this derivative matrix onto the tangent plane gives the shape matrix $B = N(I - \mathbf{n} \otimes \mathbf{n})$, where $\otimes$ is the exterior product and I is the 3×3 identity matrix. The eigenvalues of the matrix B are k_1, k_2 and zero, and the eigenvectors are the principle directions and the normal, respectively. Because the third eigenvalue is zero, we can compute k_1, k_2 , and various differential invariants directly from the invariants of B . Thus the weighted-curvature flow is computing from B using the identities $D = \|B\|_2$, $H = \text{Tr}(B)/2$, and $K = 2H^2 - D^2/2$. The choice of numerical methods for computing B is discussed in the following section.

8.6.4 Implementation

The level set equations are solved by finite differences on a discrete grid, i.e. a volume. This raises several important issues in the implementation. These issues are the choice of numerical approximations to the PDE, efficient and accurate schemes for representing the volume, and mechanisms for computing the sinogram-based deformation in Eq. (8.47).

8.6.4.1 Numerical Schemes

Osher *et al.* [30] have proposed an up-wind method for solving equations of the form $\phi_t = \nabla \phi \cdot \mathbf{v}$, of which $\phi_t = |\nabla \phi| \sum_i e_i(\mathbf{x})$, from Eq. (8.47), is an example. The up-wind scheme utilizes one-sided derivatives in the computation of $|\nabla \phi|$, where the direction of the derivative depends, point-by-point, on the sign of the speed term $\sum_i e_i(\mathbf{x})$. With strictly regulated time steps, this scheme avoids overshooting (ringing) and instability.

Under normal circumstances, the curvature term, which is a directional diffusion, does not suffer from overshooting; it can be computed directly from first- and second-order derivatives of ϕ using central difference schemes. However, we have found that central differences do introduce instabilities when computing flows that rely on quantities other than the mean curvature. Therefore, we use the method of *differences of normals* [101, 102] in lieu of central differences. The strategy is to compute normalized gradients at staggered grid points and take the difference of these staggered normals to get centrally located approximations to N (as in Fig. 8.20). The normal projection operator $\mathbf{n} \otimes \mathbf{n}$ is computed with gradient estimates from central differences. The resulting curvatures are

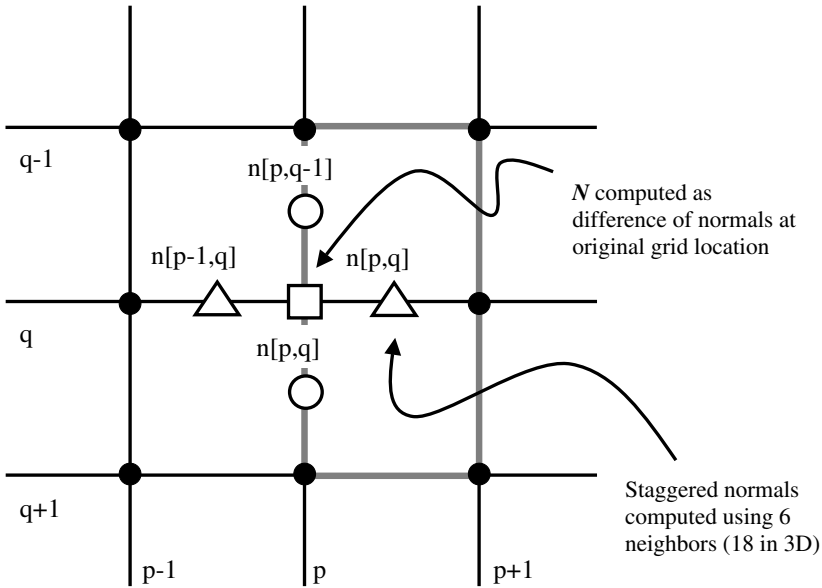


Figure 8.20: The shape matrix B is computed by using the differences of staggered normals.

treated as speed terms (motion in the normal direction), and the associated gradient magnitude is computed using the up-wind scheme.

8.6.4.2 Sparse-Field Method

The computational burden associated with solving the 3D, second-order, non-linear level set PDE is significant. For this reason several papers [34, 35] have proposed narrow-band methods, which compute solutions only for a relatively small set of pixels in the vicinity of k level set. The authors [36] have proposed a sparse-field algorithm, which uses an approximation to the distance transform and makes it feasible to recompute the neighborhood of the level set model at each time step. It computes updates on a band of grid points, called the *active set*, that is one point wide. Several layers around this active set are updated in such a way as to maintain a neighborhood in order to calculate derivatives. The position of the surface model is determined by the set of active points and their values.

8.6.4.3 Incremental Projection Updates

The tomographic surface reconstruction problem entails an additional computational burden, because the measured data must be compared to the projected model at *each iteration*. Specifically, computing $\hat{p}_{ij}$ can be a major bottleneck. Computing this term requires recomputing the sinogram of the surface/object model as it moves. In the worst case, we would reproject the entire model every iteration.

To address this computational concern, we have developed the method of *incremental projection updates* (IPU). Rather than fully recompute $\hat{p}$ at every iteration, we maintain a current running version of $\hat{p}$ and update it to reflect the changes in the model as it deforms. Changes in the model are computed only on a small set of grid points in the volume, and therefore the update time is proportional to the area of the surface, rather than the size of the volume it encloses.

The IPU strategy works with the the sparse-field algorithm as follows. At each iteration, the sparse-field algorithm updates only the active layer (one voxel wide) and modifies the set of active grid points as the surface moves. The incremental projection update strategy takes advantage of this to selectively update

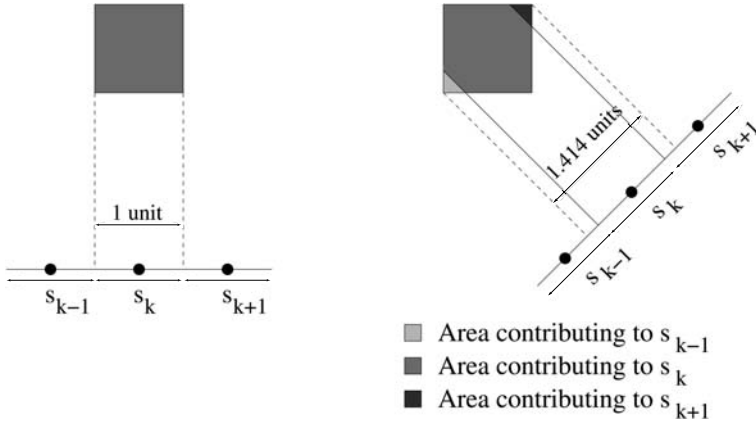


Figure 8.21: A weighting coefficient for each voxel determines the portions of the discrete sinogram influenced by incremental changes to a grid point.

the model projection to reflect those changes. At each iteration, the amount of change in an active point's value determines the motion of that particular surface point as well as the percentage of the surrounding voxel that is either inside or outside of the surface. By the linearity of projection, we can map these changes in the object shape, computed at grid points along the surface boundary, back into the sinogram space and thereby *incrementally* update the sinogram. Note that each 3D grid point has a weighting coefficient (these are precomputed and fixed), which is determined by its geometric mapping of the surrounding voxel back into the sinogram, as in Fig. 8.21. In this way the IPU method maintains subvoxel accuracy at a relatively low computational cost.

8.6.4.4 Initialization

The deformable model fitting approach requires an initial model, i.e. $\phi(\mathbf{x}, t = 0)$. This initial model should be obtained using the “best” information available prior to the surface fitting. In some cases this will mean thresholding a grayscale reconstruction, such as FBP, knowing that it has artifacts. In practice the initial surface estimate is impacted by the reconstruction method and the choice of threshold, and because we perform a local minimization, these choices can affect the final result. Fortunately, the proposed formulation is moderately robust with respect to the initial model, and our results show that the method works well under a range of reasonable initialization strategies.

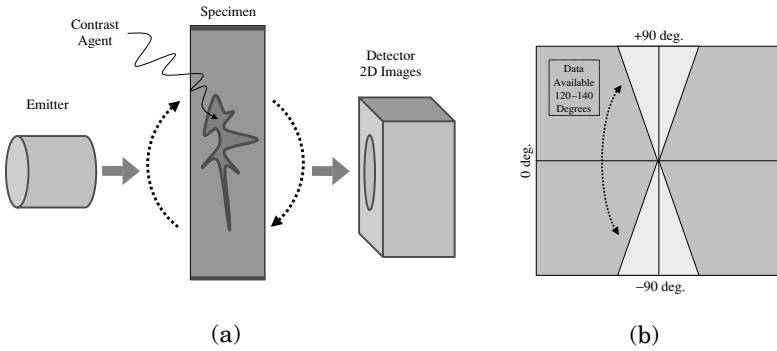


Figure 8.22: (a) Transmission electron microscopy is used to image very small specimens that have been set apart from the substrate by a contrast agent. (b) TEM imaging technology provides projections over a limited set of angles.

8.6.5 Results

8.6.5.1 Transmission Electron Microscopy

Transmission electron microscopy is the process of using transmission images of electron beams to reveal biological structures on very small dimensions. Typically transmission electron microscopy (TEM) datasets are produced using a dye that highlights regions of interest, e.g. the interior of a microscopic structure, such as a cell (see Fig. 8.22(a)). There are technical limits to the projection angles from which data can be measured. These limits are due to the mechanical apparatus used to tilt the specimens and the trade-off between the destructive effects of electron energy and the effective specimen thickness, which increases with tilt angle. Usually, the maximum tilt angle is restricted to about ± 60 – 70° . Figure 8.22(b) shows an illustration of the geometry of this limited-angle scenario. The TEM reconstruction problem is further aggravated by the degree of electron scattering, which results in projection images (sinograms) that are noisy relative to many other modalities, e.g. X-ray CT. Finally, due to the flexible nature of biological objects and the imperfections in the tilting mechanism, the objects undergo some movements while being tilted. Manual alignment procedures used to account for this tend to produce small misregistration errors.

We applied the proposed algorithm to 3D TEM data obtained from a 3 MeV electron microscope. This 3D dataset consists of 67 tilt series images, each corresponding to one view of the projection. Each tilt series image is of size 424×334 . The volume reconstructed by FBP is of size $424 \times 424 \times 334$. Figures 8.23(a)

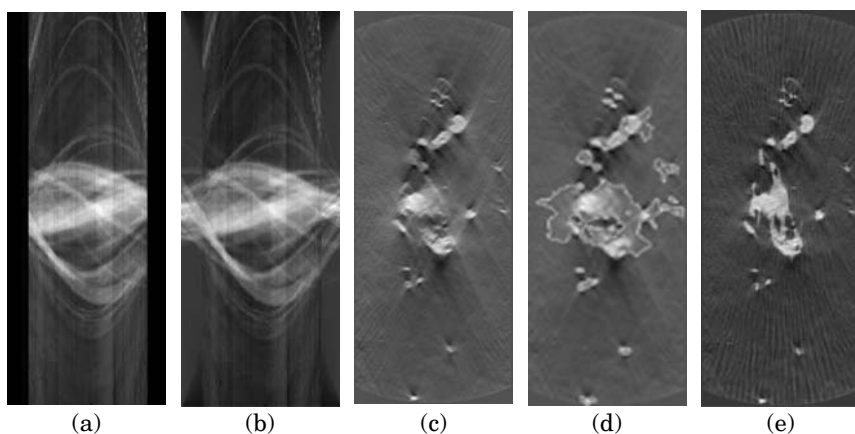


Figure 8.23: 2D slice of dendrite data: (a) sinogram of one slice, (b) sinogram estimated by the proposed method, (c) back projection showing artifacts, (d) initial model obtained by thresholding the back projection (white curve overlaid on the back projection), and (e) final surface estimate.

and (b) show the sinogram corresponding to a single slice of this dataset and the estimate of the same sinogram created by the method. Figure 8.23(e) shows the surface estimate intersecting this slice overlaid on the back projected slice. Some structures not seen in the back projection are introduced in the final estimation, but the orientation of the structures introduced suggests that these are valid features that were lost due to reconstruction artifacts from the FBP. Also, the proposed method captures line-by-line brightness variations in the input sinogram (as explained in Section 8.6.2.1). This suggests that the density estimation procedure is correct.

Figure 8.24 shows the 3D initialization and the final 3D surface estimate. The figure also shows enlarged initial and final versions of a small section of the surface. Computing the surface estimate for the TEM dendrite with 150 iterations took approximately 3 hours on a single 300 MHz processor of a Silicon Graphics Onyx2 workstation. We consider these results positive for several reasons. First, the biology is such that one expects the spines (small protrusions) to be connected to the dendrite body. The proposed method clearly establishes those connections, based solely on consistency of the model with the projected data. The second piece of evidence is the shapes of the spines themselves. The reconstructed model shows the recurrence of a characteristic shape—a long thin spine with a cup-like structure on the end. This characteristic structure, which

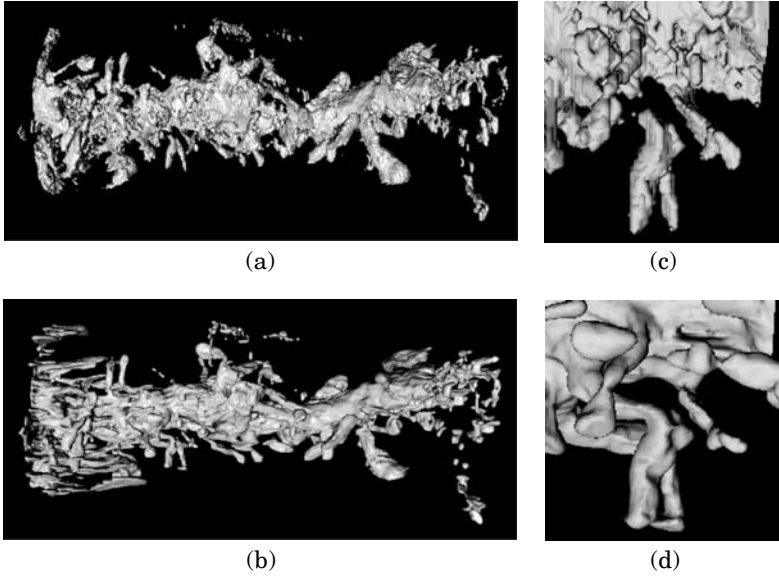


Figure 8.24: 3D results: (a) surface initialization, (b) final surface estimated after 150 iterations, (c) a portion of the initial surface enlarged, and (d) the corresponding portion in the final surface.

often fails to show up in the FBP reconstruction, does appear quite regularly in hand-segmentations of the same datasets.

8.6.5.2 Sinogram Extrapolation

The fitting of surfaces to this data is a simplification. It is justified in the context of segmentation, but there *are* underlying inhomogeneities in the density of this specimen, which could be indicative of relevant structures. Thus for some applications *direct* visualization of the measured data, by volume rendering, offers advantages over the segmented surfaces. We propose to use the surface estimation algorithm as a mechanism for estimating the missing data in the sinograms.

Figures 8.25(a) and (b) show the input sinogram and the sinogram of the estimated model (for one slice) of the TEM dendrite data. The estimated sinogram demonstrates that the surface estimation method recovers the missing information in a reasonable way. Thus, we combine the sinograms from the model with original sinograms to produce a “full” sinogram that still contains *all* of the

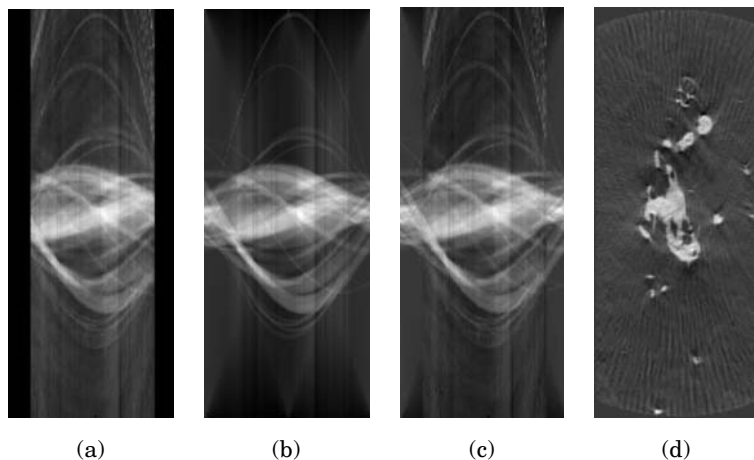


Figure 8.25: Sinogram extrapolation for slice number 150 of dendrite data: (a) input sinogram, (b) sinogram estimated by the proposed method, (c) augmented sinogram constructed using original data and estimating missing data from the segmentation, and (d) FBP reconstruction of the augmented sinogram.

original, measured data. FBP reconstructions from such augmented sinograms should have fewer limited-angle streak artifacts.

We demonstrate this by comparing volume renderings with and without the augmentation. We create augmented sinograms by using sinogram data from the estimated model only where the data is missing from the measured sinograms. The augmented sinogram for a single slice is shown in Fig. 8.25(c). The slice reconstructed (FBP) from the augmented sinogram is shown in Fig. 8.25(d). Note that this reconstructed slice does not contain the limited-angle artifacts that appear in the slice in Fig. 8.23(c). Maximum intensity projection (MIP) volume renderings of the volume created from original sinograms and the volume created from augmented sinograms are compared in Fig. 8.26. The main body of the dendrite, which exhibited a very convoluted and fuzzy boundary, shows better definition. Also, several of the spines which were dangling in the original reconstruction are now connected.

8.7 Conclusions

This chapter has described a level set segmentation framework and the pre-processing and data analysis techniques needed for a number of segmentation

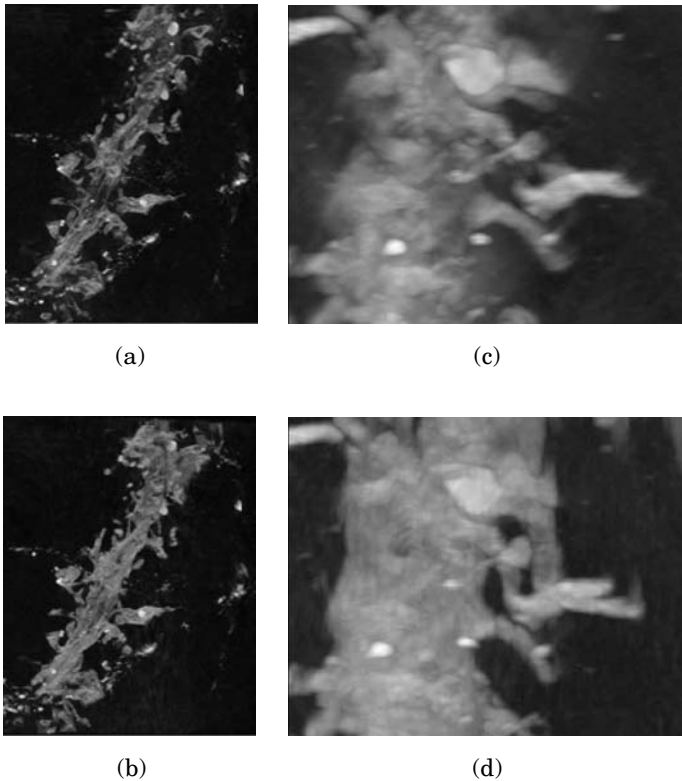


Figure 8.26: Sinogram extrapolation results: (a) MIP volume rendering of volume reconstructed from original sinograms, (b) MIP volume rendering of volume reconstructed from augmented (extrapolated) sinograms, (c) a portion of original MIP enlarged, and (d) the corresponding portion in augmented MIP enlarged.

applications. Several standard volume processing algorithms have been incorporated into the framework in order to segment datasets generated from MRI, CT, and TEM scans. A technique based on moving least-squares has been developed for segmenting multiple nonuniform scans of a single object. New scalar measures have been defined for extracting structures from diffusion tensor MRI scans. Finally, a direct approach to the segmentation of incomplete tomographic data using density parameter estimation is described. These techniques, combined with level set surface deformations, allow us to segment many different types of biological volume datasets.

8.8 Acknowledgements

Several people provided valuable technical assistance and support to our work. They are Dr. Alan Barr, Dr. Jason Wood, Dr. John Wood, Dr. Cyrus Papan, Dr. Russ Jacobs, Dr. Scott Fraser, Dr. J. Michael Tyszka, Dr. Miriam Scadeng, Dr. David Dubowitz, Dr. Eric Ahrens, Dr. Mark Ellisman, Dr. Maryanne Martone, Dr. Chris Johnson, and Dr. Mark Bastin. Datasets were provided by Caltech Biological Imaging Center (e.g. Fig. 8.8), National Center for Microscopy and Imaging Research (e.g. Fig. 8.7, funded by NIH grant P41-RR04050), Caltech Multi-Res Modeling Group (Fig. 8.9 (top)), Stanford Computer Graphics Laboratory (Fig. 8.9 (top)), Childrens Hospital—Los Angeles (Fig. 8.10), University of Utah’s SCI Institute (e.g. Fig. 8.14), and the University of Edinburgh, UK (Fig. 8.16).

This work was supported by National Science Foundation grants ASC-89-20219, ACI-9982273, ACI-0083287, and ACI-0089915, the Office of Naval Research Volume Visualization grant N00014-97-0227, the National Institute on Drug Abuse and the National Institute of Mental Health, as part of the Human Brain Project, the National Library of Medicine “Insight” Project N01-LM-0-3503, and the Caltech SURF Program.

Bibliography

- [1] Drebin, R., Carpenter, L., and Hanrahan, P., Volume rendering, In: Proceedings SIGGRAPH 88 Conference, pp. 65–74, 1988.
- [2] Levoy, M., Display of surfaces from volume data, *IEEE Comput. Graph. Appl.*, Vol. 9, No. 3, pp. 245–261, 1990.
- [3] Laur, D. and Hanrahan, P., Hierarchical splatting: A progressive refinement algorithm for volume rendering, In: SIGGRAPH '91 Proceedings, Sederberg, T. W., ed., pp. 285–288, 1991.
- [4] Parker, S., Parker, M., Livnat, Y., Sloan, P., Hansen, C., and Shirley, P., Interactive Ray Tracing for volume visualization, *IEEE Trans. Vis. Comput. Graph.*, Vol. 5, No. 3, pp. 238–250, 1999.
- [5] Leventon, M., Faugeraus, O., Grimson, W., and Wells, W. III, Level set based segmentation with intensity and curvature priors, In: Workshop on Mathematical Methods in Biomedical Image Analysis Proceedings, pp. 4–11, 2000.
- [6] Malladi, R., Sethian, J., and Vemuri, B., Shape modeling with front propagation: A level set approach, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 2, pp. 158–175, 1995.
- [7] Sethian, J., *Level Set Methods and Fast Marching Methods*, 2nd edn., Cambridge University Press, Cambridge, UK, 1999.
- [8] Staib, L., Zeng, X., Schultz, R., and Duncan, J., Shape constraints in deformable models, In: *Handbook of Medical Imaging*, Bankman, I., ed., Academic Press, New York, Chapter 9, pp. 147–157, 2000.
- [9] Wu, Z., Chung, H.-W., and Wehrli, F. W., A Bayesian approach to sub-voxel tissue classification in NMR microscopic images of trabecular bone, *J. Comput. Assist. Tomogr.*, Vol. 12, No. 1, pp. 1–9, 1988.
- [10] Kao, Y.-H., Sorenson, J. A., and Winkler, S. S., MR image segmentation using vector decomposition and probability techniques: A general model and its application to dual-echo images, *Magn. Reson. Med.*, Vol. 35, pp. 114–125, 1996.

- [11] Cline, H. E., Lorensen, W. E., Kikinis, R., and Jolesz, F., Three-dimensional segmentation of MR images of the head using probability and connectivity, *J. Comput. Assist. Tomogr.*, Vol. 14, No. 6, pp. 1037–1045, 1990.
- [12] Laidlaw, D. H., Fleischer, K. W., and Barr, A. H., Partial-volume Bayesian classification of material mixtures in MR volume data using voxel histograms, *IEEE Trans. Med. Imaging*, Vol. 17, No. 1, pp. 74–86, 1998.
- [13] Johnson, V. E., A framework for incorporating structural prior information into the estimation of medical images, In: *Information Processing in Medical Imaging (IPMI'93)*, Barrett, H. H. and Gmitro, A. F., eds., No. 687 In *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 307–321, 1993.
- [14] Marr, D. and Hildreth, E., Theory of Edge Detection, *Proc. R. Soc. London*, Vol. B, No. 207, pp. 187–217, 1980.
- [15] Marr, D., *Vision*, Freeman, San Francisco, 1982.
- [16] Canny, J., A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, No. 6, pp. 679–698, 1986.
- [17] Cootes, T., Hill, A., Taylor, C., and Haslam, J., The use of active shape models for locating structures in medical images, In: *Information Processing in Medical Imaging (IPMI'93)*, Barrett, H. H. and Gmitro, A. F., eds., No. 687 In *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 33–47, 1993.
- [18] Stetten, G. and Pizer, S., Medial node models to identify and measure objects in real-time 3D echocardiography, *IEEE Trans. Med. Imaging*, Vol. 18, No. 10, pp. 1025–1034, 1999.
- [19] Wood, Z., Desbrun, M., Schröder, P., and Breen, D., Semi-regular mesh extraction from volumes, In: *Proceedings of IEEE Visualization 2000*, pp. 275–282, 2000.
- [20] Miller, J., Breen, D., Lorensen, W., O'Bara, R., and Wozny, M., Geometrically deformed Models: A method for extracting closed geometric models from volume data, In: *SIGGRAPH '91 Proceedings*, pp. 217–226, 1991.

- [21] Pentland, A. P., Perceptual organization and the representation of natural form, *Artif. Intell.*, Vol. 28, pp. 293–331, 1986.
- [22] Terzopoulos, D. and Metaxas, D., Dynamic 3D models with local and global deformations: Deformable superquadrics, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 13, No. 7, pp. 703–714, 1991.
- [23] Gupta, A. and Bajcsy, R., Volumetric segmentation of range images of 3D objects using superquadric models, *CVGIP: Image Underst.*, Vol. 58, No. 3, pp. 302–326, 1993.
- [24] Muraki, S., Volumetric shape description of range data using “Blobby Model,” In: *SIGGRAPH ’91 Proceedings*, Sederberg, T. W., ed., pp. 227–235, 1991.
- [25] Szeliski, R., Tonnesen, D., and Terzopoulos, D., Modeling surfaces of arbitrary topology with dynamic particles, In: *Proc. Fourth Int. Conf. on Comp. Vision (ICCV’93)*, pp. 82–87, IEEE Computer Society Press, Berlin, 1993.
- [26] McInerney, T. and Terzopoulos, D., A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis, *Comput. Med. Imaging Graph.*, Vol. 19, No. 1, pp. 69–83, 1995.
- [27] Park, J., Metaxas, D., Young, A. A., and Axel, L., Deformable models with parameter functions for cardiac motion analysis from tagged MRI data, *IEEE Trans. Med. Imaging*, Vol. 15, No. 3, pp. 278–289, 1996.
- [28] DeCarlo, D. and Metaxas, D., Shape evolution with structural and topological changes using blending, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 20, No. 11, pp. 1186–1205, 1998.
- [29] Ramamoorthi, R. and Arvo, J., Creating generative models from range images, In: *SIGGRAPH ’99 Proceedings*, pp. 195–204, 1999.
- [30] Osher, S. and Sethian, J., Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.*, Vol. 79, pp. 12–49, 1988.
- [31] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer, Berlin, 2002.

- [32] Sethian, J., A fast marching level set method for monotonically advancing fronts, In: *Proceedings of the National Academy of Science*, Vol. 93 of 4, pp. 1591–1595, 1996.
- [33] Tsitsiklis, J., Efficient algorithms for globally optimal trajectories, *IEEE Trans. Autom. Control*, Vol. 40, No. 9, pp. 1528–1538, 1995.
- [34] Adalsteinsson, D. and Sethian, J. A., A fast level set method for Propagating interfaces, *J. Comput. Phys.*, Vol. 118, No. 2, pp. 269–277, 1995.
- [35] Peng, D., Merriman, B., Osher, S., Zhao, H.-K., and Kang, M., A PDE-based fast local level set method, *J. Comput. Phys.*, Vol. 155, pp. 410–438, 1999.
- [36] Whitaker, R., A level-set approach to 3D reconstruction from range data, *Int. J. Comput. Vis.*, Vol. 29, No. 3, pp. 203–231, 1998.
- [37] Whitaker, R., Breen, D., Museth, K., and Soni, N., Segmentation of biological datasets using a level-set framework, In: *Volume Graphics 2001*, Chen, M. and Kaufman, A., eds., Springer, Vienna, pp. 249–263, 2001.
- [38] van den Boomgaard, R. and Smeulders, A. W. M., The morphological structure of images, the differential equations of morphological scale-space, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 16, No. 11, pp. 1101–1113, 1994.
- [39] Maragos, P., Differential morphology and image processing, *IEEE Trans. Image Process.*, Vol. 5, No. 6, pp. 922–937, 1996.
- [40] Requicha, A. and Voelcker, H., Boolean operations in solid modeling: Boundary evaluation and merging algorithms, *Proc. IEEE*, Vol. 73, No. 1, pp. 30–44, 1985.
- [41] Whitaker, R. T., Volumetric deformable models: Active blobs, In: *Visualization in Biomedical Computing*, Robb, R. A., ed., SPIE, Mayo Clinic, Rochester, MN, pp. 122–134, 1994.
- [42] Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, Cambridge, UK, 2001.

- [43] Museth, K., Breen, D., Zhukov, L., and Whitaker, R., Level set segmentation from multiple non-uniform volume datasets, In: Proc. IEEE Visualization Conference, pp. 179–186, 2002.
- [44] Shepard, D., A two-dimensional interpolation function for irregularly spaced points, In: Proc. ACM Nat. Conf., pp. 517–524, 1968.
- [45] Lancaster, P. and Salkauskas, K., Surfaces generated by moving least squares methods, *Math. Comput.*, Vol. 37, pp. 141–159, 1981.
- [46] Farwig, R., Multivariate interpolation of arbitrarily spaced data by moving least-squares methods, *J. Comput. Appl. Math.*, Vol. 16, pp. 79–93, 1986.
- [47] Zhao, H.-K., Osher, S., and Fedkiw, R., Fast surface reconstruction using the level set method, In: Proc. 1st IEEE Workshop on Variational and Level Set Methods, pp. 194–202, 2001.
- [48] Turk, G. and Levoy, M., Zippered polygon meshes from range images, In: Proc. of SIGGRAPH '94, pp. 311–318, ACM SIGGRAPH, 1994.
- [49] Curless, B. and Levoy, M., A volumetric method for building complex models from range images, In: Proc. SIGGRAPH '96, pp. 303–312, 1996.
- [50] Tamez-Pena, J., Totterman, S., and Parker, K., MRI isotropic resolution reconstruction from two orthogonal scans, In: Proc. SPIE Medical Imaging, Vol. 4322, pp. 87–97, 2001.
- [51] Goshtasby, A. and Turner, D. A., Fusion of short-axis and long-axis cardiac MR images, In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, San Francisco, pp. 202–211, 1996.
- [52] Brejl, M. and Sonka, M., Directional 3D Edge Detection in anisotropic data: Detector design and performance assessment, *Comput. Vis. Image Underst.*, Vol. 77, pp. 84–110, 2000.
- [53] Haralick, R. M. and Shapiro, L. G., *Computer and Robot Vision*, Addison-Wesley, Reading, MA, 1991.

- [54] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W., *Numerical Recipes in C*, 2nd edn., Cambridge University Press, New York, NY, 1992.
- [55] Bassar, P. J., Mattiello, J., and Bihan, D. L., Estimation of the effective self-diffusion tensor from the NMR spin echo, *J. Magn. Reson., B*, Vol. 103, No. 3, pp. 247–254, 1994.
- [56] Bassar, P. J., Mattiello, J., and Bihan, D. L., MR diffusion tensor spectroscopy and imaging, *Biophys. J.*, Vol. 66, No. 1, pp. 259–267, 1994.
- [57] Bassar, P. J. and Pierpaoli, C., Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI, *J. Magn. Reson., B*, Vol. 111, No. 3, pp. 209–219, 1996.
- [58] Westin, C.-F., Peled, S., Gudbjartsson, H., Kikinis, R., and Jolesz, F. A., Geometrical diffusion measures for MRI from tensor basis analysis, In: *Proceedings ISMRM 5th Annual Meeting*, p. 1742, 1997.
- [59] Peled, S., Gudbjartsson, H., Westin, C., Kikinis, R., and Jolesz, F., Magnetic resonance imaging shows orientation and asymmetry in white matter fiber tracts, *Brain Res.*, Vol. 780, pp. 27–33, 1998.
- [60] Bassar, P. and Pajevic, S., Statistical artifacts in diffusion tensor MRI caused by background noise, *Magn. Reson. Med.*, Vol. 44, pp. 41–50, 2000.
- [61] Ulug, A. and van Zijl, P., Orientation-independent diffusion imaging without tensor diagonalization: Anisotropy definitions based on physical attributes of the diffusion ellipsoid, *J. Magn. Reson. Imaging*, Vol. 9, pp. 804–813, 1999.
- [62] Laidlaw, D., Ahrens, E., Kremers, D., Avalos, M., Jacobs, R., and Readhead, C., Visualizing diffusion tensor images of the mouse spinal cord, In: *Proceedings IEEE Visualization '98*, pp. 127–134, 1998.
- [63] Kindlmann, G. and Weinstein, D., Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields, In: *Proc. IEEE Visualization '99*, pp. 183–189, 1999.

- [64] Zhukov, L., , Museth, K., Breen, D., Whitaker, R., and Barr, A., Level set modeling and segmentation of DT-MRI brain data, *J. Electron. Imaging*, Vol. 12, No. 1, pp. 125–133, 2003.
- [65] Bassar, P., Pajevic, S., Pierpaoli, C., Duda, J., and Aldroubi, A., *In vivo* fiber tractography using DT-MRI data, *Magn. Reson. Med.*, Vol. 44, pp. 625–632, 2000.
- [66] Poupon, C., Clark, C., Frouin, V., Regis, J., Bloch, I., Bihan, D. L., and Mangin, J.-F., Regularization of diffusion-based direction maps for the tracking of brain white matter fascicles, *Neuroimage*, Vol. 12, pp. 184–195, 2000.
- [67] Singh, A., Goldgof, D., and Terzopoulos, D., eds., *Deformable Models in Medical Image Analysis*, IEEE Computer Society Press, Los Alamitos, CA, 1998.
- [68] Kindlmann, G. and Durkin, J., Semi-automatic generation of transfer functions for direct volume rendering, In: *Proc. IEEE Symposium on Volume Visualization*, pp. 79–86, 1998.
- [69] Zhukov, L., Weinstein, D., and Johnson, C., Independent component analysis for EEG source localization in realistic head model, *IEEE Eng. Med. Biol.*, Vol. 19, pp. 87–96, 2000.
- [70] Gibson, S. *et al.*, Volumetric object modeling for surgical simulation, *Med. Image Anal.*, Vol. 2, No. 2, pp. 121–132, 1998.
- [71] Bailey, M., Manufacturing isovolumes, In: *Volume Graphics*, Chen, M., Kaufman, A., and Yagel, R., eds., Springer-Verlag, London, pp. 79–83, 2000.
- [72] Lorensen, W. and Cline, H., Marching cubes: A high resolution 3D surface construction algorithm, In: *Proc. SIGGRAPH '87*, pp. 163–169, 1987.
- [73] Ramm, A. G. and Katsevich, A. I., *The radon transform and local tomography*, CRC Press, Inc., Boca Raton, FL, 1996.
- [74] Elangovan, V. and Whitaker, R., From Sinograms to Surfaces: A Direct Approach to the Segmentation of Tomographic Data, In: *Proc. MICCAI*

- 2001, Vol. 2208 of *Lecture Notes in Computer Science*, Springer, Berlin, 2001.
- [75] Herman, G. T., *Image reconstruction from projections, The Fundamentals of Computerized Tomography*, Academic Press, New York, 1980.
- [76] Roerdink, J. B. T. M., *Computerized tomography and its applications: A guided tour*, *Nieuw Archief voor Wiskunde*, Vol. 10, No. 3, pp. 277–308, 1992.
- [77] Wang, G., Vannier, M., and Cheng, P., *Iterative X-ray cone-beam tomography for metal artifact reduction and local region reconstruction*, *Microsc. Microanal.*, Vol. 5, pp. 58–65, 1999.
- [78] Inouye, T., *Image reconstruction with limited angle projection data*, *IEEE Trans. Nucl. Sci.*, Vol. NS-26, pp. 2666–2684, 1979.
- [79] Prince, J. L. and Willsky, A. S., *Hierarchical reconstruction using geometry and sinogram restoration*, *IEEE Trans. Image Process.*, Vol. 2, No. 3, pp. 401–416, 1993.
- [80] Herman, G. T. and Kuba, A., eds., *Discrete Tomography: Foundations, Algorithms, and Applications*, Birkhauser, Boston, 1999.
- [81] Thirion, J. P., *Segmentation of tomographic data without image reconstruction*, *IEEE Trans. Med. Imaging*, Vol. 11, pp. 102–110, 1992.
- [82] Sullivan, S., Noble, A., and Ponce, J., *On reconstructing curved object boundaries from sets of X-ray images*, In: *Proceedings of the 1995 Conference on Computer Vision, Virtual Reality, and Robotics in Medicine*, Ayache, N., ed., *Lecture Notes in Computer Science 905*, pp. 385–391, Springer-Verlag, Berlin, 1995.
- [83] Hanson, K., Cunningham, G., Jr., and Wolf, D., *Tomographic reconstruction based on flexible geometric models*, In: *IEEE Int. Conf. on Image Processing (ICIP 94)*, pp. 145–147, 1994.
- [84] Battle, X. L., Cunningham, G. S., and Hanson, K. M., *3D tomographic reconstruction using geometrical models*, In: *Medical Imaging: Image Processing*, Hanson, K. M., ed., Vol. 3034, pp. 346–357, SPIE, 1997.

- [85] Battle, X. L., Bizais, Y. J., Rest, C. L., and Turzo, A., Tomographic reconstruction using free-form deformation models, In: *Medical Imaging: Image Processing*, Hanson, K. M., ed., Vol. 3661, pp. 356–367, SPIE, 1999.
- [86] Battle, X. L., LeRest, C., Turzo, A., and Bizais, Y., Three-dimensional attenuation map reconstruction using geometrical models and free-form deformations, *IEEE Trans. Med. Imaging*, Vol. 19, No. 5, pp. 404–411, 2000.
- [87] Mohammad-Djafari, A., Sauer, K., Khayi, Y., and Cano, E., Reconstruction of the shape of a compact object from a few number of projections, In: *IEEE International Conference on Image Processing (ICIP)*, Vol. 1, pp. 165–169, 1997.
- [88] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic active contours, In: *5th Int. Conf. on Comp. Vision*, pp. 694–699, IEEE, IEEE Computer Society Press, 1995.
- [89] Santosa, F., A level-set approach for inverse problems involving obstacles, *European Series in Applied and Industrial Mathematics: Control Optimization and Calculus of Variations*, Vol. 1, pp. 17–33, 1996.
- [90] Dorn, O., Miller, E. L., and Rappaport, C., A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets, *Inverse Prob.: Special issue on Electromagnetic Imaging and Inversion of the Earth's Subsurface (Invited Paper)*, Vol. 16, pp. 1119–1156, 2000.
- [91] Dorn, O., Miller, E. L., and Rappaport, C., Shape reconstruction in 2D from limited-view multi-frequency electromagnetic data, *AMS series Contemp. Math.*, Vol. 278, pp. 97–122, 2001.
- [92] Chan, T. F. and Vese, L. A., A level set algorithm for minimizing the Mumford–Shah functional in image processing, *Tech. Rep. CAM 00-13*, UCLA, Department of Mathematics, 2000.
- [93] Tsai, A., Yezzi, A., and Willsky, A., A curve evolution approach to smoothing and segmentation using the Mumford–Shah functional, In:

- Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 119–124, 2000.
- [94] Debreuve, E., Barlaud, M., Aubert, G., and Darcourt, J., Attenuation map segmentation without reconstruction using a level set method in nuclear medicine imaging, In: IEEE International Conference on Image Processing (ICIP), Vol. 1, pp. 34–38, 1998.
 - [95] Yu, D. and Fessler, J., Edge-preserving tomographic reconstruction with nonlocal regularization, In: Proceedings of IEEE Intl. Conf. on Image Processing, pp. 29–33, 1998.
 - [96] Whitaker, R. and Gregor, J., A maximum likelihood surface estimator for dense range data, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 24, No. 10, pp. 1372–1387, 2002.
 - [97] Sapiro, G., Geometric Partial Differential Equations and Image Analysis, Cambridge University Press, Cambridge, 2001.
 - [98] Lorigo, L., Faugeras, O., Grimson, E., Keriven, R., Kikinis, R., Nabavi, A., and Westin, C.-F., Co-dimension 2 geodesic active contours for the segmentation of tubular structures, In: Proceedings of IEEE Conf. on Comp. Vision and Pattern Recognition, pp. 444–452, 2000.
 - [99] Koenderink, J. J., Solid Shape, MIT Press, Cambridge, MA, 1990.
 - [100] do Carmo, M., Differential Geometry of Curves and Surfaces, Prentice-Hall, Englewood Cliffs, NJ, 1976.
 - [101] Rudin, L., Osher, S., and Fatemi, C., Nonlinear total variation based noise removal algorithms, Physica D, Vol. 60, pp. 259–268, 1992.
 - [102] Whitaker, R. and Xue, X., Variable-conductance, level-set curvature for image denoising, In: Proc. IEEE International Conference on Image Processing, pp. 142–145, 2001.

Chapter 9

Advanced Segmentation Techniques

*Aly A. Farag,¹ Mohamed N. Ahmed,² Ayman El-Baz,¹ and
Hossam Hassan¹*

9.1 Introduction

The principal goal of the segmentation process is to partition an image into regions that are homogeneous with respect to one or more characteristics or features. Segmentation is an important tool in medical image processing and it has been useful in many applications including lesion quantification, surgery simulations, surgical planning, multiple sclerosis, functional mapping, computer assisted diagnosis, image registration and matching, etc.

A wide variety of segmentation techniques has been proposed. However, there is no one standard segmentation technique that can produce satisfactory results for all imaging applications. Quite often, methods are optimized to deal with specific imaging modalities such as magnetic resonance (MR) imaging and X-ray computed tomography (CT), or modeled to segment specific anatomic structures such as the brain, the lungs, and the vascular system.

Recent research has demonstrated that the segmentation of anatomical structures from MRI and CT will benefit from the exploitation of three different types of knowledge: intensity models that describe the gray-level appearance of individual structures, shape models that describe the shape of different structures as well as imaging models that capture the characteristics of the imaging process.

¹Computer Vision and Image Processing Laboratory, Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA

²Software Research, C19L, Lexmark International Inc., Lexington, KY 40550, USA, E-mail: farag@cvip.uofl.edu

Stochastic image models are useful in quantitatively specifying natural constraints and general assumption about the physical world and the imaging process. Random field models permit the introduction of spatial context into pixel labeling problem. An introduction to random fields and its application in lung CT segmentation will be presented in Section 9.2.

Crisp segmentation, by which a pixel is assigned to a one particular region, often presents problems. In many situations, it is not easy to determine if a pixel should belong to a region or not. This is because the features used to determine homogeneity may not have sharp transitions at region boundaries. To alleviate this situation, we can inset fuzzy set concepts into the segmentation process. In Section 9.4, we will present an algorithm for fuzzy segmentation of MRI data and estimation of intensity inhomogeneities using fuzzy logic. MRI intensity inhomogeneities can be attributed to imperfections in the RF coils or to problems associated with the acquisition sequences. The result is a slowly varying shading artifact over the image that can produce errors with conventional intensity-based classification. The algorithm is formulated by modifying the objective function of the standard fuzzy c-means (FCM) algorithm to compensate for such inhomogeneities and to allow the labeling of a pixel (voxel) to be influenced by the labels in its immediate neighborhood. The neighborhood effect acts as a regularizer and biases the solution toward piecewise-homogeneous labelings. Such a regularization is useful in segmenting scans corrupted by salt and pepper noise.

Section 9.5 is devoted to the description of geometrical methods and their application in image segmentation. Among many methods used for shape recovery, the level sets has proven to be a successful tool. The level set is a method for capturing moving fronts introduced by Osher and Sethian in 1987. It was used in many applications like fluid dynamics, graphics, visualization, image processing, and computer vision. In this chapter, we introduce an overview of the level set and its use in image segmentation with application in vascular segmentation. The human cerebrovascular system is a complex three-dimensional anatomical structure. Serious types of vascular diseases such as carotid stenosis, aneurysm, and vascular malformation may lead to brain stroke, which are the third leading cause of death and the main cause of disability. An accurate model of the vascular system from MRA data volume is needed to detect these diseases at early stages and hence may prevent invasive treatments. In this section, we will use

a method based on level sets and statistical models to improve the accuracy of the vascular segmentation.

9.2 Stochastic Image Models

The objective of modeling in image analysis is to capture the intrinsic character of images in a few parameters so as to understand the nature of the phenomenon generating the images. Image models are also useful in quantitatively specifying natural constraints and general assumptions about the physical world and the imaging process. The introduction of stochastic models in image analysis has led to the development of many practical algorithms that would not have been realized with ad hoc processing. Approaching problems in image analysis from the modeling viewpoint, we focus on the key issues of model selection, sampling, parameter estimation, and goodness-of-fit.

Formal mathematical image models have long been used in the design of image algorithms for applications such as compression, restoration, and enhancement [1]. Such models are traditionally low stochastic models of limited complexity. In recent years, however, important theoretical advances and increasingly powerful computers have led to more complex and sophisticated image models. Depending on the application, researchers have proposed both low-level and high-level models.

Low-level image models describe the behavior of individual image pixels relative to one another. Markov random fields and other spatial interaction models have proven useful for a variety of applications, including image segmentation and restoration [2, 3]. Bouman et al. [4], along with Willsky and Benveniste [5, 6], have developed multiscale stochastic models for image data.

High-level models are generally used to describe a more restrictive class of images. These models explicitly describe larger structures in the image, rather than describing individual pixel interactions. Grenander et al., for example, propose a model based on deformable templates to describe images of nonrigid objects [7], while Kopec and his colleagues model document images using a Markov source model for symbol generation in conjunction with a noisy channel [8, 9].

The following part of this chapter is organized as follows: First, a short introduction about Gibbs random field (GRF) and Markov random field (MRF)

is given. A detailed description of our proposed approach to get an accurate image model is then presented. Finally, we will apply the proposed model in the segmentation of lung CT.

9.2.1 Statistical Framework

The observed image is assumed to be a composites of two random process: a high-level process X , which represents the classes that form the observed image; and a low-level process Y , which describes the statistical characteristics of each class.

The high-level process X is a random field defined on a rectangular grid S of N^2 points, and the value of X will be written as X_s . Points in X will take values in the set $(\Gamma_1, \dots, \Gamma_m)$, where m is the number of classes in the given image.

Given x , the conditional density function of y is assumed to exist and to be strictly positive and is denoted by $p(Y = y | X = x)$ or $p(y | x)$.

Finally, an image is a square grid of pixels, or sites, $\{(i, j) : i = 1 \text{ to } N, j = 1 \text{ to } N\}$. We adopt a simple numbering of sites by assigning sequence number $t = j + N(i - 1)$ to site s . This scheme numbers the sites row by row from 1 to N^2 , starting in the upper left.

9.2.2 Gibbs Random Fields

In 1987, Boltzmann investigated the distribution of energy states in molecules of an ideal gas. According to the Boltzmann distribution, the probability of a molecule to be in a state with energy ε is

$$p(\varepsilon) = \frac{1}{Z} e^{-\frac{1}{KT}\varepsilon}, \quad (9.1)$$

where Z is a normalization constant, that makes the sum of probabilities equal to 1. T is the absolute temperature, and K is Boltzmann's constant. For simplicity we assume that the temperature is measured in energy units, hence KT will be replaced by T .

Gibbs used a similar distribution in 1901 to express the probability of a whole system with many degrees of freedom to be in a state with a certain energy. A discrete GRF provides a global model for an image by specifying a probability

mass function in the following form

$$p(x) = \frac{1}{Z} e^{-\frac{E(x)}{T}}, \quad (9.2)$$

where $Z = \sum_{x \in \Omega} e^{-\frac{E(x)}{T}}$, and the function $E(x)$ is called energy function.

9.2.3 Markov Random Fields

Hassner and Sklansky introduced Markov random fields to image analysis and throughout the last decade Markov random fields have been used extensively as representations of visual phenomena. A Gibbs random field describes the global properties of an image in terms of the joint distributions of colors for all pixels. An MRF is defined in terms of local properties. Before we show the basic properties of MRF, we will show some definitions related to Gibbs and Markov random fields [10–15].

Definition 1: A clique Λ is a subset of S for which every pair of sites is a neighbor. Single pixels are also considered cliques. The set of all cliques on a grid is called Λ .

Definition 2: A random field X is an MRF with respect to the neighborhood system $\eta = \{\eta_s, s \in S\}$ if and only if

- $p(X = x) > 0$ for all $x \in \Omega$, where Ω is the set of all possible configurations on the given grid;
- $p(X_s = x_s | X_{s|r} = x_{s|r}) = p(X_s = x_s | X_{\partial s} = x_{\partial s})$, where $s|r$ refers to all N^2 sites excluding site r , and ∂s refer to the neighborhood of site s ;
- $p(X_s = x_s | X_{\partial s} = x_{\partial s})$ is the same for all sites s .

The structure of the neighborhood system determines the order of the MRF. For a first-order MRF the neighborhood of a pixel consists of its four nearest neighbors. In a second-order MRF the neighborhood consists of the eight nearest neighbors. The cliques structure are illustrated in Figs 9.1 and 9.2.

Consider a graph (t, η) as shown in Fig. 9.3 having a set of N^2 sites. The energy function for a pairwise interaction model can be written in the following form:

$$E(x) = \sum_{t=1}^{N^2} F(x_t) + \sum_{t=1}^{N^2} \sum_{r=1}^w H(x_t, x_{t:r}), \quad (9.3)$$

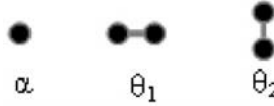


Figure 9.1: Cliques for a first-order neighborhood, where α , θ_1 , and θ_2 are the cliques coefficients for first-order neighborhood system.



Figure 9.2: Cliques for a second-order neighborhood, where α , $\theta_1, \dots, \theta_9$ are the cliques coefficients for second-order neighborhood system.

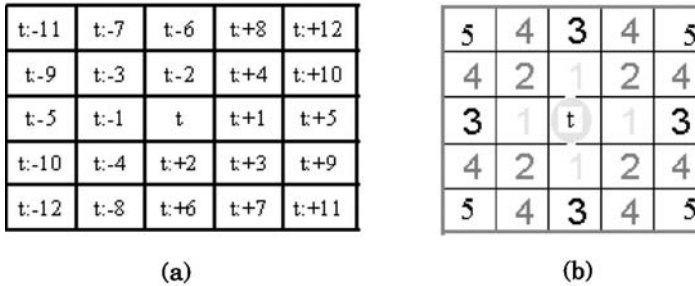


Figure 9.3: Numbering and order coding of the neighborhood structure.

where F is the potential function for single-pixel cliques and H is the potential function for all cliques of size 2. The parameter w depends on the size of the neighborhood around each site. For example, w is 2, 4, 6, 10, and 12 for neighborhoods of orders 1, 2, 3, 4, 5, respectively.

Using the Derin–Elliott model [15] to compute F and H , we have

$$F(x_t) = \alpha x_t \quad \text{and} \quad H(x_t, x_{t:+r}) = \theta_r I(x_t, x(t : +r)),$$

where $I(a, b)$ is called indicator function where

$$\begin{aligned} I(a, b) &= -1 \quad \text{if } a = b \\ &= 1 \quad \text{if } a \neq b. \end{aligned}$$

9.2.4 Image Models

As mentioned before, the observed image is modeled as a composite of two random processes, a high-level process X and a low-level process Y [16–20].

The maximum a posteriori parameters estimation involves the determination of x that maximizes $p(x|y)$ with respect to x . By Bayes' rule,

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}. \quad (9.4)$$

Since the denominator of Eq. 9.4 does not affect the optimization, the MAP parameters estimation can be obtained, equivalently, by maximizing the numerator of Eq. 9.4 or its natural logarithm; that is, we need to find x which maximizes the following criterion:

$$L(x|y) = \ln p(y|x) + \ln p(x). \quad (9.5)$$

The first term in Eq. 9.5 is the likelihood due to the low-level process and the second term is due to the high-level process. Based on the models of the high-level and low-level processes, the MAP estimate can be obtained.

In order to carry out the MAP parameters estimation in Eq. 9.5, one needs to specify the parameters of the two processes. A popular model for the high-level process is the Gibbs Markov model. In the following sections we introduce a new accurate model to model the low-level process. In this model we will assume that each class consists of a mixture of normal distributions which follow the following equation:

$$p(y|\Gamma_i) = \sum_{l=1}^{n_i} \pi_l p(y|C_l), \quad \text{for } i = 1, 2, \dots, m, \quad (9.6)$$

where n_i is the number of normal components that formed class Γ_i , π is the corresponding mixing proportion, and $\{C_l\}_{l=1}^{n_i}$ is the number of Gaussian components that formed class Γ_i . So the overall model for the low-level process can be expressed as follows:

$$p_{es}(y) = \sum_{i=1}^m p(\Gamma_i) p(y|\Gamma_i). \quad (9.7)$$

In our proposed algorithm the priori probability $p(\Gamma_i)$ is included in the mixing proportion for each class.

9.2.5 Parameter Estimation for Low-Level Process

In order to estimate the parameters for low-level process, we need to estimate the number of Gaussian components that formed the distribution for each class, their means, the variances, and mixing proportions for each Gaussian

component. To estimate the distribution for each class, we use the expectation maximization algorithm. The first step to estimate the distribution for each class is to estimate the dominant Gaussian components in the given empirical distribution.

9.2.5.1 Dominant Gaussian Components Extracting Algorithm

1. Assume the number of Gaussian components that represent the classes $\Gamma_i, i = 1, \dots, m$. Initialize the parameters of each distribution randomly.
2. **The E-step:** Compute δ_{it} that represent responsibility that the given pixel value is extracted from certain distribution as

$$\delta_{it}^k = \frac{\pi_i^k p(y_t | \Theta_i^k, \Gamma_i)}{\sum_{l=1}^m \pi_l^k p(y_t | \Theta_l^k, \Gamma_l)}, \quad \text{for } t = 1 \text{ to } N^2, \quad (9.8)$$

where y_t is the gray level at location t in the given image, π_i^k is the mixing proportion of Gaussian component i at step k , and Θ_i^k is estimated parameter for Gaussian component i at step k .

3. **The M-step:** we compute the new mean, the new variance, and the new proportion from the following equations:

$$\pi_i^{k+1} = \sum_{t=1}^{N^2} \delta_{it}^k, \quad (9.9)$$

$$\mu_i^{k+1} = \frac{\sum_{t=1}^{N^2} \delta_{it}^k y_t}{\sum_{t=1}^{N^2} \delta_{it}^k}, \quad (9.10)$$

$$(\sigma_i^{k+1})^2 = \frac{\sum_{t=1}^{N^2} \delta_{it}^k (y_t - \mu_i^k)^2}{\sum_{t=1}^{N^2} \delta_{it}^k}. \quad (9.11)$$

4. Repeat steps 1 and 2 until the relative difference of the subsequent values of Eqs. 9.9, 9.10, and 9.11 are sufficiently small.

Let $p_{I\Gamma 1}(y), p_{I\Gamma 2}(y), \dots, p_{I\Gamma m}(y)$ be the dominant Gaussian components that are estimated from the above algorithm. Then the initial estimated density ($p_I(y)$) for the given image can be defined as follows:

$$p_I(y) = \pi_1 p_{I\Gamma 1}(y) + \pi_2 p_{I\Gamma 2}(y) + \dots + \pi_m p_{I\Gamma m}(y). \quad (9.12)$$

Because the empirical data does not exactly follow mixture of normal distribution, there will be error between $p_I(y)$ and $p_{em}(y)$. So we suggest the following

models for the empirical data:

$$p_{em}(y) = p_I(y) + \zeta(y), \quad (9.13)$$

where $\zeta(y)$ represent the error between $p_{em}(y)$ and $p_I(y)$. From Eq. 9.13, $\zeta(y)$ can be rewritten as

$$\zeta(y) = |p_{em}(y) - p_I(y)| \text{sign}(p_{em}(y) - p_I(y)). \quad (9.14)$$

We assume that the absolute value of $\zeta(y)$ is another density which consists of a mixture of normal distributions and we will use the following EM algorithm to estimate the number of Gaussian components in $\zeta(y)$ and the mean, the variance, and mixing proportion.

9.2.5.2 Sequential EM Algorithm

1. Assume the number of Gaussian components (n) in $\zeta(y)$ is 2.
2. **The E-step:** Given the current value of the number of Gaussian components in $\zeta(y)$, compute δ_{it} as

$$\delta_{it}^k = \frac{\pi_i^k p(y_t | \Theta_i^k)}{\sum_{l=1}^n \pi_l^k p(y_t | \Theta_l^k)}, \quad \text{for } i = 1 \text{ to } n \text{ and } t = 1 \text{ to } N^2. \quad (9.15)$$

3. **The M-step:** We compute the new mean, the new variance, and the new proportion from the following equations:

$$\pi_i^{k+1} = \sum_{t=1}^{N^2} \delta_{it}, \quad (9.16)$$

$$\mu_i^{k+1} = \frac{\sum_{t=1}^{N^2} \delta_{it}^k y_t}{\sum_{t=1}^{N^2} \delta_{it}^k}, \quad (9.17)$$

$$(\sigma_i^{k+1})^2 = \frac{\sum_{t=1}^{N^2} \delta_{it}^k (y_t - \mu_i^k)^2}{\sum_{t=1}^{N^2} \delta_{it}^k}. \quad (9.18)$$

4. Repeat steps 1 and 2 until the relative differences of the subsequent values of Eqs. 9.16, 9.17, and 9.18 are sufficiently small.
5. Compute the conditional expectation and the error between $|\zeta(y)|$ and the estimated density ($p_\zeta(y)$) for $|\zeta(y)|$ from the following equations:

$$Q(n) = \sum_{t=1}^{N^2} \sum_{i=1}^n \delta_{it} \ln p_\zeta(y | \Theta_i), \quad (9.19)$$

$$\epsilon(n) = |\zeta(y)| - \sum_{i=1}^n \pi_i p_{\zeta i}(y). \quad (9.20)$$

6. Repeat steps 2, 3, 4, and 5, and increase the number of Gaussian components n by 1 if the conditional expectation $Q(n)$ is still increasing and $\epsilon(n)$ is still decreasing, otherwise stop and select the parameters which correspond to maximum $Q(n)$ and minimum $\epsilon(n)$.

Since EM algorithm can be trapped in a local minimum, we run the above algorithm several times and select the number of Gaussian components and their parameters that give maximum $Q(n)$ and minimum $\epsilon(n)$.

After we determined the number of Gaussian components that formed $|\zeta(y)|$, we need to determine which components belong to class Γ_1 , and belong to class Γ_2 , and so on. In this model we classify these components based on the minimization of risk function under 0–1 loss. In order to minimize the risk function, we can use the following algorithm. Note that the following algorithm is written for two classes but it is easy to generalize to n classes.

9.2.5.3 Components Classification Algorithm

1. All Gaussian components that have mean less than the estimated mean for $p_{\Gamma_1}(y)$ belong to the first class.
2. All Gaussian components that have mean greater than the estimated mean for $p_{\Gamma_2}(y)$ belong to the second class.
3. For the components which have mean greater than the estimated mean for $p_{\Gamma_1}(y)$ and less than the estimated mean for $p_{\Gamma_2}(y)$, do the following:
 - (a) Assume that the first component belongs to the first class and the other components belong to the second class. Compute the risk value from the following equation:

$$R(Th) = \int_{Th}^{\infty} p(y|\Gamma_1)dy + \int_{-\infty}^{Th} p(y|\Gamma_2)dy, \quad (9.21)$$

where Th is the threshold that separates class Γ_1 from class Γ_2 . The above integration can be done using a second-order spline.

- (b) Assume that the first and second components belong to the first class and other components belong to the second class, and from Eq. 9.21 compute $R(Th)$. Continue this process as $R(Th)$ decreases, and stop when $R(Th)$ starts to increase.

Finally, to show the convergence of the proposed model, we will show experimentally, when we use this model, the Levy distance will decrease between the estimated distribution $P_{es}(y)$ and empirical distribution $P_{em}(y)$. The Levy distance $\rho(P_{em}, P_{es})$ is defined as

$$\rho(P_{em}, P_{es}) = \inf\{\xi > 0 : \forall y P_{em}(y - \xi) - \xi \leq P_{es}(y) \leq P_{em}(y + \xi) + \xi\}. \quad (9.22)$$

As $\rho(P_{em}, P_{es})$ approach zero, $P_{es}(y)$ converge weakly to $P_{em}(y)$.

9.2.6 Parameter Estimation for High-Level Process

In order to carry out the MAP parameters estimation in Eq. 9.5, one needs to specify the parameters of high-level process. A popular model for the high-level process is the Gibbs Markov model which follows Eq. 9.2. In order to estimate the parameters of GMRF, we will find the parameters that maximize Eq. 9.2, and we will use the Metropolis algorithm and genetic algorithm (GA).

The Metropolis algorithm is a relaxation algorithm to find a global maximum. The algorithm assumes that the classes of all neighbors of y are known. The high-level process is assumed to be formed of m -independent processes; each of the m processes is modeled by Gibbs Markov random which follow Eq. 9.2. Then y can be classified using the fact that $p(x_i|y)$ is proportional to $p(y|x_i) P(x_i|\eta_s)$, where s is the neighbor set to site S belonging to class x_i , $p(x_i|\eta_s)$ is computed from Eq. 9.2, and $p(y|x_i)$ is computed from the estimated density for each class.

By using the Bayes classifier, we get initial labeling image. In order to run the Metropolis algorithm, first we must know the coefficients of potential function $E(x)$, so we will use GA to estimate the coefficient of $E(x)$ and evaluate these coefficients through the fitness function.

9.2.6.1 Maximization Using Genetic Algorithm

To build the genetic algorithm, we define the following parameters:

Chromosome: A chromosome is represented in binary digits and consists of representations for model order and clique coefficients. Each chromosome has 41 bits. The first bit represent the order of the system (we use digit “0” for first-order and digit “1” for second-order-GMRF). The remaining bits represent the

clique coefficients, where each clique coefficient is represented by 4 bits (note that for first-order system, we estimate only five parameters, and the remaining clique's coefficient will be zero, but for the second-order system we will estimate ten parameters).

Fitness Function: Since our goal is to select the high-level process X that maximize Eq. 9.5, we can use Eq. 9.5 as the fitness function.

High-level parameters estimation algorithm:

1. Generate the first generation which consists of 30 chromosomes.
2. Apply the Metropolis algorithm for each chromosome on each image and then compute the fitness function as shown in Eq. 9.5.
3. If the fitness values for all chromosomes do not change from one population to another population, then stop and select the chromosome, which gives maximum fitness value. (If there are two chromosomes that give the same fitness value, we select the chromosome which represents lower order system.) Otherwise go to step 2.

Using the results obtained by this algorithm, we will repeat the estimation of low-level process and high-level process. We will stop when the difference between the current parameters and previous parameters is small.

9.3 Applications

Lung Cancer remains the leading cause of mortality cancer. In 1999, there were approximately 170 000 new cases of lung cancer [21]. The 5-year survival rate from the diseases is 14% and has increased only slightly since the early 1970s despite extensive and expensive research work to find effective therapy. The disparity in survival between early and late-stage lung cancer is substantial, with a 5-year survival rate of approximately 70% in stage 1A disease compared to less than 5% in stage IV disease according to the recently revised lung cancer staging criteria [21]. The disproportionately high prevalence of and mortality from lung cancer has encouraged attempts to detect early lung cancer with screening programs aimed at smokers. Smokers have an incidence rate of lung

cancer that is ten times that of nonsmokers and accounts for greater than 80% of lung cancer cases in the United States [21].

One in every 18 women and every 12 men develop lung cancer, making it the leading cause of cancer deaths. Early detection of lung tumors (visible on the chest film as nodules) may increase the patient's chance of survival. For this reason the Jewish Hospital designed a program for early detection with the following specific aims: A number of lung cancer screening trials have been conducted in the United States, Japan, and Europe for the purpose of developing an automatic approach of tumor detection [21].

At the University of Louisville CVIP Lab a long-term effort has been ensued to develop a comprehensive image analysis system to detect and recognize lung nodules in low dose chest CT (LDCT) scans. The LDCT scanning was performed with the following parameters: slice thickness of 8 mm reconstructed every 4 mm and scanning pitch of 1.5. In the following section we highlight our approach for automatic detection and recognition of lung nodules; further details can be found in [22].

9.3.1 Lung Extraction

The goal of lung extraction is to separate the voxels corresponding to lung tissue from those belonging to the surrounding anatomical structures. We assume that each slice consists of two types of pixels: lung and other tissues (e.g., chest, ribs, and liver). The problem in lung segmentation is that there are some tissues in the lung such as arteries, veins, bronchi, and bronchioles having gray level close to the gray level of the chest. Therefore, in this application if we depend only on the gray level we lose some of the lung tissues during the segmentation process. Our proposed model which depends on estimating parameters for two processes (high-level process and low-level process) is suitable for this application because the proposed model not only depend on the gray level but also takes into consideration the characterization of spatial clustering of pixels into regions.

We will apply the approach that was described in Section 9.2.4 on lung CT. Figure 9.4 shows a typical CT slice for the chest. We assume that each slice consists of two types of tissues: lung and other tissues (e.g., chest, ribs, and liver). As discussed above, we need to estimate parameters for both low-level process and high-level process. Table 9.1 presents the results of applying the

Table 9.1: Estimated using dominant Gaussian components extracting algorithm

Parameter	$\mu_{I\Gamma 1}$	$\mu_{I\Gamma 2}$	$\sigma_{I\Gamma 1}^2$	$\sigma_{I\Gamma 2}^2$	$\pi_{I\Gamma 1}$	$\pi_{I\Gamma 2}$
Value	59.29	139.97	177.15	344.29	0.25	0.758

dominant Gaussian components extracting algorithm described in 9.2.5.1. Figure 9.5 shows the empirical density for the CT slice shown in Fig. 9.4 and the initial estimated density (which represented the two dominant Gaussian components in the given CT). The Levy distance between the two distribution functions which represented the densities shown in Fig. 9.5 is 0.09. This value is large and this means there is a mismatch between empirical $p_{em}(y)$ and $p_I(y)$. Figure 9.6 shows the error and absolute error between $p_{em}(y)$ and $p_I(y)$.

After we apply sequential EM algorithm to $|\zeta(y)|$, we get that the number of normal components that represent $|\zeta(y)|$ is 10 as shown in Fig. 9.7. Figure 9.8

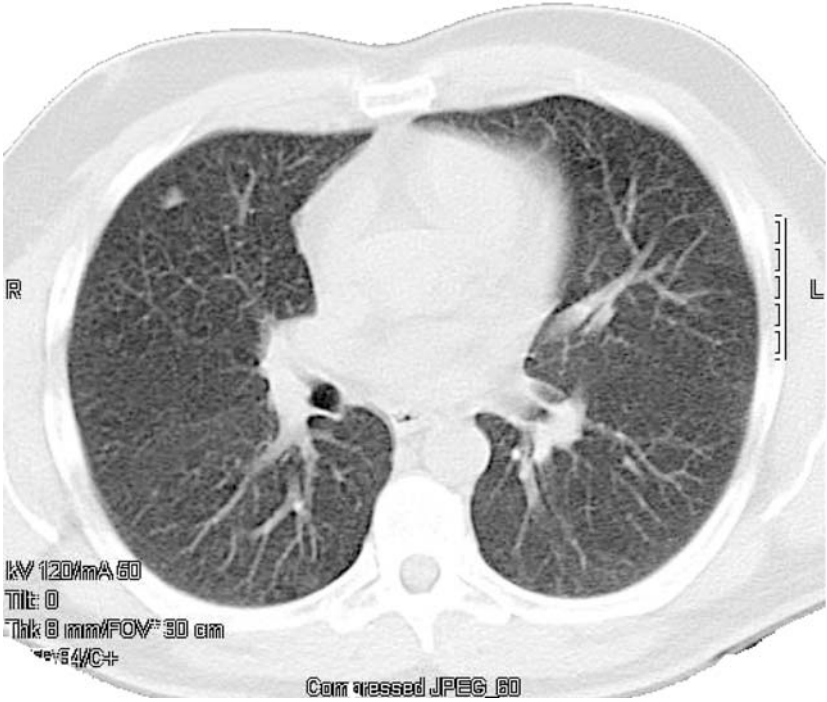


Figure 9.4: A typical slice form of a chest spiral CT scan.

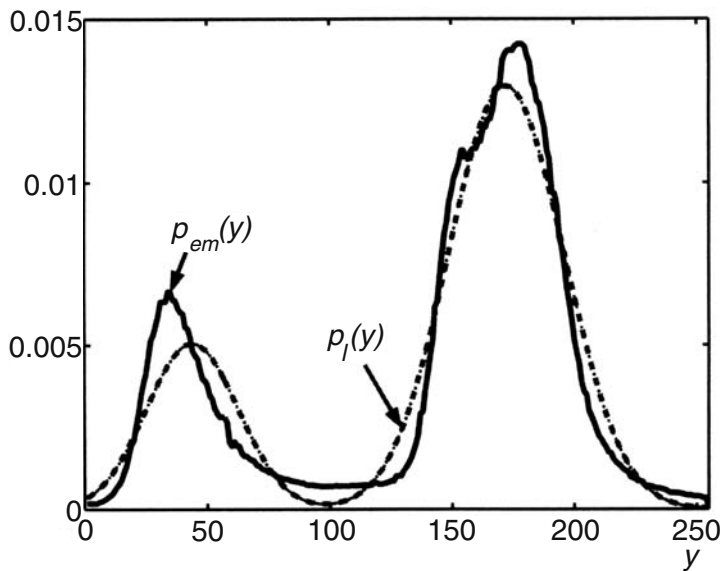


Figure 9.5: Empirical density for given CT slice and initial estimated density.

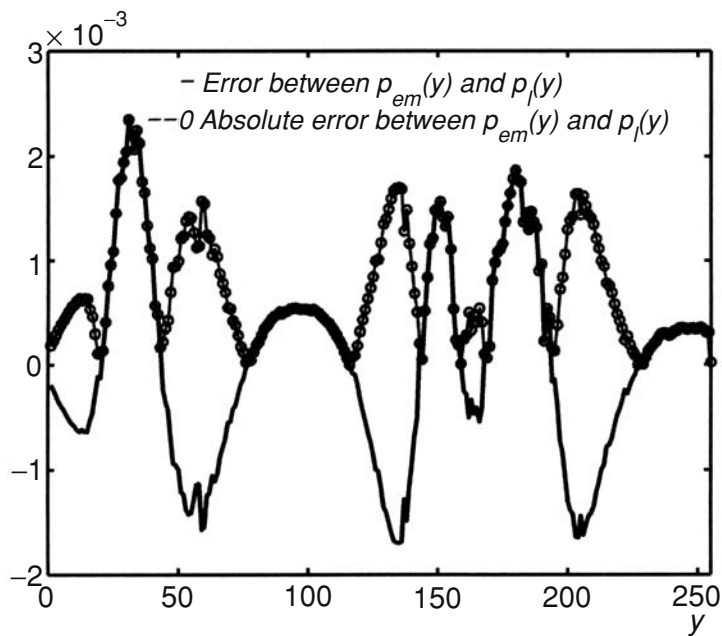


Figure 9.6: Error and absolute error between $p_{em}(Y = y)$ and $p_l(Y = y)$.

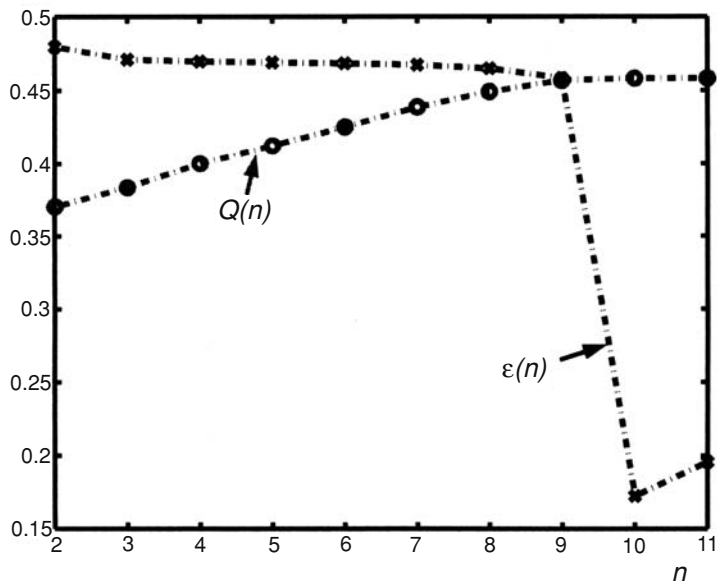


Figure 9.7: Conditional expectation $Q(n)$ and the error function ($\epsilon(n)$) versus the number of Gaussians approximating the scaled absolute deviation in Fig. 9.6.

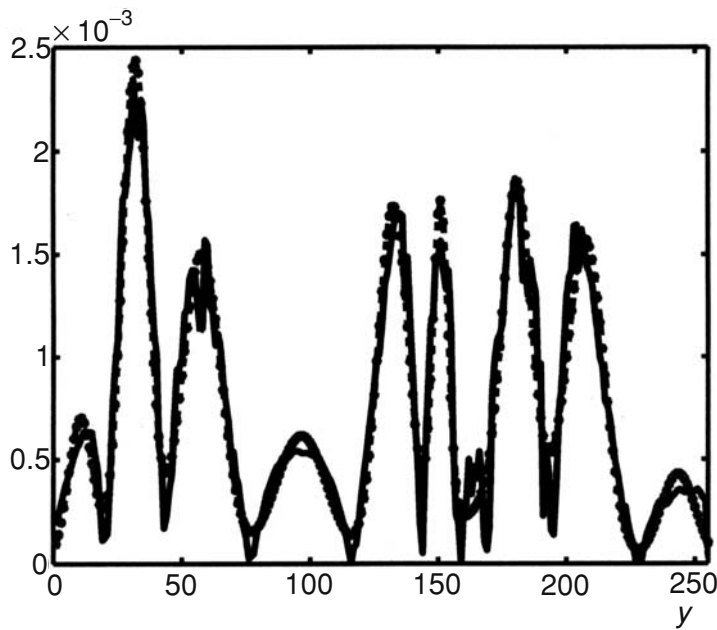


Figure 9.8: Estimated density for $|\eta(Y = y)|$.

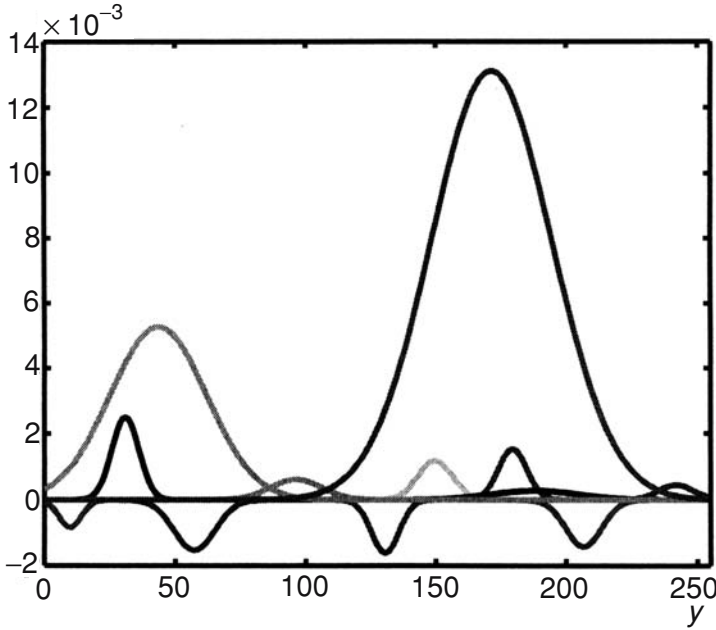


Figure 9.9: 12 Gaussian components which are used in density estimation.

shows the estimated density for $|\zeta(y)|$. Figure 9.9 shows all Gaussian components which are estimated after using dominant Gaussian components extracting algorithm and sequential EM algorithms. Figure 9.10 shows the estimated density for the CT slices shown in Figure 9.4. The Levy distance between the distributions $P_{es}(y)$ and $P_{em}(y)$ is 0.0021 which is smaller compared to the Levy distance between the distributions $P_{em}(y)$ and $PI(y)$.

Now we apply components classification algorithm on the ten Gaussian components that are estimated using sequential EM algorithm in order to determine which components belong to lung tissues and which components belong to chest tissues. The results of components classification algorithm show that the minimum risk equal to 0.00448 occurs at threshold $Th = 108$ when Gaussian components 1, 2, 3, and 4 belong to lung tissues and component 5, 6, 7, 8, 9, and 10 belong to chest tissues. Figure 9.11 shows the estimated density for lung tissues and estimated density for chest and other tissues that may appear in CT.

The next step of our algorithm is to estimate the parameters for high-level process. A popular model for the high-level process is the Gibbs Markov mode, and we use the Bayes classifier to get initial labeling image. After we run Metropolis algorithm and GA to determine the coefficients of potential function $E(x)$, we get

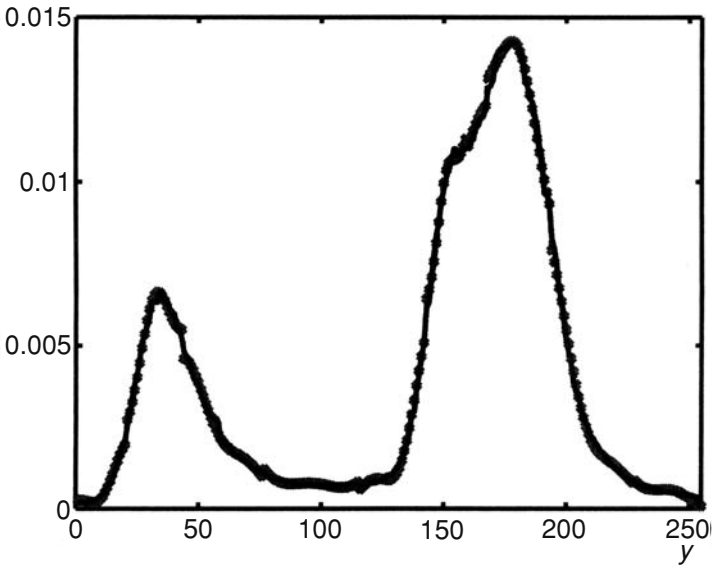


Figure 9.10: Estimated density for lung tissues and chest tissues.

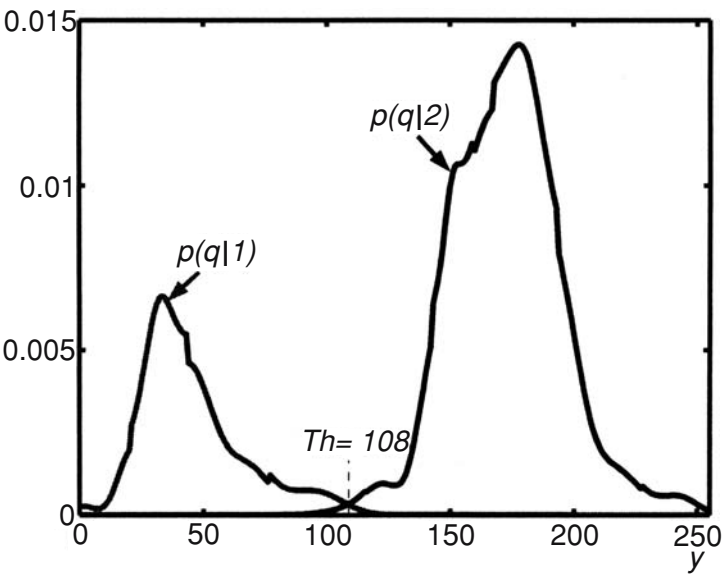


Figure 9.11: Empirical density and estimated density for CT slice shown in Fig. 9.4.

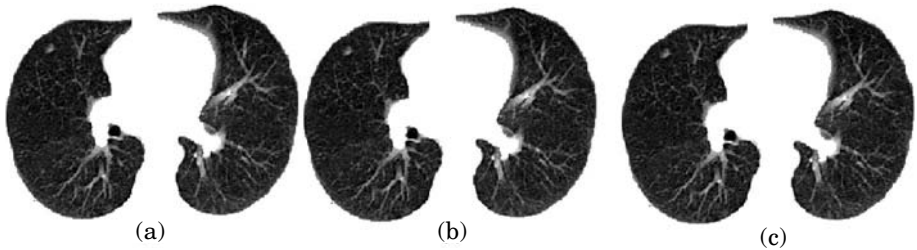


Figure 9.12: (a) Segmented lung using the proposed algorithm, error = 1.09%. (b) Output of segmentation algorithm by selecting parameters for high-level process randomly, error = 1.86%. (c) Segmented lung by radiologist.

the following results: $\alpha = 1$, $\theta_1 = 0.89$, $\theta_2 = 0.8$, $\theta_3 = 0.78$, $\theta_4 = 0.69$, $\theta_5 = 0.54$, $\theta_6 = 0.61$, $\theta_7 = 0.89$, $\theta_8 = 0.56$, and $\theta_9 = 0.99$.

The result of segmentation for the image shown in Fig. 9.4 using these parameters is shown in Fig. 9.12. Figure 9.12(a) shows the results of proposed algorithm. Figure 9.12(b) shows output of the Metropolis algorithm by selecting parameters randomly. Figure 9.12(c) shows the segmentation done by a radiologist.

As shown in Fig. 9.12(a) the accuracy of our algorithm seems good if it is compared with the segmentation of the radiologist. Figure 9.13 shows comparison between our results and the results obtained by iterative threshold method which was proposed by Hu and Hoffman [23]. It is clear from Fig. 9.13 that the

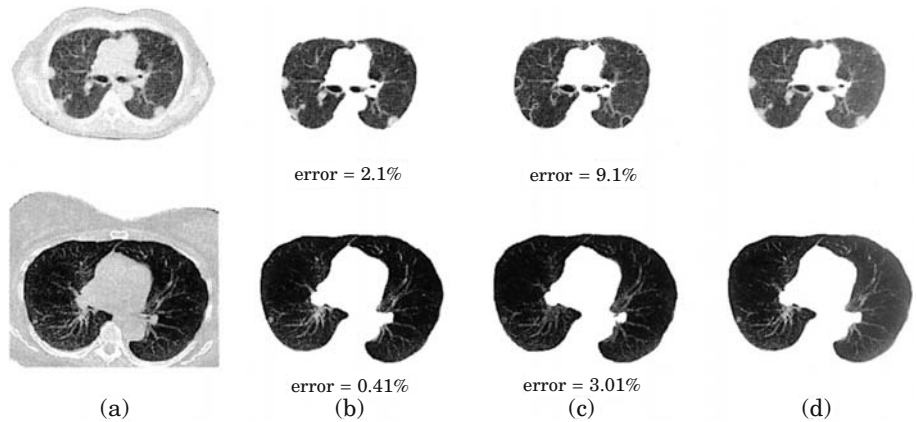


Figure 9.13: (a) Original CT, (b) segmented lung using the proposed model, (c) segmented lung using the iterative threshold method, and (d) segmented lung by radiologist. The errors with respect to this ground truth are highlighted by red color.

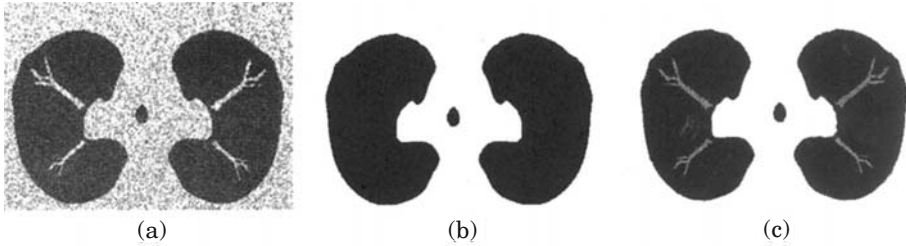


Figure 9.14: (a) Generated Phantom, (b) ground truth image (black pixel represent lung area, and gray pixels represent the chest area), and (c) segmented lung using the proposed approach (error 0.091). The errors with respect to this ground truth are highlighted by red color.

proposed algorithm segments the lung without causing any loss of abnormality tissues if it is compared with the iterative threshold method. Also, in order to validate our results we create a phantom which has the same distribution as lung and chest tissues. This phantom is shown in Fig. 9.14. One of the advantages of this phantom is that we know its ground truth. It is clear from Fig. 9.14 that the error between segmented lung and ground truth is small and this shows that the proposed model is accurate and suitable for this application.

9.4 Fuzzy Segmentation

As mentioned before, the objective of image segmentation is to divide an image into meaningful regions. Errors made at this stage would affect all higher level activities. Therefore, methods that incorporate the uncertainty of object and region definitions and the faithfulness of the features to represent various objects are desirable.

In an ideally segmented image, each region should be homogeneous with respect to some predicate such as gray level or texture, and adjacent regions should have significantly different characteristics or features. More formally, segmentation is the process of partitioning the entire image into c crisp maximally connected regions $\{R_i\}$ such that each R_i is homogeneous with respect to some criteria. In many situations, it is not easy to determine if a pixel should belong to a region or not. This is because the features used to determine homogeneity may not have sharp transitions at region boundaries. To alleviate this situation, we can inset fuzzy set concepts into the segmentation process.

In fuzzy segmentation, each pixel is assigned a membership value in each of the c regions. If the memberships are taken into account while computing properties of regions, we obtain more accurate estimates of region properties. One of the known techniques to obtain such a classification is the FCM algorithm [40, 41]. The FCM algorithm is an unsupervised technique that clusters data by iteratively computing a fuzzy membership function and mean value estimates for each class. The fuzzy membership function, constrained to be between 0 and 1, reflects the degree of similarity between the data value at that location and the prototypical data value, or centroid, of its class. Thus, a high membership value near unity signifies that the data value at that location is close to the centroid of that particular class.

FCM has been used with some success in image segmentation in general [45, 46], however, since it is a point operation, it does not preserve connectivity among regions. Furthermore, FCM is highly sensitive to noise. In the following sections, we will present a new system to segment digital images using a modified Fuzzy c-means algorithm. Our algorithm is formulated by modifying the objective function of the standard FCM algorithm to allow the labeling of a pixel to be influenced by the labels in its immediate neighborhood. The neighborhood effect acts as a regularizer and biases the solution toward piecewise-homogeneous labelings. Such a regularization is useful in segmenting scans corrupted by scanner noise. In this paper, we will present the results of applying this algorithm to segment MRI data corrupted with a multiplicative gain field and salt and pepper noise.

9.4.1 Standard Fuzzy-C-Means

The standard FCM objective function for partitioning $\{x_k\}_{k=1}^N$ into c clusters is given by

$$J = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^p \|x_k - v_i\|^2, \quad (9.23)$$

where $\{x_k\}_{k=1}^N$ are the feature vectors for each pixel, $\{v_i\}_{i=1}^c$ are the prototypes of the clusters and the array $[u_{ik}] = U$ represents a partition matrix, $U \in \mathcal{U}$, namely

$$\mathcal{U} = \{ u_{ik} \in [0, 1] \mid \sum_{i=1}^c u_{ik} = 1 \quad \forall k \}$$

and

$$0 < \sum_{k=1}^N u_{ik} < N \quad \forall i\}. \quad (9.24)$$

The parameter p is a weighting exponent on each fuzzy membership and determines the amount of fuzziness of the resulting classification. The FCM objective function is minimized when high membership values are assigned to pixels whose intensities are close to the centroid of its particular class, and low membership values are assigned when the pixel data is far from the centroid.

9.4.2 Modified Fuzzy C-Means Objective Function

We propose a modification to Eq. 9.23 by introducing a term that allows the labeling of a pixel to be influenced by the labels in its immediate neighborhood. As mentioned before, the neighborhood effect acts as a regularizer and biases the solution toward piecewise-homogeneous labeling. Such a regularization is useful in segmenting scans corrupted by salt and pepper noise. The modified objective function is given by

$$\begin{aligned} J_m = & \sum_{i=1}^c \sum_{k=1}^N u_{ik}^p \|x_k - v_i\|^2 \\ & + \frac{\alpha}{N_R} \sum_{i=1}^c \sum_{k=1}^N u_{ik}^p \left(\sum_{x_r \in \mathcal{N}_k} \|x_r - v_i\|^2 \right), \end{aligned} \quad (9.25)$$

where $\mathcal{N}_k$ stands for the set of neighbors that exist in a window around x_k and N_R is the cardinality of $\mathcal{N}_k$. The effect of the neighbors term is controlled by the parameter α . The relative importance of the regularizing term is inversely proportional to the signal to noise ratio (SNR) of the image signal. Lower SNR would require a higher value of the parameter α .

Formally, the optimization problem comes in the form

$$\min_{U, \{v_i\}_{i=1}^c} J_m \quad \text{subject to} \quad U \in \mathcal{U}. \quad (9.26)$$

9.4.3 Parameter Estimation

The objective function J_m can be minimized in a fashion similar to the standard FCM algorithm. Taking the first derivatives of J_m with respect to u_{ik} and v_i , and

setting them to zero results in two necessary but not sufficient conditions for J_m to be at a local extrema. In the following subsections, we will derive these three conditions.

9.4.3.1 Membership Evaluation

The constrained optimization in Eq. 9.26 will be solved using one Lagrange multiplier

$$F_m = \sum_{i=1}^c \sum_{k=1}^N \left(u_{ik}^p D_{ik} + \frac{\alpha}{N_R} u_{ik}^p \gamma_i \right) + \lambda \left(1 - \sum_{i=1}^c u_{ik} \right), \quad (9.27)$$

where $D_{ik} = \|x_k - v_i\|^2$ and $\gamma_i = (\sum_{x_r \in \mathcal{N}_k} \|x_r - v_i\|^2)$. Taking the derivative of F_m w.r.t. u_{ik} and setting the result to zero, we have, for $p > 1$,

$$\left[\frac{\delta F_m}{\delta u_{ik}} = p u_{ik}^{p-1} D_{ik} + \frac{\alpha p}{N_R} u_{ik}^{p-1} \gamma_i - \lambda \right]_{u_{ik}=u_{ik}^*} = 0. \quad (9.28)$$

Solving for u_{ik}^* , we have

$$u_{ik}^* = \left(\frac{\lambda}{p(D_{ik} + \frac{\alpha}{N_R} \gamma_i)} \right)^{\frac{1}{p-1}}. \quad (9.29)$$

Since $\sum_{j=1}^c u_{jk} = 1 \quad \forall k$,

$$\sum_{j=1}^c \left(\frac{\lambda}{p(D_{jk} + \frac{\alpha}{N_R} \gamma_j)} \right)^{\frac{1}{p-1}} = 1 \quad (9.30)$$

or

$$\lambda = \frac{p}{\left(\sum_{j=1}^c \left(\frac{1}{(D_{jk} + \frac{\alpha}{N_R} \gamma_j)} \right)^{\frac{1}{p-1}} \right)^{p-1}} \quad (9.31)$$

Substituting into Eq. 9.29, the zero-gradient condition for the membership estimator can be rewritten as

$$u_{ik}^* = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ik} + \frac{\alpha}{N_R} \gamma_i}{D_{jk} + \frac{\alpha}{N_R} \gamma_j} \right)^{\frac{1}{p-1}}}. \quad (9.32)$$

9.4.3.2 Cluster Prototype Updating

Using the standard Euclidean distance and taking the derivative of F_m w.r.t. v_i and setting the result to zero, we have

$$\left[\sum_{k=1}^N u_{ik}^p (x_k - v_i) + \sum_{k=1}^N u_{ik}^p \frac{\alpha}{N_R} \sum_{y_r \in \mathcal{N}_k} (x_r - v_i) \right]_{v_i=v_i^*} = 0. \quad (9.33)$$

Solving for v_i , we have

$$v_i^* = \frac{\sum_{k=1}^N u_{ik}^p \left((x_k) + \frac{\alpha}{N_R} \sum_{x_r \in \mathcal{N}_k} (x_r) \right)}{(1 + \alpha) \sum_{k=1}^N u_{ik}^p}. \quad (9.34)$$

9.4.4 Application: Adaptive MRI Segmentation

In this section, we describe the application of the MFCM segmentation on MRI images having intensity inhomogeneity. Spatial intensity inhomogeneity induced by the radio frequency (RF) coil in magnetic resonance imaging (MRI) is a major problem in the computer analysis of MRI data [24–27]. Such inhomogeneities have rendered conventional intensity-based classification of MR images very difficult, even with advanced techniques such as nonparametric, multichannel methods [28–30]. This is due to the fact that the intensity inhomogeneities appearing in MR images produce spatial changes in tissue statistics, i.e. mean and variance. In addition, the degradation on the images obstructs the physician's diagnoses because the physician has to ignore the inhomogeneity artifact in the corrupted images [31].

The removal of the spatial intensity inhomogeneity from MR images is difficult because the inhomogeneities could change with different MRI acquisition parameters from patient to patient and from slice to slice. Therefore, the correction of intensity inhomogeneities is usually required for each new image. In the last decade, a number of algorithms have been proposed for the intensity inhomogeneity correction. Meyer *et al.* [32] presented an edge-based segmentation scheme to find uniform regions in the image followed by a polynomial surface fit to those regions. The result of their correction is, however, very dependent on the quality of the segmentation step.

Several authors have reported methods based on the use of phantoms for intensity calibration. Wicks *et al.* [26] proposed methods based on the signal

produced by a uniform phantom to correct for MRI images of any orientation. Similarly, Tincher *et al.* [33] modeled the inhomogeneity function by a second-order polynomial and fitted it to a uniform phantom-scanned MR image. These phantom approaches, however, have the drawback that the geometry relationship of the coils and the image data is typically not available with the image data. They also require the same acquisition parameters for the phantom scan and the patient. In addition, these approaches assume the intensity corruption effects are the same for different patients, which is not valid in general [31].

The homomorphic filtering approach to remove the multiplicative effect of the inhomogeneity has been commonly used due to its easy and efficient implementation [29, 34]. This method, however, is effective only on images with relatively low contrast. Some researchers [33, 35] reported undesirable artifacts with this approach.

Dawant *et al.* [35] used operator-selected reference points in the image to guide the construction of a thin-plate spline correction surface. The performance of this method depends substantially on the labeling of the reference points. Considerable user interactions are usually required to obtain good correction results. More recently, Gilles *et al.* [36] proposed an automatic and iterative B-spline fitting algorithm for the intensity inhomogeneity correction of breast MR images. The application of this algorithm is restricted to MR images with a single dominant tissue class, such as breast MR images. Another polynomial surface fitting method [37] was proposed based on the assumption that the number of tissue classes, the true means, and standard deviations of all the tissue classes in the image are given. Unfortunately, the required statistical information is usually not available.

A different approach used to segment images with intensity inhomogeneities is to simultaneously compensate for the shading effect while segmenting the image. This approach has the advantage of being able to use intermediate information from the segmentation while performing the correction. Recently, Wells *et al.* [28] developed a new statistical approach based on the EM algorithm to solve the bias field correction problem and the tissue classification problem. Guillemaud *et al.* [38] further refined this technique by introducing the extra class “other.” There are two main disadvantages of this EM approach. First, the EM algorithm is extremely computationally intensive, especially for large

problems. Second, the EM algorithm requires a good initial guess for either the bias field or the classification estimate. Otherwise, the EM algorithm could be easily trapped in a local minimum, resulting in an unsatisfactory solution [31].

Another approach based on the FCM [40, 41] clustering technique has been introduced lately [42–44]. FCM has been used with some success in image segmentation in segmenting MR images [42, 47, 50]. Xu *et al.* [42] proposed a new adaptive FCM technique to produce fuzzy segmentation while compensating for intensity inhomogeneities. Their method, however, is also computationally intensive. They reduced the computational complexity by iterating on a coarse grid rather than the fine grid containing the image. This introduced some errors in the classification results and was found to be sensitive to a considerable amount of salt and pepper noise [43].

To solve the problem of noise sensitivity and computational complexity of the Pham and Prince method, we will generalize the MFCM algorithm to segment MRI data in the presence of intensity inhomogeneities.

9.4.4.1 Signal Modeling

The observed MRI signal is modeled as a product of the true signal generated by the underlying anatomy and a spatially varying factor called the gain field:

$$Y_k = X_k G_k \quad \forall k \in [1, N] \quad (9.35)$$

where X_k and Y_k are the true and observed intensities at the k th voxel, respectively, G_k is the gain field at the k th voxel, and N is the total number of voxels in the MRI volume.

The application of a logarithmic transformation to the intensities allows the artifact to be modeled as an additive bias field [28]

$$y_k = x_k + \beta_k \quad \forall k \in [1, N], \quad (9.36)$$

where x_k and y_k are the true and observed log-transformed intensities at the k th voxel, respectively, and β_k is the bias field at the k th voxel. If the gain field is known, it is relatively easy to estimate the tissue class by applying a conventional intensity-based segmenter to the corrected data. Similarly, if the tissue classes are known, we can estimate the gain field, but it may be problematic to estimate

either without the knowledge of the other. We will show that by using an iterative algorithm based on fuzzy logic, we can estimate both.

9.4.4.2 Bias Corrected Fuzzy C-means (BCFCM) Objective Function

Substituting Eq. 9.36 into Eq. 9.25, we have

$$J_m = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^p \|y_k - \beta_k - v_i\|^2 + \frac{\alpha}{N_R} \sum_{i=1}^c \sum_{k=1}^N u_{ik}^p \left(\sum_{y_r \in \mathcal{N}_k} \|y_r - \beta_r - v_i\|^2 \right). \quad (9.37)$$

Formally, the optimization problem comes in the form

$$\min_{U, \{v_i\}_{i=1}^c, \{\beta_k\}_{k=1}^N} J_m \quad \text{subject to} \quad U \in \mathcal{U}. \quad (9.38)$$

9.4.4.3 BCFCM Parameter Estimation

The objective function J_m can be minimized in a fashion similar to the MFCM algorithm. Taking the first derivatives of J_m with respect to u_{ik} , v_i , and β_k and setting them to zero results in three necessary but not sufficient conditions for J_m to be at a local extrema. In the following subsections, we will derive these three conditions.

9.4.4.4 Membership Evaluation

Similar to the MFCM algorithm, the constrained optimization in Eq. 9.38 will be solved using one Lagrange multiplier

$$F_m = \sum_{i=1}^c \sum_{k=1}^N \left(u_{ik}^p D_{ik} + \frac{\alpha}{N_R} u_{ik}^p \gamma_i \right) + \lambda \left(1 - \sum_{i=1}^c u_{ik} \right) \quad (9.39)$$

where $D_{ik} = \|y_k - \beta_k - v_i\|^2$ and $\gamma_i = \left(\sum_{y_r \in \mathcal{N}_k} \|y_r - \beta_r - v_i\|^2 \right)$. The zero-gradient condition for the membership estimator can be written as

$$u_{ik}^* = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ik} + \frac{\alpha}{N_R} \gamma_i}{D_{jk} + \frac{\alpha}{N_R} \gamma_j} \right)^{\frac{1}{p-1}}}. \quad (9.40)$$

9.4.4.5 Cluster Prototype Updating

Taking the derivative of F_m w.r.t. v_i and setting the result to zero, we have

$$\left[\sum_{k=1}^N u_{ik}^p (y_k - \beta_k - v_i) + \sum_{k=1}^N u_{ik}^p \frac{\alpha}{N_R} \sum_{y_r \in \mathcal{N}_k} (y_r - \beta_r - v_i) \right]_{v_i=v_i^*} = 0. \quad (9.41)$$

Solving for v_i , we have

$$v_i^* = \frac{\sum_{k=1}^N u_{ik}^p \left((y_k - \beta_k) + \frac{\alpha}{N_R} \sum_{y_r \in \mathcal{N}_k} (y_r - \beta_r) \right)}{(1 + \alpha) \sum_{k=1}^N u_{ik}^p}. \quad (9.42)$$

9.4.4.6 Bias Field Estimation

In a similar fashion, taking the derivative of F_m w.r.t. β_k and setting the result to zero we have

$$\left[\sum_{i=1}^c \frac{\partial}{\partial \beta_k} \sum_{k=1}^N u_{ik}^p (y_k - \beta_k - v_i)^2 \right]_{\beta_k=\beta_k^*} = 0. \quad (9.43)$$

Since only the k th term in the second summation depends on β_k , we have

$$\left[\sum_{i=1}^c \frac{\partial}{\partial \beta_k} u_{ik}^p (y_k - \beta_k - v_i)^2 \right]_{\beta_k=\beta_k^*} = 0. \quad (9.44)$$

Differentiating the distance expression, we obtain

$$\left[y_k \sum_{i=1}^c u_{ik}^p - \beta_k \sum_{i=1}^c u_{ik}^p - \sum_{i=1}^c u_{ik}^p v_i \right]_{\beta_k=\beta_k^*} = 0. \quad (9.45)$$

Thus, the zero-gradient condition for the bias field estimator is expressed as

$$\beta_k^* = y_k - \frac{\sum_{i=1}^c u_{ik}^p v_i}{\sum_{i=1}^c u_{ik}^p}. \quad (9.46)$$

9.4.4.7 BCFCM Algorithm

The BCFCM algorithm for correcting the bias field and segmenting the image into different clusters can be summarized in the following steps:

Step 1. Select initial class prototypes $\{v_i\}_{i=1}^c$. Set $\{\beta_k\}_{k=1}^N$ to equal and very small values (e.g. 0.01).

Step 2. Update the partition matrix using Eq. 9.40.

Step 3. The prototypes of the clusters are obtained in the form of weighted averages of the patterns using Eq. 9.42.

Step 4. Estimate the bias term using Eq. 9.46.

Repeat steps 2–4 till termination. The termination criterion is as follows

$$||\mathbf{V}_{new} - \mathbf{V}_{old}|| < \epsilon, \quad (9.47)$$

where $|| \cdot ||$ is the Euclidean norm, $\mathbf{V}$ is a vector of cluster centers, and ϵ is a small number that can be set by the user.

9.4.4.8 BCFCM Results

In this section, we describe the application of the BCFCM segmentation to synthetic images corrupted with multiplicative gain, as well as digital MR phantoms [51] and real brain MR images. The MR phantoms simulated the appearance and image characteristics of the T1 weighted images. There are many advantages of using digital phantoms rather than real image data for validating segmentation methods. These advantages include prior knowledge of the true tissue types and control over image parameters such as mean intensity values, noise, and intensity inhomogeneities. We used a high-resolution T1 weighted phantom with in-plane resolution of 0.94 mm^2 , Gaussian noise with $\sigma = 6.0$, and 3D linear shading of 7% in each direction. All of the real MR images shown in this section were obtained using a General Electric Signa 1.5 T clinical MR imager with the same in-plane resolution as the phantom. In all the examples, we set the parameter α (the neighbors effect) to be 0.7, $p = 2$, $N_R = 9$ (a 3×3 window centered around each pixel), and $\epsilon = 0.01$. For low SNR images, we set $\alpha = 0.85$. The choice of these parameters seems to give the best results.

Figure 9.15(a) shows a synthetic test image. This image contains a two-class pattern corrupted by a sinusoidal gain field of higher spatial frequency. The test image is intended to represent two tissue classes, while the sinusoid represents an intensity inhomogeneity. This image was constructed so that it would be difficult to correct using homomorphic filtering or traditional FCM approaches. As shown in Fig. 9.15(b), FCM algorithm was unable to separate the two classes, while the BCFCM and EM algorithms have succeeded in correcting and classifying the data as shown in Fig. 9.15(c). The estimate of the multiplicative gain

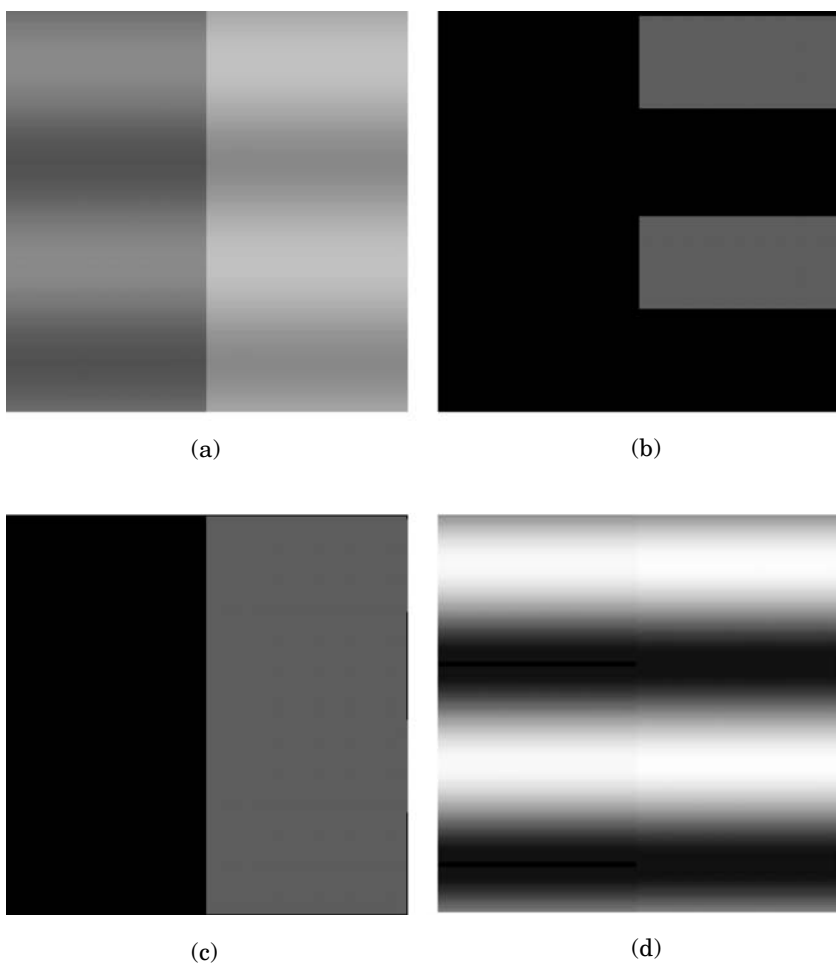


Figure 9.15: Comparison of segmentation results on a synthetic image corrupted by a sinusoidal bias field. (a) The original image, (b) FCM results, (c) BCFCM and EM results, and (d) bias field estimations using BCFCM and EM algorithms: this was obtained by scaling the bias field values from 1 to 255.

using either BCFCM or EM is presented in Fig. 9.15(d). This image was obtained by scaling the values of the bias field from 1 to 255. Although the BCFCM and EM algorithms produced similar results, BCFCM was faster to converge to the correct classification, as shown in Fig. 9.16.

Figures 9.17 and 9.18 present a comparison of segmentation results between FCM, EM, and BCFCM, when applied on T1 weighted MR phantom corrupted with intensity inhomogeneity and noise. From these images, we can see that

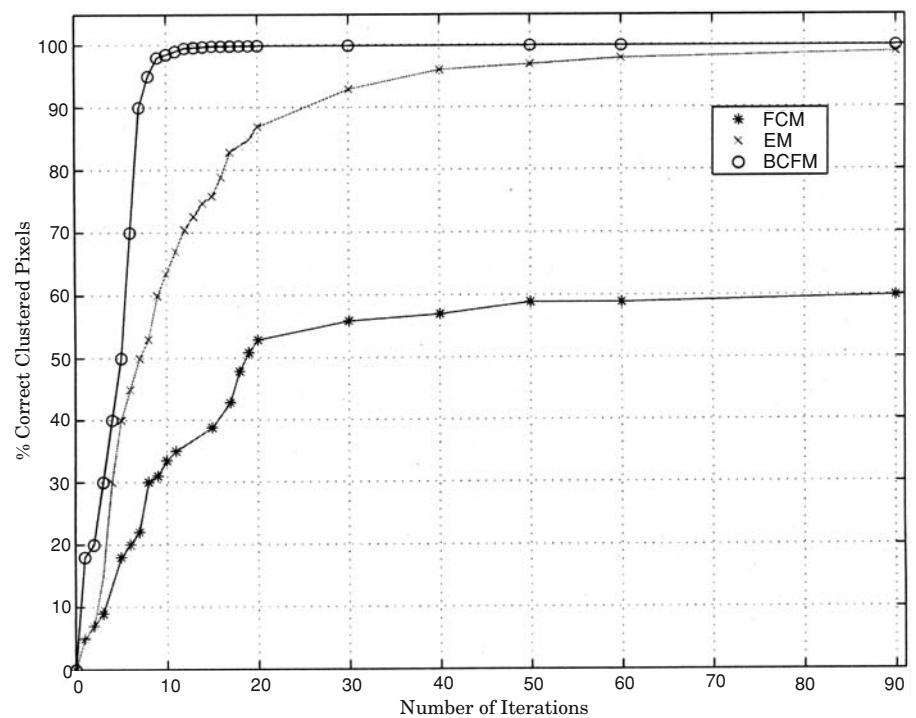


Figure 9.16: Comparison of the performance of the proposed BCFCM algorithm with EM and FCM segmentation when applied to the synthetic two-class image shown in Fig. 9.15(a).

traditional FCM was unable to correctly classify the images. Both BCFCM and EM segmented the image into three classes corresponding to background, gray matter (GM), and white matter (WM). BCFCM produced slightly better results than EM due to its ability to cope with noise. Moreover, BCFCM requires far less number of iterations to converge compared to the EM algorithm. Table 9.2 depicts the segmentation accuracy (SA) of the three mentioned method when applied to the MR phantom. SA was measured as follows:

$$SA = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}} \times 100\% \tag{9.48}$$

SA was calculated for different SNR. From the results, we can see that the three methods produced almost similar results for high SNR. BCFCM method, however, was found to be more accurate for lower SNR.

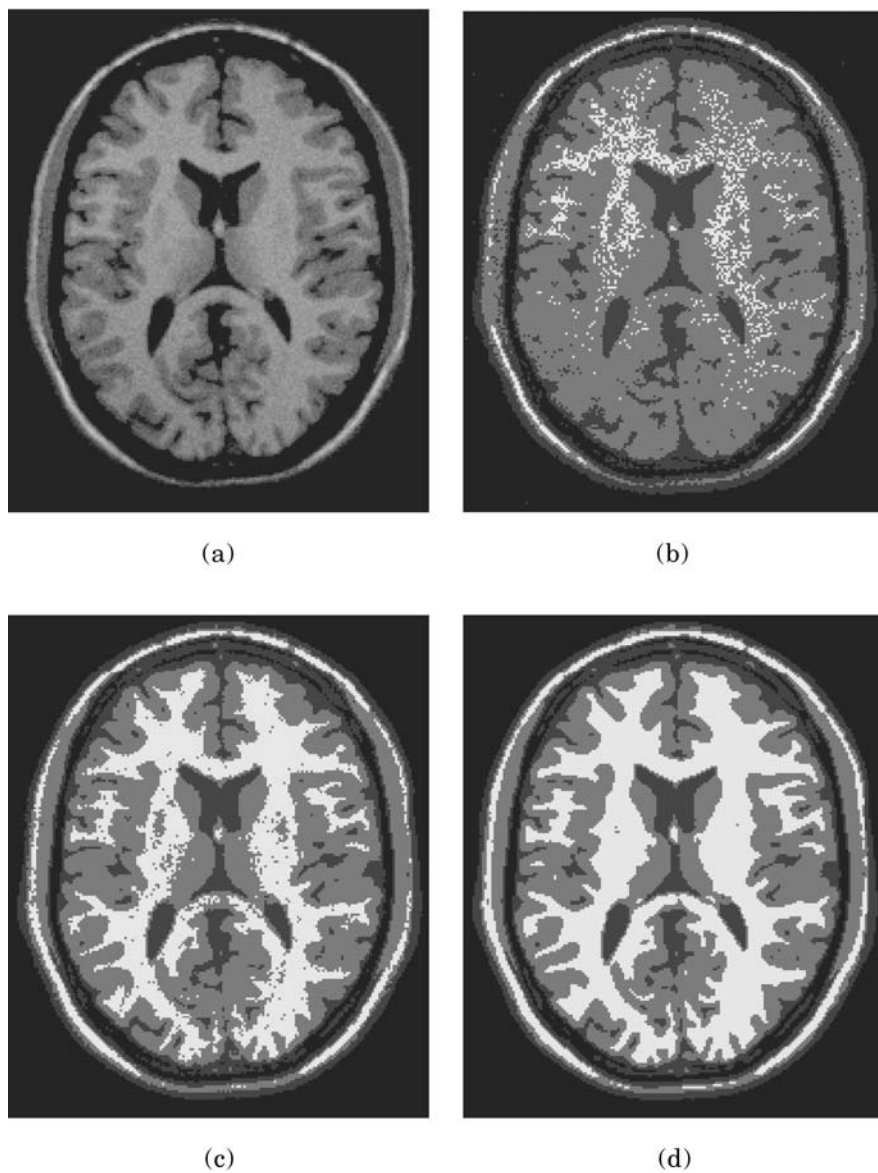


Figure 9.17: Comparison of segmentation results on a MR phantom corrupted with 5% Gaussian noise and 20% intensity inhomogeneity: (a) original T1 weighted image, (b) using FCM, (c) using EM, and (d) using the proposed BCFCM.

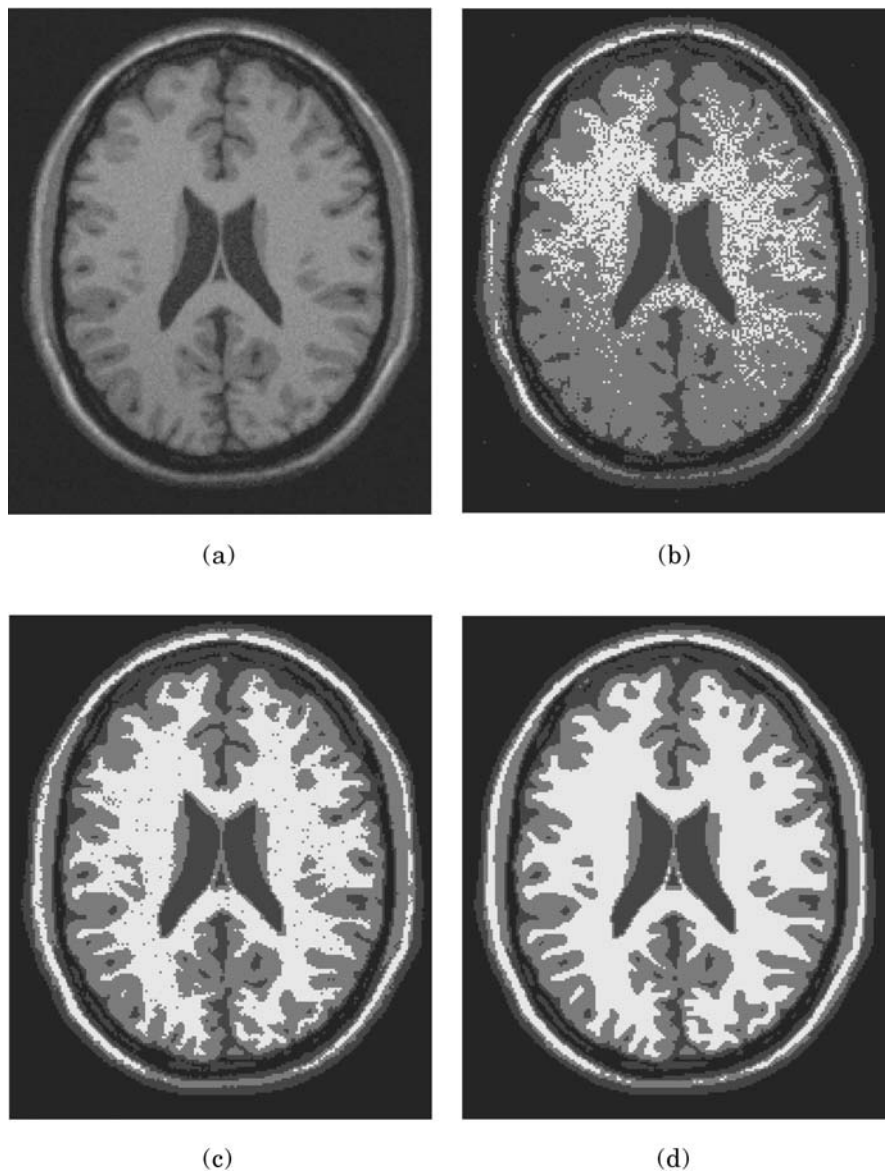


Figure 9.18: Comparison of segmentation results on an MR phantom corrupted with 5% Gaussian noise and 20% intensity inhomogeneity: (a) original T1 weighted image, (b) using FCM, (c) using EM, and (d) using the proposed BCFCM.

Table 9.2: Segmentation accuracy of different methods when applied on MR simulated data

Segmentation Method	SNR		
	13 db	10 db	8 db
FCM	98.92	86.24	78.9
EM	99.12	93.53	85.11
BCFCM	99.25	97.3	93.7

Figure 9.19 shows the results of applying the BCFCM algorithm to segment a real axial-sectioned T1 MR brain. Strong inhomogeneities are apparent in the image. The BCFCM algorithm segmented the image into three classes corresponding to background, GM, and WM. The bottom right image shows the estimate of the multiplicative gain, scaled from 1 to 255.

Figure 9.20 shows the results of applying the BCFCM for the segmentation of noisy brain images. The results using traditional FCM without considering the neighborhood field effect and the BCFCM are presented. Notice that the BCFCM segmentation, which uses the the neighborhood field effect, is much less fragmented than the traditional FCM approach. As mentioned before, the relative importance of the regularizing term is inversely proportional to the SNR of MRI signal. It is important to note, however, that the incorporation of spatial constraints into the classification has the disadvantage of blurring some fine details. There are current efforts to solve this problem by including contrast information into the classification. High contrast pixels, which usually represent boundaries between objects, should not be included in the neighbors.

9.5 Level Sets

The mathematical foundation of deformable models represents the confluence of physics and geometry. Geometry serves to represent object shape and physics puts some constraints on how it may vary over space and time. Deformable models have had great success in imaging and computer graphics. Deformable models include snakes and active contours. Snakes are used based on the geometric properties in image data to extract objects and anatomical structures in medical imaging. After initialization, snakes evolve to get the object. The change of

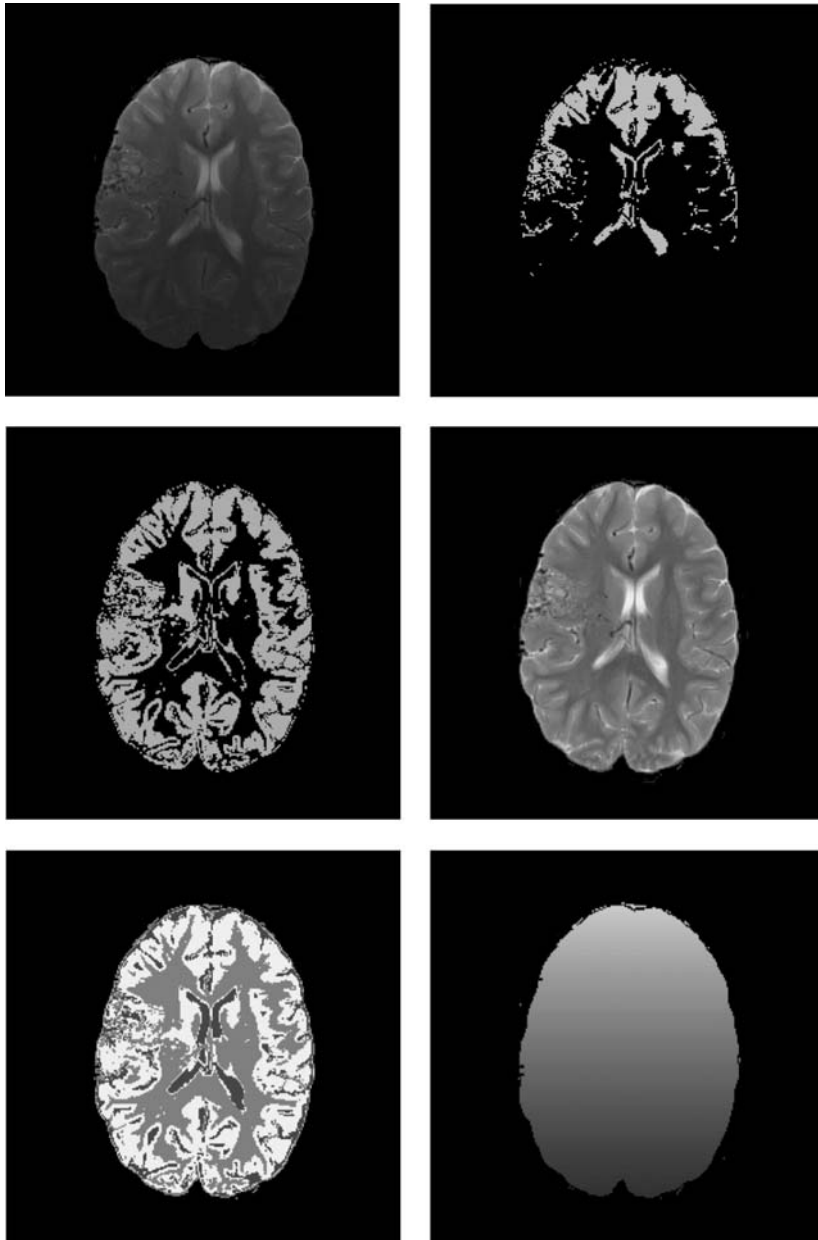


Figure 9.19: Brain MRI example: (upper left) the original MR image corrupted with intensity inhomogeneities. (Upper right) crisp gray matter membership using traditional FCM. (Middle left) crisp gray matter membership using the proposed BCFCM algorithm. (Middle right) the bias-field corrected image using BCFCM. The segmented image and bias field estimate using BCFCM are shown in bottom left and bottom right, respectively.

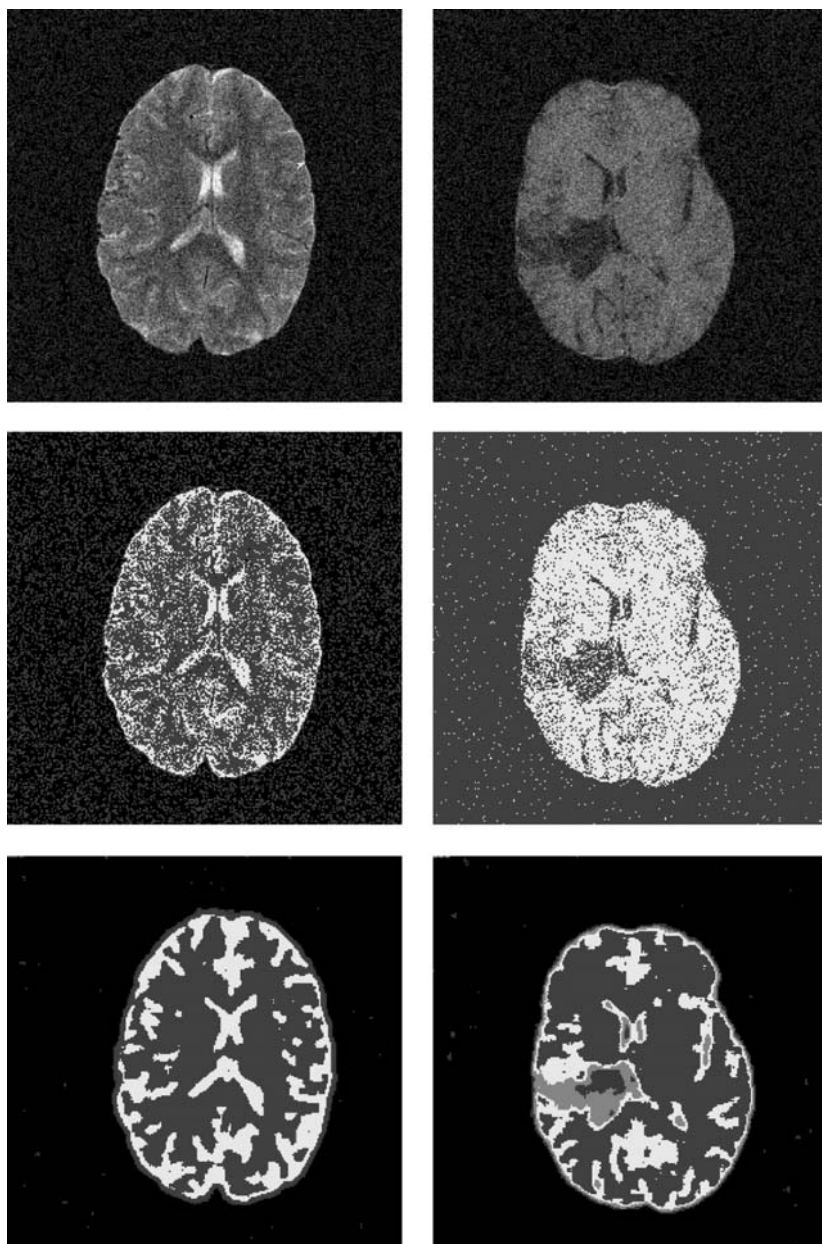


Figure 9.20: Brain tumor MRI examples. Upper row: Original MR images corrupted with salt and pepper noise. Middle row: the segmented images using FCM without any neighborhood consideration. Bottom row: The segmented images using BCFCM ($\alpha = 0.85$).

snakes with time is guided by differential equations. These equations are derived from the energy minimization concept to describe the change of snakes with time. The output obtained using snakes depends highly on the initialization. It was found that initial curve has to lie close to the final solution to obtain required results. The initialization is relatively easy in the case of 2D images but in the 3D case it is very difficult. Also the topology change of the solution needs a special regulation to the model.

Level sets were invented to handle the problem of changing topology of curves. The level sets has had great success in computer graphics and vision. Also, it was used widely in medical imaging for segmentation and shape recovery. It proved to have advantages over statistical approaches followed by mathematical morphology. In the following section we will give a brief overview on level sets and its application in image segmentation.

9.5.1 Level Set Function Representation

Level sets was invented by Osher and Sethian [52] to handle the topology changes of curves. A simple representation is that a surface intersects with the zero plane to give the curve. When this surfaces changes the curve changes. The surface can be described by the following equation:

$$\phi(x, t) > 0 \text{ if } x \in \Omega, \quad \phi(x, t) < 0 \text{ if } x \notin \Omega, \quad \text{and } \phi(x, t) = 0 \text{ if } x \in \Gamma, \quad (9.49)$$

where ϕ represents the surface function, Ω denotes the set of points where the function is positive, and Γ represents the set of points at which the function is zero. In Fig. 9.21, an example of a surface and its intersection with the zero plane is shown. This intersection is called the front. The surface changes with time, resulting in different fronts. So the level set function is positive at some points, negative at other points, and zero at the front Γ . The time as extra dimension is added to the problem to track the changes of the front. The topology changes of the curve are handled naturally by this presentation as we see from Fig. 9.22. The first row represents the surface and the zero plane at different time samples and the second row represents the resulting curves. The front is initially two ellipses, then the two ellipses merge to make a closed curve and it changes and so on. This representation allows the front to merge and break.

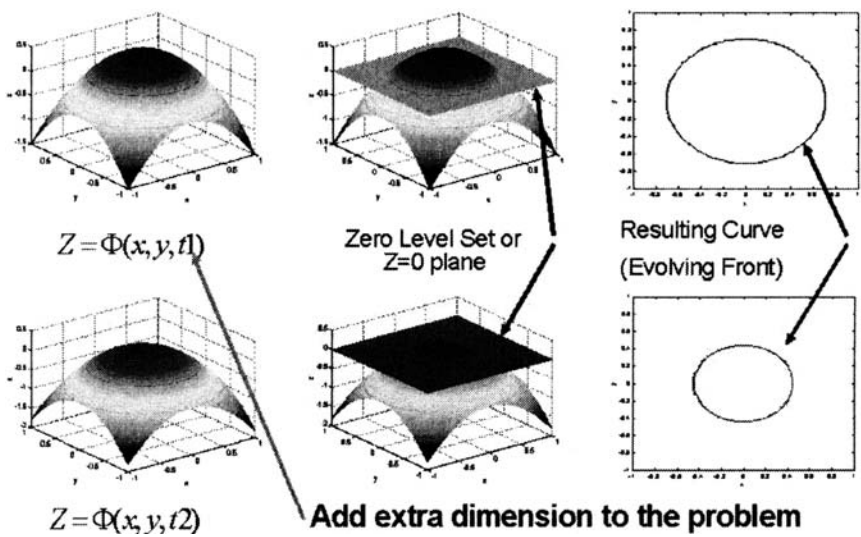


Figure 9.21: Change of the level set function with time resulting in different curves.

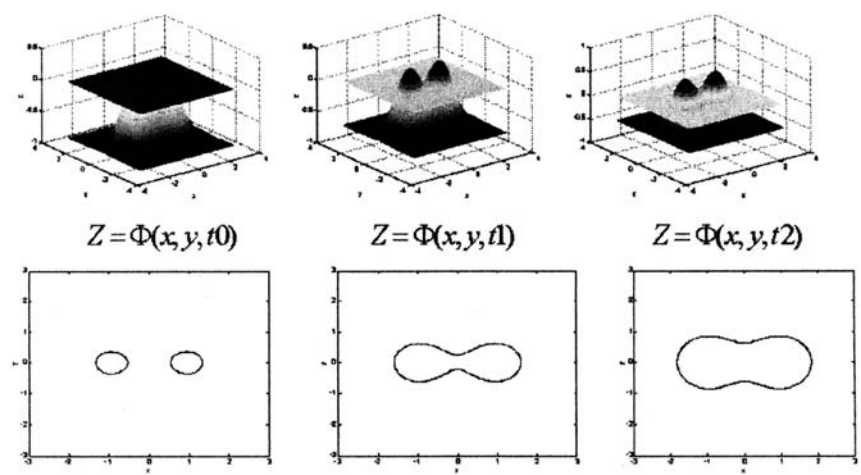


Figure 9.22: Topology change of curves with time.

9.5.2 Curve Evolution with Level Sets

To get an equation describing the change of the curve or the front with time, we will start with the assumption that the level set function is zero at the front as follows:

$$\phi(x, y, t) = 0 \quad \text{if } (x, y) \in \Gamma, \tag{9.50}$$

and then compute its derivative which is also zero,

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \phi}{\partial y} \frac{\partial y}{\partial t} = 0, \quad (9.51)$$

Converting the terms to the dot product form of the gradient vector and the x and y derivatives vector, we get

$$\frac{\partial \phi}{\partial t} + \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y} \right) \cdot \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t} \right) = 0. \quad (9.52)$$

Multiplying and dividing by $|\nabla \phi|$ and taking the other part to be F , we get the following equation:

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0, \quad (9.53)$$

Where F , the speed function, is given by

$$F = \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y} \right) \cdot \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t} \right) / |\nabla \phi|. \quad (9.54)$$

The selection of the speed function is very important to keep the change of the front smooth and also it is application dependent. Equation 9.55 represents speed function containing the mean curvature k . The positive sign means that the front is shrinking and the negative sign means that the front is expanding and ϵ is selected to be a small value for smoothness. The curvature term allows the front to merge and break and also handles sharp corners,

$$F = \pm 1 - \epsilon k, \quad (9.55)$$

Where k is given by

$$k = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}. \quad (9.56)$$

In 3D, the front will be an evolving surface rather than an evolving curve.

9.5.3 Stability and CFL Restriction

The numerical solution of the partial differential equation (PDE) describing the front is very important to be accurate and stable. For simplicity, Taylor's series expansion is used to handle the partial derivatives of ϕ as listed below,

$$\phi(x, y, t + \Delta t) = \phi(x, y, t) - \Delta t F |\nabla \phi|, \quad (9.57)$$

$$\phi_x(x, y, t) = (\phi(x + \Delta x, y, t) - \phi(x, y, t)) / \Delta x, \quad (9.58)$$

$$\phi_y(x, y, t) = (\phi(x, y + \Delta y, t) - \phi(x, y, t))/\Delta y, \quad (9.59)$$

$$\phi_{xx}(x, y, t) = (\phi(x + 2\Delta x, y, t) - 2\phi(x, y, t) + \phi(x - 2\Delta x, y, t))/(2\Delta x^2), \quad (9.60)$$

$$\phi_{yy}(x, y, t) = (\phi(x, y + 2\Delta y, t) - 2\phi(x, y, t) + \phi(x, y - 2\Delta y, t))/(2\Delta y^2). \quad (9.61)$$

There are different numerical techniques used for this problem and the details are given in [52]. The solution is very sensitive to the time step. Time step is selected based on the Courant–Friedrichs–Levy (CFL) restriction. It requires the front to cross no more than one grid cell at each time step Δt . This calculation will give the maximum time step that guarantees stability. From Eq. 9.62, we maximize the denominator and minimize the nominator to get the best value of the time step. The time step is calculated at each iteration of the process to maintain the stability of the solution:

$$\Delta t \leq \frac{(\phi_x^2 + \phi_y^2)^{1/2}}{F(|\phi_x|/\Delta x + |\phi_y|/\Delta y)} \quad (9.62)$$

9.5.4 Tracking the Front

Now, the solution is to find the front iteratively at different time steps. We get the front by intersecting the surface with the zero plane. We need to track this front by getting the length of the front or getting the area enclosed. This information is very important in the segmentation problem as we will see in the next sections. Simply the enclosed area contains all the points at which the level set function is greater than or equal to zero and the points of the front are the points at which the level set function is zero. Applying the heaviside step and delta functions is very useful in getting the area and the front respectively. For numerical implementation, it is desirable to replace the heaviside and the delta functions by some counterparts. Approximations of these two functions are used to handle smoothness problem as follows:

$$H_\alpha(\phi) = \begin{cases} 1, & \text{if } |\phi| > \alpha \\ 0.5(1 + \frac{\phi}{\alpha} + \frac{1}{\pi} \sin(\frac{\pi\phi}{\alpha})) & \text{if } |\phi| \leq \alpha \end{cases}, \quad (9.63)$$

$$\delta_\alpha(\phi) = \begin{cases} 0, & \text{if } |\phi| > \alpha \\ \frac{1}{2\alpha}(1 + \cos(\frac{\pi\phi}{\alpha})), & \text{if } |\phi| \leq \alpha \end{cases}. \quad (9.64)$$

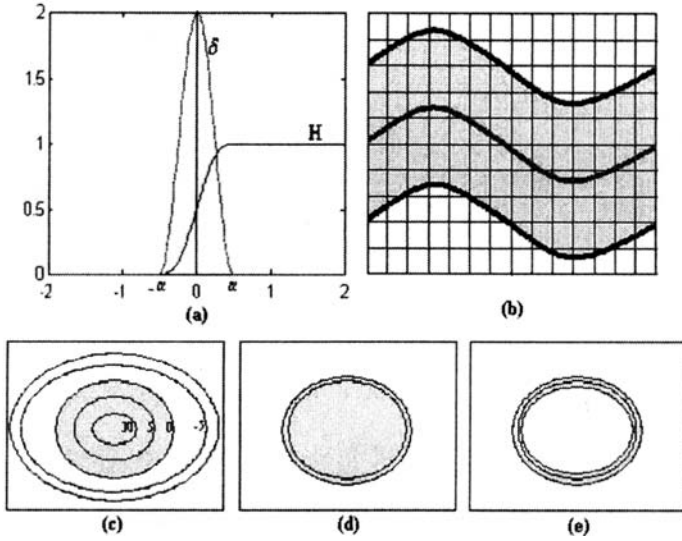


Figure 9.23: (a) The plot of the heaviside and delta functions for a specific value of α , (b) the narrow band points, (c) the level set function, (d) applying the heaviside step function, and (e) applying the delta function.

In Fig. 9.23(a), the two functions are plotted for $\alpha = 0.5$. The value of α is always taken to be $1.5\Delta x$ to make the band equal to $3\Delta x$ where Δx is the mesh size, which is always 1. The enclosed area (A) and the length of the interface or front (L) are calculated as follows:

$$A = \int \int_D H_\alpha(\phi) dx dy, \quad (9.65)$$

$$L = \int \int_D \delta_\alpha(\phi) |\nabla \phi| dx dy, \quad (9.66)$$

where D is the domain. A proof of Eq. 9.66 to be the length of the front is found in [53].

In Fig. 9.23(b), the red line represents the front and the yellow area represents the points around the front where this area is called the narrow band. In (c), (d), and (e) an example of a level set function and application of the heaviside step and delta functions are shown.

9.5.5 Narrow Banding and Speed of the Solution

Solving the PDE of the level set function requires numerical processing at each point of the image domain which is a time consuming process. Only we are

interested in the change of the front. It is not important to get the solution at points far away from the front, so the solution is important at the points near the front. The points (highlighted in Fig. 9.23(b)) are called the narrow band points. The change of the level set function at these points only is considered. Other points (outside the narrow band) are called the far away points and they are given large positive or large negative values to be out of interest (not processed), and it speeds up the iterations. The use of the delta function defined by Eq. 9.64 is very important to give the narrow band points.

9.5.6 Reinitialization

The existence of the front means that the level set function has positive and negative parts, then it has negative and positive values including zeroes. The level set function with this property is called a signed distance function. This property should be kept through the iterations in order not to lose the front. There are different solutions for this problem [54]. We will discuss only the solution introduced by Osher *et al.* [55]. It was proved that recomputing the level set function by solving Eq. 9.67 frequently enough will maintain the function as signed distance function:

$$\frac{\partial \phi}{\partial t} = \text{sign}(\phi)(1 - |\nabla \phi|), \quad (9.67)$$

where it contains the sign function sign . When the level set function is negative, the information flows one way and when it is positive, the information flows the other way. The net effect is to “straighten out” the level set function on either sides of the zero level set,

$$0 = \text{sign}(\phi)(1 - |\nabla \phi|). \quad (9.68)$$

By solving this equation, the derivative of ϕ with respect to time will vanish resulting in Eq. 9.68. $|\nabla \phi| = 1$ denotes the measure for signed distance function.

9.6 Application: MRA Data Segmentation Using Level Sets

The human cerebrovascular system is a complex three-dimensional anatomical structure. Serious types of vascular diseases such as carotid stenosis, aneurysm,

and vascular malformation may lead to brain stroke, which is the third leading cause of death and the main cause of disability. An accurate model of the vascular system from MRA data volume is needed to detect these diseases at early stages and hence may prevent invasive treatments. A variety of methods have been developed for segmenting vessels within MRA. One class of methods is based on a statistical model, which classifies voxels within the image volume into either vascular or nonvascular class for time-of-flight MRA [56]. Another class of segmentation is based on intensity threshold where points are classified as either greater or less than a given intensity. This is the basis of the isointensity surface reconstruction method [57–59]. This method suffers from errors due to image inhomogeneities in addition; the choice of the threshold level is subjective. An alternative to segmentation is axis detection known as skeletonization process, where the central line of the tree vessels is extracted based on the tubular shape of vessels [60]. Other approaches for MRA vessel segmentation are the manually defined seed locations for segmentation [61].

In this section, we use level set method for image segmentation to improve the accuracy of the vascular segmentation. This work is a supervised classification which means that the number of classes and the class distribution are assumed to be known. Usually, the class distribution is assumed to be Gaussian with known mean and variance. In [53], classes were assumed to be phases separated by interface boundaries where each class has its corresponding level set function. A set of functionals were developed with properties of regularity. The level set function representation depends on these functionals. Each class occupies certain areas (regions) in the image. The level set function is represented based on the regions i.e. it is positive inside the region, negative outside, and zero on the boundary. The classes have no common areas i.e., the intersection between classes is not allowed. The sum of lengths of the interfaces between the areas is taken in consideration. The functionals are dependent mainly on these properties and they are expected to have a local minimum which is the segmented image. The change of each level set is guided by two forces, the minimal length of interfaces which is the internal force and the homogeneous class distribution which is the external one.

A PDE guides the motion of each level set. This work saves the manual initialization of level set functions [62]. Bad initialization for these functions makes the segmentation fail. Automatic seed initialization is made for each slice of the volume by dividing the image into windows, and based on the gray

level a corresponding signed distance level set function is initialized for each window. After segmenting the volume, a connectivity filter [63] is used to exploit the fact that the vascular system is a tree-like structure and makes use of the 3D computer graphics region-filling algorithm to extract the vascular tree. The used algorithm with MRA data volumes is evaluated using a phantom, showing a good accuracy. The algorithm is applied to different types of MRA data sets, showing good results. This approach can be extended to be not dependent only on the gray level, but also on the geometrical features of the segmented areas, leading to more accuracy.

9.6.1 Level Sets and Segmentation

Consider an image of c classes. We assign a level set function ϕ_i for each class. From the definition of the level set function in Eq. 9.49, Ω is the class and Γ is the interface of the class. Class interface denotes the boundary line between the class and the other classes. From the following equations, F_1 is the partitioning condition as follows,

$$F_1 = \frac{\lambda_i}{2} \int_{\Omega} \sum_{i=1}^c (H_{\alpha}(\phi_i) - 1)^2 dx, \text{ where } \lambda_i \in R^+, \forall i \in [1, c]. \quad (9.69)$$

The partitioning condition penalizes the vacuum points and prevents the overlapping between regions:

$$F_2 = \sum_{i=1}^c e_i \int_{\Omega} H_{\alpha}(\phi_i) \frac{(u_0 - u_i)^2}{\sigma_i^2} dx, \text{ where } e_i \in R, \forall i \in [1, c]. \quad (9.70)$$

F_2 is the data term condition with mean u_i and variance σ_i^2 where u_0 is the data value.

$$F_3 = \sum_{i=1}^c \gamma_i \int_{\Omega} \delta_{\alpha}(\phi_i) |\nabla \phi_i| dx, \text{ where } \gamma_i \in R, \forall i \in [1, c]. \quad (9.71)$$

F_3 is the sum of interfaces length between classes. The summation $F_1 + F_2 + F_3$ is minimized with respect to ϕ to get the following equation:

$$\phi_i^{t+1} = \phi_i^t - \Delta t \delta_{\alpha}(\phi_i^t) \left[e_i \frac{(u_0 - u_i)^2}{\sigma_i^2} - \gamma_i \operatorname{div} \left(\frac{\nabla \phi_i^t}{|\phi_i^t|} \right) + \lambda_i \left(\sum_{i=1}^K H_{\alpha}(\phi_i^t) - 1 \right) \right]. \quad (9.72)$$

This solution represents the level set function variation with time. When the function approaches the steady state, it does not change. It has positive, negative, and zero parts. We are interested only in the positive parts. Each pixel in the positive parts belongs to the associated class of its function. By this representation, the level set function formulation allows breaking and merging fronts since Eq. 9.72 contains the curvature term which is considered to be a smoothing part.

9.6.2 Volume Segmentation Algorithm

Step 0: Initialize ϕ_i , $\forall i \in [1, c]$.

Step 1: $t = t + 1$.

Step 2: Update each function using Eq. 9.72.

Step 3: Solve Eq. 9.67 for each of n iterations to keep the signed distance function property.

Step 4: Smooth each function and remove noise.

Step 5: If steady state is not reached, then go to Step 1, else go to next slice.

Step 0 is very important since bad initialization leads to bad segmentation. Automatic seed initialization is used to speed up the process and it is also less sensitive to noise. Automatic seed initialization is to divide the image into nonoverlapped windows of predefined size. Then the average gray level is calculated and compared to the mean of each class to specify the nearest class it belongs to. A signed distance function is initialized to each window. The connectivity filter is applied to remove the nonvessel tissues. The filter exploits the fact that the vascular system is a tree-like structure.

9.6.3 Segmentation Quality Measurement

A 2D phantom is designed to simulate the MRA. This phantom image contains many circles with decreasing diameters such as the cerebrovascular tree shape which is a cone-shaped. Then using the level set segmentation algorithm with this image, we obtain a resultant image containing the vessels. The SA is measured by Eq. 9.48.

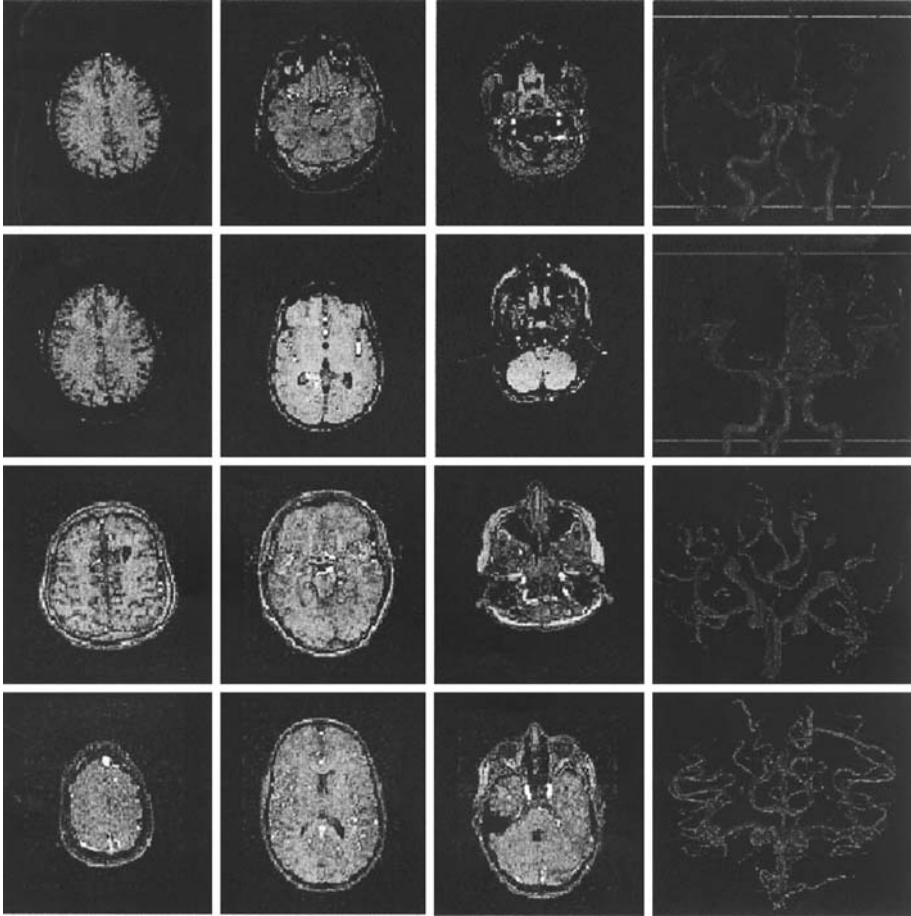


Figure 9.24: Segmentation and Visualization of different data sets.

9.6.4 Results and Discussion

The technique has been applied to different data sets of MR angiography phase contrast and time-of-flight types. For each type two volumes are used to prove the accuracy of the technique. The first type of data is $117 \times 256 \times 256$ (the first two rows of Fig. 9.24) and the second type is $93 \times 512 \times 512$ (the second two rows of Fig. 9.24). First, level sets are initialized by automatic seed initialization. Automatic seed initialization is used in each slice and each slice is divided into windows of size 5×5 . An average mean is estimated for each class from the average histogram of the volume, and signed distance functions are assigned where each level set function is a collection of Gaussian surfaces added together with a

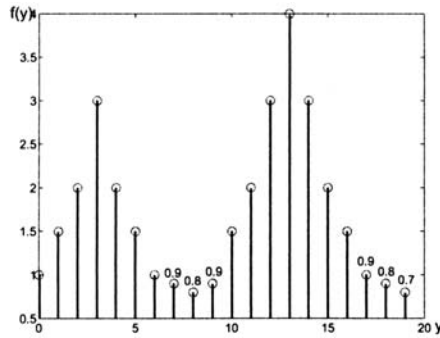


Figure 9.25: Histogram of empirical data.

time step of 0.1 sec. Using this initialization decreases the number of iterations, leading to fast extraction of the vascular tree. The volume segmentation takes about 20 min. on the unix workstation with the super computer. Segmentation results are exposed to the connectivity filter to remove the nonvessel areas. Each volume is visualized to show the vascular tree. The segmentation accuracy was measured to be 94% which is very good for this type of data. The 2D phantom can be modified to be a 3D one simulating the whole volume leading to more accuracy. The results are promising with a good accuracy. This model can be extended to unsupervised case including a parameter estimation capability in future work. Future work will include geometrical features to the segmentation model to enhance the segmentation results.

Questions

1. What are the main three properties of MRF?
2. Using traditional EM algorithm, estimate the mean, the variance, and the proportional for the two classes shown in Fig. 9.25? (Hint: Before applying EM algorithm, normalize $f(y)$ such that $\sum_{\text{for all } y} f(y) = 1$, and assume each class comes from normal distribution).
3. What are the main advantages of using the genetic algorithm as optimization tool?
4. When it is useful to use GMRF in image segmentation, and when is it not useful?

5. *What is the advantages of using GMRF in image segmentation?*
6. *Derive the CFL restriction to find the optimal time step in 3D case.*
7. *Suggest an algorithm to mark the narrow band points in both 2D and 3D. Compare it with the use of the Dirac delta function.*
8. *Level sets are used to extract anatomical structures from 2D and 3D data. What are the advantages of using level sets in 3D?*
9. *Using the front as the zero level embedded in the surface has many advantages over using scattered points representing the front. What are these advantages?*
10. *If we have the front as a surface embedded in a 4D function, can we slice the front as curves in 2D to make the implementation easier? Why?*

Bibliography

- [1] Jain, A. K., Advances in mathematical models for image processing, *Proc. IEEE*, Vol. 69, No. 5, pp. 502–528, 1981.
- [2] Chellapa, R. and Kashyap, R. L., Digital image restoration using spatial interaction models, *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. 30, No. 3, pp. 461–471, 1982.
- [3] Besag, J. E., On the statistical analysis of dirty pictures, *J. R. stat. Soc.*, London, B, Vol. B-48, pp. 259–302, 1986.
- [4] Bouman, C. A. and Shapiro, M., A multiscale random field model for Bayesian image segmentation, *IEEE Trans. Image Process.*, Vol. 3, No. 2, pp. 162–177, 1994.
- [5] Benveniste, A., Nikoukhah, R., and Willsky, A. S., Multiscale system theory, In: *Proc. 29th Conf. Decision and Control*, Dec. 1990, pp. 2484–2489, 1990.
- [6] Basseville, M., Benveniste, A., Chou, K. C., Golden, S. A., Nikoukhah, R., and Willsky, A. S., Modeling and estimation of multiresolution stochastic processes, *IEEE Trans. Inform. Theory*, Vol. 38, No. 2, pp. 766–784, 1992.
- [7] Amit, Y., Grenander, U., and Piccioni, M., Structure image restoration through deformable templates, *J. Am. Stat. Assoc.*, Vol. 86, No. 414, pp. 376–387, 1991.
- [8] Kam, A. C. and Kopec, G. E., Heuristic image decoding using separable source models, In *Proc. Int. Conf. Acoustic., Speech, Signal Processing*, Vol. 5, Adelaide, Australia, Apr. 19–22, pp. 145–148, 1994.
- [9] Kopec, G. E. and Chou, P. A., Document image decoding using Markov source models, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 16, No. 6, pp. 602–617, 1994.
- [10] Jain, A. K. and Dubes, R. C., Random field models in image analysis, *J. Appl. stat.*, Vol. 16, No. 2, 1989.

- [11] Geman, S. and Geman, D., Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images, *IEEE Trans. on Pattern Anal. Mach. Intell.*, Vol. PAMI, No. 6, pp. 721–741, 1984.
- [12] Geman, S. and Geman, D., Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 6, No. 6, pp. 721–741, 1984.
- [13] Besag, J. E., Spatial interaction and the statistical analysis of lattice system, *J. R. Stat. Soc., London, B*, Vol. 36, pp. 192–236, 1974.
- [14] Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
- [15] Derin, H. and Elliott, H., Modeling and segmentation of noisy and texture images using Gibbs random fields, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 9, pp. 39–55, 1987.
- [16] Farag, A. A. and Delp, E. J., Image segmentation based on composite random field models, *J. Opt. Eng.*, Vol. 12, pp. 2594–2607, 1992.
- [17] Bouman, C. A. and Liu, B., Multiple resolution segmentation of textured images, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 13, pp. 99–113, 1991.
- [18] El-Baz, A. and Farag, A. A., Image segmentation using GMRF models: Parameters estimation and applications, *IEEE International Conference on Image Processing*, Barcelona, Spain, pp. 177–180, 2003.
- [19] El-Baz, A. and Farag, A. A., Parameter estimation in Gibbs Markov image models, *6th international Conf. Information Fusion*, Queensland, Australia, pp. 934–942, 2003.
- [20] El-Baz, A. and Farag, A. A., Stochastic models in image analysis: Parameter estimations and case studies in image segmentation, *IEEE Workshop on Statistical Signal Processing*, St. Louis, Mo, USA pp. 154–157, 2003.
- [21] Boisselle, P. M. and White, C. S., *New Technique in Thoracic Imaging*, Marcel Daker Inc., New York, 2002.

- [22] Farag, A. A., El-Baz, A., and Gimel'farb G., Detection and recognition of lung abnormalities using deformable templates, *Proc. IAPR Int. Conf. Pattern Recognition (ICPR 2004)*, Cambridge, UK, 2004.
- [23] Hu, S. and Hoffman, E. A., Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images, *IEEE Trans. Med. Imaging*, Vol. 20, No. 6. pp. 490–498, 2001.
- [24] Condon, B. R., Patterson, J., and Wyper, D., Image nonuniformity in magnetic resonance imaging: Its magnitude and methods for its correction, *Br. J. Radiol.*, Vol. 60, pp. 83–87, 1987.
- [25] McVeigh, E. R., Bronskil, M. J., and Henkelman, R. M., Phase and sensitivity of receiver coils in magnetic resonance imaging, *Med. Phys.*, Vol. 13, pp. 806–814, 1986.
- [26] Wicks, D. A. G., Barker, G. J., and Tofts, P. S., Correction of intensity nonuniformity in MR images of any orientation, *Magn. Reson. Imaging*, Vol. 11, pp. 183–196, 1993.
- [27] Simmons, A., Tofts, P. S., Barker, G. J., and Arridge, S. R., Sources of intensity nonuniformity in spin echo images at 1.5 T, *Magn. Reson. Med.*, Vol. 32, pp. 121–128, 1994.
- [28] Wells, W. M., III, Grimson, W. E. L., Kikinis, R., and Jolesz, F. A., Adaptive segmentation of MRI data, *IEEE Trans. Med. Imaging*, Vol. 15, pp. 429–442, 1996.
- [29] Johnston, B., Atkins, M. S., Mackiewicz, B., and Anderson, M., Segmentation of multiple sclerosis lesions in intensity corrected multispectral MRI, *IEEE Trans. Med. Imaging*, Vol. 15, No. 2, pp. 154–169, 1996.
- [30] Sled, J. G., Zijdenbos, A. P., and Evans, A. C., A nonparametric method for automatic correction of intensity nonuniformity in MRI data, *IEEE Trans. Med. Imaging*, Vol. 17, No. 1, pp. 87–97, 1998.
- [31] Lai, S. and Fang, M., A new variational shape-from-orientation approach to correcting intensity inhomogeneities in MR images, *Workshop on Biomedical Image Analysis, CVPR98*, Santa Barbara, CA, pp. 56–63, 1998.

- [32] Meyer, C. R., Bland, P. H., and Pipe, J., Retrospective correction of intensity inhomogeneities in MRI, *IEEE Trans. Med. Imaging*, Vol. 14, No. 1, pp. 36–41, 1995.
- [33] Tincher, M., Meyer, C. R., Gupta, R., and Williams, D. M., Polynomial modeling and reduction of RF body coil spatial inhomogeneity in MRI, *IEEE Trans. Med. Imaging*, Vol. 12, No. 2, pp. 361–365, 1993.
- [34] Brinkmann, B. H., Manduca, A., and Robb, R. A., Optimized homomorphic unsharp masking for MR grayscale inhomogeneity correction, *IEEE Trans. Med. Imaging*, Vol. 17, No. 2, pp. 161–171, 1998.
- [35] Dawant, B., Zijdenbos, A., and Margolin, R., Correction of intensity variations in MR images for computer aided tissue classification, *IEEE Trans. Med. Imaging*, Vol. 12, pp. 770–781, 1993.
- [36] Gilles, S., Brady, M., Declerck, J., Thirion, J. P., and Ayache, N., Bias field correction of breast MR images, *Proceedings of the Fourth International Conference on Visualization in Biomedical Computing*, Hamburg, Germany, pp. 153–158, 1996.
- [37] Brechbuhler, C., Gerig, G., and Szekely, G., Compensation of spatial inhomogeneity in MRI based on a parametric bias estimate, *Proceedings of the Fourth International Conference on Visualization in Biomedical Computing*, Hamburg, Germany, pp. 141–146, 1996.
- [38] Guillemaud, R. and Brady, M., Estimating the bias field of MR images, *IEEE Trans. Med. Imaging*, Vol. 16, No. 3, pp. 238–251, 1997.
- [39] Bezdek, J. C. and Pal, S. K., *Fuzzy models for pattern recognition*, IEEE Press, Piscataway, NJ, 1991.
- [40] Dunn, J. C., A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *J. Cybern.*, Vol. 3, pp. 32–57, 1973.
- [41] Bezdek, J., A convergence theorem for the fuzzy ISODATA clustering algorithms, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1980.
- [42] Xu, C., Pham, D., and Prince, J., Finding the brain cortex using fuzzy segmentation, isosurfaces, and deformable surfaces, *Proc. of the*

- fifteenth Int. Conf. on Information Processing in Medical Imaging (IPMI 97), pp. 399–404, 1997.
- [43] Pham, D. L. and Prince, J. L., Adaptive fuzzy segmentation of magnetic resonance image, *IEEE Trans. Med. Imaging*, Vol. 18, No. 9, pp. 737–752, 1999.
- [44] Ahmed, M. N., Yamany, S. M., Mohamed, N. A., Farag, A. A., and Moriarty, T., Bias field estimation and adaptive segmentation of MRI data using modified fuzzy C-means algorithm, *Proc. IEEE Int. Conf. Computer Vision and Pattern Recogn.*, Fort Collins, Co., Vol. 1, pp. 250–255, 1999.
- [45] Udupa, J. K. and Samarasekera, S., Fuzzy connectedness and object definition: Theory, algorithm and applications in image segmentation, *Graph. Models and Image Process.*, Vol. 58, No. 3, pp. 246–261, 1996.
- [46] Yamany, S. M., Farag, A. A., and Hsu, S., A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems, *Pattern Recognit. Lett.*, Vol. 20, pp. 1431–1438, 1999.
- [47] Bezdek, J., Hall, L., and Clarke, L., Review of MR image segmentation using pattern recognition, *Med. Phys.*, Vol. 20, pp. 1033–1048, 1993.
- [48] Brandt, M. E., Bohan, T. P., Kramer, L. A., and Fletcher, J. M., Estimation of CSF, white matter and gray matter volumes in hydrocephalic children using fuzzy clustering of MR images, *Comput. Med. Imaging Graph*, Vol. 18, pp. 25–34, 1994.
- [49] Hall, L. O., Bensaid, A. M., Clarke, L. P., Velthuizen, R. P., Silbiger, M. S., and Bezdek, J. C., A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain, *IEEE Trans. Neural Net.*, Vol. 3, pp. 672–682, 1992.
- [50] Mohamed, Nevin A., Modified Fuzzy C-Mean Algorithm for Medical Image Segmentation, M.Sc. Thesis, Elect. Eng. Dept., University of Louisville, 1999.

- [51] Goldszal, A. F., Davatzikos, C., Pham, D. L., Yan, M. X. H., Bryan, R. N., and Resnick, S. M., An image processing system for qualitative and quantitative volumetric analysis of brain images, *J. Comput. Assist. Tomogr*, Vol. 22, No. 5, pp. 827–837, 1998.
- [52] Sethian, J. A., *Level Set Methods and Fast Marching Methods*, Cambridge, 1999.
- [53] Zaho, H-K., Chan, T., Merriman, B., and Osher, S., A variational level set approach to multiphase motion, *J. Comput. Phys.*, Vol. 127, pp. 179–195, 1996.
- [54] Hongchuan Yu, Dejun Wang, Zesheng Tang, Level set methods and image segmentation, *IEEE International Workshop on Medical Imaging and Augmented Reality (MIAR'01)*, 2001.
- [55] Sussman, M., Smereka, P., and Osher, S., A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.*, Vol. 114, pp. 146–159, 1994.
- [56] Wilson, D. L. and Noble, J. A., An adaptive segmentation algorithm for time-of-flight MRA data, *IEEE Trans. Med. Imaging*, Vol. 18, No. 10, pp. 938–945, 1999.
- [57] Cline, H. E., Lorensen, W. E., Kikinis, R., and Jolesz, R., Three-dimensional segmentation of MR images of the head using probability and connectivity, *Neurosurgery*, Vol. 14, pp. 1037–1045, 1990.
- [58] Nakajima, S., Atsumi, H., and Bhalerao, A. H., Computer-assisted surgical planning for cerebrovascular neurosurgery, *Neurosurgery*, Vol. 41, pp. 403–409, 1997.
- [59] Cline, H. E., Lorensen, W. E., Souza, S. P., Jolesz, F. A., Kikinis, R., Gerig, G., and Kennedy, T. E., 3D surface rendered MR images of the brain and its vasculature, *J. Comput. Assist. Tomogr.*, Vol. 15, pp. 344–351, 1991.
- [60] Yim, Peter J., Choyke, Peter L., and Ronald M., Summers, Gray-scale skeletonization of small vessels in magnetic resonance angiography, *IEEE Trans. Med. Imaging*, Vol. 19, No. 6, pp. 568–576, 2000.

- [61] Bullitt, E., Symbolic description of intracerebral vessels segmented from magnetic resonance angiograms and evaluation by comparison with X-ray angiograms, *Med. Image Anal.*, Vol. 5, pp. 157–169, 2001.
- [62] Hossam, H. and Farag, Aly A., MRA data segmentation using level sets, *Proc. of the International Conference of Image Processing (ICIP)*, Barcelona, Spain, pp. 173–176, 2003.
- [63] Mohamed Sabry, Sites, Charles B., Farag, Aly A., Stephen Hushek, and Thomas Moriarty, Statistical Cerebrovascular segmentation for phase-contrast MRA data, *Proc. of the First International Conf. on Biomedical Engineering*, Cairo, Egypt, 2002.

Chapter 10

A Region-Aided Color Geometric Snake¹

Xianghua Xie² and Majid Mirmehdi²

10.1 Introduction

Deformable contour models or snakes are commonly used in image processing and computer vision due to their natural handling of shape variation and independence of operation (once initialized). A hypothesized contour, represented as a curve or surface, evolves under the influence of internal forces, external image-dependent forces, and certain constraints, till it converges on the object(s) of interest.

Generally, there are two types of snakes, parametric snakes and geometric snakes. The parametric model minimizes a deforming curve toward the pull of features such as edges and lines. The energy is composed of terms that control its smoothness and attract it to the object boundary. Although significant improvements have been made in this field over the last decade, parametric contours still suffer from imprecise shape representation. The geometric model of active contours, which avoids the need to parameterize the curve, has been hailed as the solution to topological problems. Geometric snakes are based on the theory of curve evolution and are numerically implemented via the level set algorithm. They are totally intrinsic, which means they can automatically handle topological changes without resorting to dedicated contour tracking, and unknown numbers of objects can be detected simultaneously. Furthermore, they can enjoy much larger capture areas than parametric snakes.

¹Portions reprinted, with permission, from IEEE T-IP, 13(5): 640–652 by the same authors.

²Xianghua Xie and Majid Mirmehdi Department of Computer Science, University of Bristol, England.

Whilst geometric or geodesic snakes go a long way in improving on parametric snakes, they still suffer from two significant shortcomings. First, they allow leakage into neighboring image regions when confronted with weak edges; hereafter we refer to this as the weak-edge leakage problem. Second, they may rest at local maxima in noisy image regions. In this chapter, both of these problems are dealt with by introducing diffused region forces into the standard geometric snake formulation. The proposed method is referred to as the region-aided geometric snake or RAGS. It integrates gradient flows with a diffused region vector flow. The gradient flow forces supplant the snake with local object boundary information, while the region vector flow force gives the snake a global view of object boundaries. The diffused region vector flow is derived from the region segmentation map which in turn can be generated from any image segmentation technique. This chapter demonstrates that RAGS can indeed act as a refinement of the results of the initial region segmentation. It also illustrates RAGS' weak edge leakage improvements and tolerance to noise through various examples. Using color edge gradients, RAGS will be shown to naturally extend to object detection in color images. The partial differential equations (PDEs) resulting from the proposed method will be implemented numerically using level set theory, which enables topological changes to be dealt with automatically.

In Section 10.2 we review the geometric snake model, encompassing its strength and its shortcomings. Section 10.3 provides a brief overview of the geometric GGVF snake, also outlining its shortcomings. The former section is essential as RAGS' theory is built upon it, and the latter is necessary since we shall make performance comparisons to it. Section 10.4 presents the derivation of the RAGS snake including its level set representation. Then, in Section 10.5, the numerical solutions for obtaining the diffused region force and level set implementation of RAGS are introduced. Section 10.6 describes the extension of RAGS to vector-valued images, again showing the equivalent level set numerical representation. Since RAGS is independent of any particular region segmentation method, its description so far is not affected by the fact that no discussion of region segmentation has yet taken place! This happens next in Section 10.7 where the mean shift algorithm is employed as a typical, suitable method for obtaining a region segmentation map for use with RAGS. Following a brief summary of the RAGS algorithm in Section 10.8, examples and results illustrating the improvements obtained on noisy images and images with weak edges are presented in Section 10.9. This includes an application with

quantitative results comparing the performance of RAGS against the standard geometric snake.

10.2 The Geometric Snake

Geometric active contours were introduced by Caselles *et al.* [1] and Malladi *et al.* [2] and are based on the theory of curve evolution. Using a reaction–diffusion model from mathematical physics, a planar contour is evolved with a velocity vector in the direction normal to the curve. The velocity contains two terms: a constant (hyperbolic) motion term that leads to the formation of shocks³ from which more varied and precise representations of shapes can be derived, and a (parabolic) curvature term that smooths the front, showing up significant features and shortening the curve. The geodesic active contour, hereafter also referred to as the *standard geometric snake*, is now introduced. Let $C(x, t)$ be a 2D active contour. The Euclidean curve shortening flow is given by

$$C_t = \kappa \vec{N}, \quad (10.1)$$

where t denotes the time, κ is the Euclidean curvature, and $\vec{N}$ is the unit inward normal of the contour. This formulation has many useful properties. For example, it provides the fastest way to reduce the Euclidean curve length in the normal direction of the gradient of the curve. Another property is that it smooths the evolving curve (see Fig. 10.1).

In [3,4], the authors unified curve evolution approaches with classical energy minimization methods. The key insight was to multiply the Euclidean arc length by a function tailored to the feature of interest in the image.

Let $I : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$ be an input image in which the task of extracting an object contour is considered. The Euclidean length of a curve C is given by

$$L := \oint |C'(q)| dq = \oint ds, \quad (10.2)$$

where ds is the Euclidean arc length. The standard Euclidean metric $ds^2 = dx^2 + dy^2$ of the underlying space over which the evolution takes place is modified to

³A discontinuity in orientation of the boundary of a shape; it can also be thought of as a zero-order continuity.

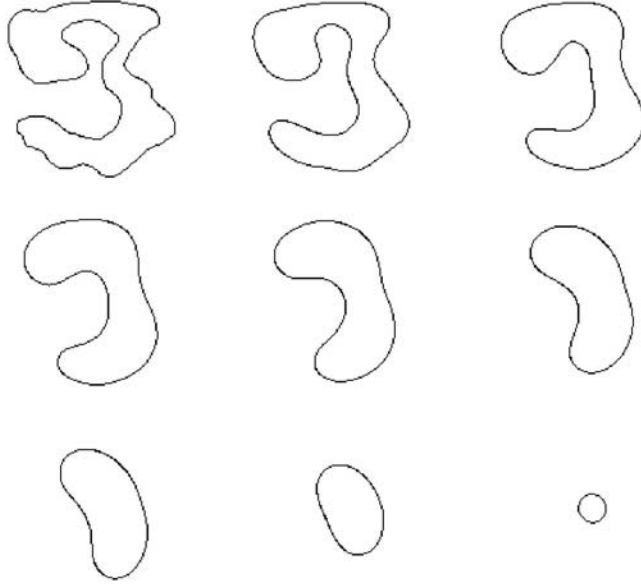


Figure 10.1: Motion under curvature flow: A simple closed curve will (become smoother and) disappear in a circular shape no matter how twisted it is.

a conformal metric given by

$$ds_g^2 = g(|\nabla I(C(q))|)^2(dx^2 + dy^2), \quad (10.3)$$

where $g(\cdot)$ represents a monotonically decreasing function such that $g(x) \rightarrow 0$ as $x \rightarrow \infty$, and $g(x) \rightarrow 1$ as $x \rightarrow 0$. A typical function for $g(x)$ can be

$$g(x) = \frac{1}{1+x}. \quad (10.4)$$

This is plotted in Fig. 10.2. Using this metric, a new length definition in Riemannian space is given by

$$L_{\mathfrak{R}} := \int_0^1 g(|\nabla I(C(q))|) |C'(q)| dq. \quad (10.5)$$

Then it is no longer necessary that the minimum path between two points in this metric be a straight line, which is the case in the standard Euclidean metric. The minimum path is now affected by the weighting function $g(\cdot)$. Two distant points in the standard Euclidean metric can be considered to be very close to each other in this metric if there exists a route along which values of $g(\cdot)$ are nearer to zero. The steady state of the active contour is achieved by searching

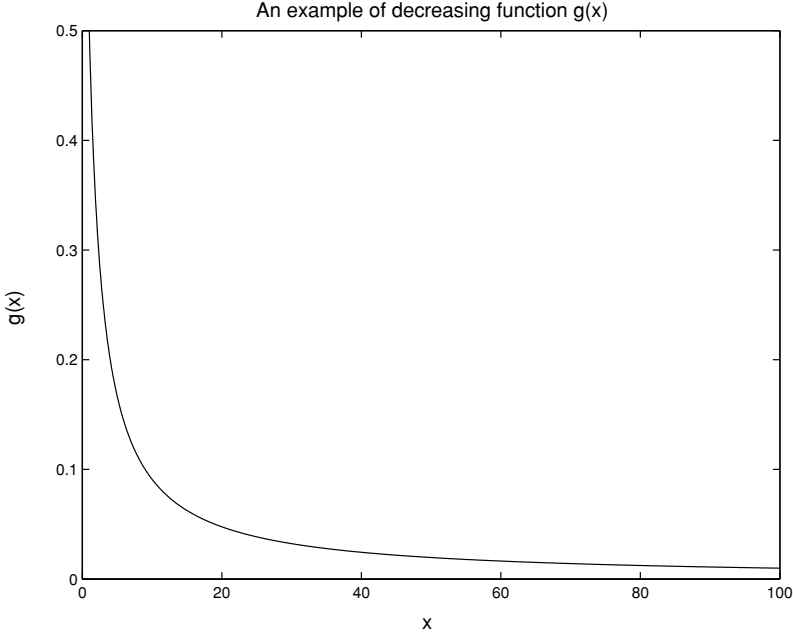


Figure 10.2: Plot of the monotonically decreasing function $g(x) = 1/(1+x)$.

for the minimum length curve in the modified Euclidean metric:

$$\min \int_0^1 g(|\nabla I(C(q))|) |C'(q)| dq. \quad (10.6)$$

Caselles *et al.* [4] have shown that this steady state is achieved by determining how each point in the active contour should move along the normal direction in order to decrease the length. The Euler–Lagrange of (10.6) gives the right-hand side of (10.7), i.e., the desired steady state:

$$C_t = g(|\nabla I|) \kappa \vec{N} - (\nabla g(|\nabla I|) \cdot \vec{N}) \vec{N}. \quad (10.7)$$

Two forces are represented by (10.7). The first is the curvature term multiplied by the weighting function $g(\cdot)$ and moves the curve toward object boundaries constrained by the curvature flow that ensure regularity during propagation. In application to shape modeling, the weighting factor could be an edge indication function that has larger values in homogeneous regions and very small values on the edges. Since (10.7) is slow, Caselles *et al.* [4] added a constant inflation term to speed up the convergence. The constant flow is given by $C_t = \vec{N}$ showing each point on the contour moves in the direction of its normal and on



Figure 10.3: Motion under constant flow: It causes a smooth curve to evolve to a singular one.

its own can cause a smooth curve to evolve to a singular one (see Fig. 10.3). However, integrating it into the geometric snake model lets the curvature flow (10.1) remain regular:

$$C_t = g(|\nabla I|)(\kappa + c)\vec{N} - (\nabla g(|\nabla I|) \cdot \vec{N})\vec{N}, \quad (10.8)$$

where c is a real constant making the contour shrink or expand to the object boundaries at a constant speed in the normal direction.

The second term of (10.7) or (10.8) depends on the gradient of the conformal factor and acts like a doublet (Fig. 10.4), which attracts the active contour further to the feature of interest since the vectors of $-\nabla g$ point toward the valley of $g(\cdot)$, the middle of the boundaries. This $-\nabla g$ increases the attraction of the active contour toward the boundaries. For an ideal edge, $g(\cdot)$ tends to zero. Thus, it



Figure 10.4: The doublet effect of the second term of Eq. 10.7. The gradient vectors are all directed toward the middle of the boundary, which forces the snake into the valley of $g(\cdot)$.

tries to force the curve to stop at the edge, but the convergence quality still highly depends on this stopping term. If $g(\cdot)$ is not small enough along edges, there will be an underlying constant force caused by c .

The geodesic or geometric active contour can be numerically implemented using level sets. This is demonstrated later in Section 10.4.4 when we deal with the extended formulation of the standard geometric snake into RAGS.

10.2.1 Examples of the Standard Geometric Snake

The standard geometric snake has been applied successfully in many application areas, not least in the medical imaging arena. Figure 10.5(left) shows an example of a geometric snake initialized in the stomach region of an abdominal section in a CT image. The final snake is shown in Fig. 10.5(right). In the next example an extension of the geometric snake for color images is shown in Fig. 10.6, a thigh slice from the Visible Human project. The figure on the left shows the initial snake as before and the final converged snakes are shown on the right, demonstrating the topological adaptation of the snake's level set implementation. Note that the top snake has failed to fully converge. Hence, while adequate for many situations, geometric snakes can suffer from certain shortcomings and the next section briefly deals with the nature of some such failings.

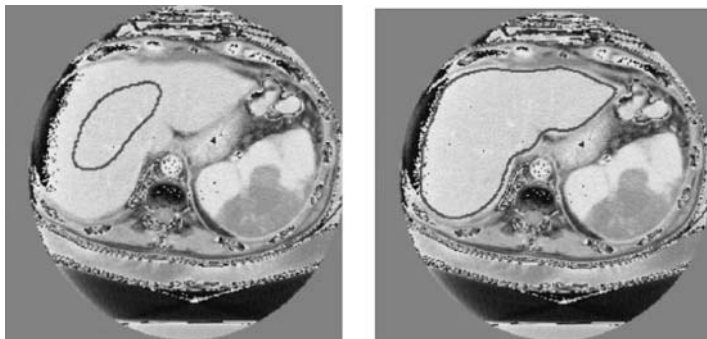


Figure 10.5: Example of geometric snake segmenting an inner boundary: recovery of the stomach region of an abdominal CT section—from [5], ©2003 IEEE.

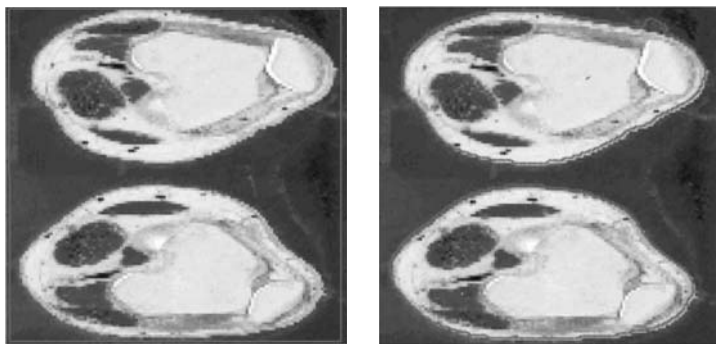


Figure 10.6: Example of a color image. (Left) original image with initial snake and (right) converged geometric snake (with a minor convergence problem in the top right corner of the snake)—from [6] (color slide).

10.2.2 Shortcomings of the Geometric Snake

Geometric active contour models have the significant advantage over classical snakes that changes in topology due to the splitting and merging of multiple contours are handled in a natural way. However, they suffer in two specific ways:

1. They use only local information and hence are sensitive to local minima. This means they are attracted to noisy pixels and can fail to converge on the desired object when they rest at such strong “features.” They fail to recognize, possibly weaker but true features further away in the image landscape, for lack of a better global understanding of the image. An example is shown in Fig. 10.7 (left).

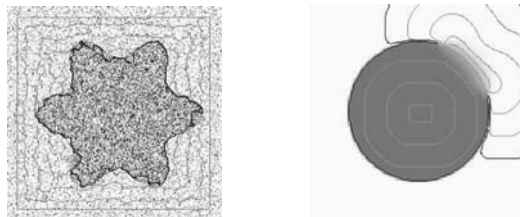


Figure 10.7: Noise sensitivity and weak-edge leakage problems. In each case the evolving snake is shown in a light color and the final snake in a dark one.

2. The constant flow term makes the snake expand or shrink. It can speed up the convergence and push the snake into concavities easily when the objects have good contrast, i.e. when the gradient magnitudes at object boundaries are large. However, when the object boundary is indistinct or has gaps, the snake tends to leak through the boundary mainly because of this constant force. The second term in (10.8) is designed to attract the contour further close to the object boundary and also to pull back the contour if it leaks through the boundary, but the force may just not be strong enough since it still depends on the gradient values. It cannot resolve the existence of a weak edge. Figure 10.7 (right) demonstrates this shortcoming of the standard geometric snake. The evolving of the snake is based on the gradient information, and as there is a gradual change of the intensity, the contour leaks through.

The result of such failures is that the geometric snake will converge to a nonsensical form. Both these effects are demonstrated in Fig. 10.8 where the cells contain fuzzy borders and strong but tiny dark “granules” that have led the standard geometric snake astray (top-right image). The images in the bottom

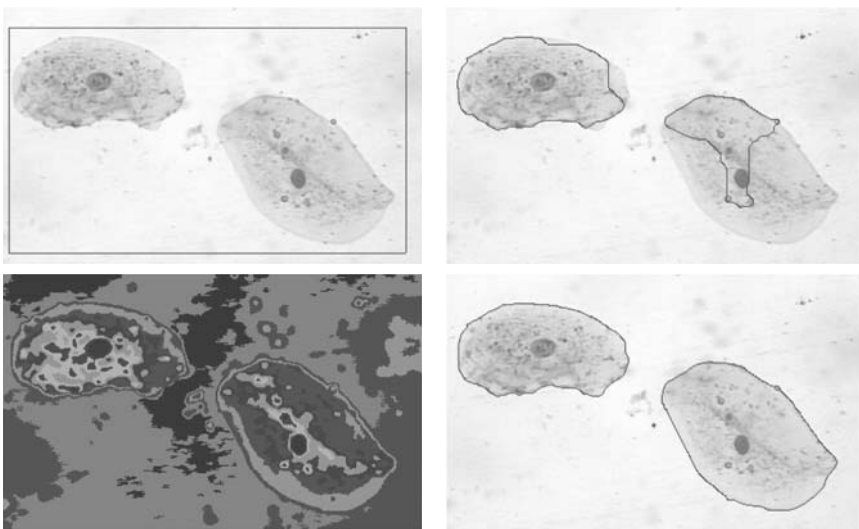


Figure 10.8: Multiple objects—top row: initial snake and standard geometric snakes, bottom row: region segmentation used by RAGS and converged RAGS snakes (original image courtesy of Dr. Douglas Kline, Department of Biological Sciences, Kent State University, US) (color slide).

row of Fig. 10.8 show the region map used for the RAGS formulation outlined later in this chapter and the converged RAGS snakes. This figure also illustrates the power of the geometric snake in splitting to find multiple objects.

10.3 The Geometric GGVF Snake

In this section we briefly introduce the geometric GGVF snake and consider its advantages and shortcomings. Later in the chapter, the GGVF snake will be used along with the standard geometric snake to make comparisons to the performance of RAGS.

The gradient vector flow (GVF) active contour was first introduced by Xu *et al.* [7] in a parametric framework. The authors proposed a new external force: *a diffusion of the gradient vectors of a gray level or binary edge map* derived from the original image. The GVF goes some way toward forcing a snake into boundary concavities while providing a larger capture range due to its diffused gradient vector field. Figure 10.9 (right) shows the diffused gradient vectors for a simple object in Fig. 10.9 (left). The traditional potential force is shown in Fig. 10.9 (center).

The same authors have also introduced the GGVF, a generalized GVF snake model. The GGVF improves the GVF by replacing the constant weighting factor with two spatially varying weighting functions, resulting in a new external force field. The weighting factors provide a trade-off between the smoothness of the GVF field and its conformity to the gradient of the edge map. The result is

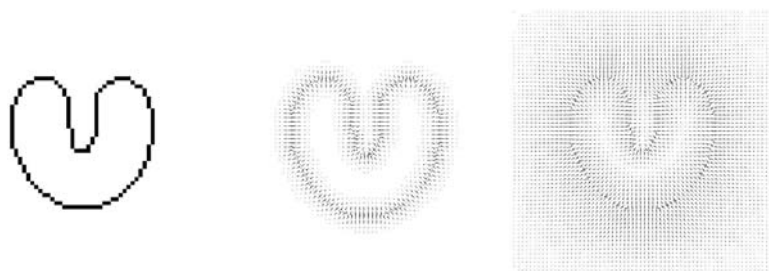


Figure 10.9: GVF field compared to traditional potential force vector field. From left: a simple line-drawing U shape (binary) image, the traditional potential force vector field, and GVF field (from [8], ©2003 IEEE).

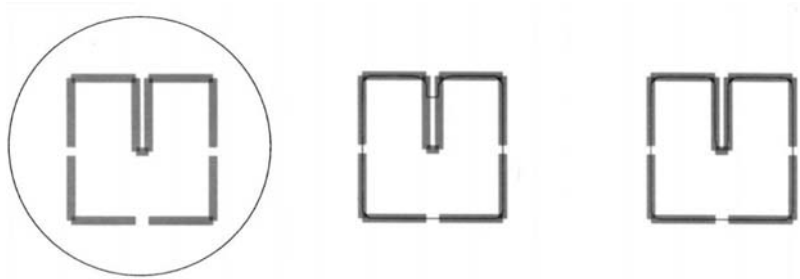


Figure 10.10: Concavity convergence comparison. From left: initial snake, GVF snake result, and GGVF snake result, from [9].

that contours can converge into long, thin boundary indentations. The GGVF preserves clearer boundary information while performing vector diffusion, while the GVF will diffuse everywhere within image. As shown in Fig. 10.10, the GGVF snake shows clear ability to reach concave regions.

Later in [10], Xu *et al.* showed the GGVF equivalence in a geometric framework. A simple bimodal region force generated as a two-class fuzzy membership function was added to briefly demonstrate weak-edge leakage handling. The geometric GGVF snake is useful when dealing with boundaries with small gaps. However, it is still not robust to weak edges, especially when a weak boundary is close to a strong edge, the snake readily steps through the weak edge and stops at the strong one. This is illustrated in Fig. 10.11 (left).

A further problem with the GGVF snake is that it does not always allow the detection of multiple objects. These topological problems arise, even though

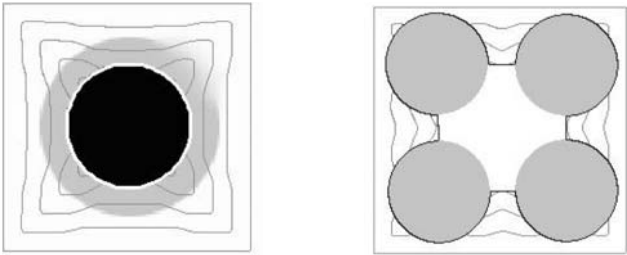


Figure 10.11: GGVF weaknesses. Left: The GGVF snake steps through a weak edge toward a neighboring strong one (final snake in white). Right: It also can encounter topological problems (final snake in black). The evolving snake is shown in a lighter color in both cases.

the GGVF snake was specified in the geometric model, when the vector field is tangent to the snake contour. In such cases there would be no force to push or pull it in the perpendicular direction (to the vectors). This effect is shown in Fig. 10.11 (right).

10.4 Region-Aided Geometric Snake

We now describe a novel approach to make the geometric snake much more tolerant toward weak edges and image noise. It comprises the integration of gradient flow forces with diffused region forces in the image, resulting in the region-aided geometric snake:

- The gradient flow forces supplant the snake with local object boundary information. They play a main role in all active contours⁴.
- The region forces are based on the global image features and supplant the snake with global image information.

We show that this combination of forces not only improves the performance of the geometric snake toward weak edges, but also makes it more immune to noise. The PDE thus obtained evolves an initial contour toward final convergence under the influence of both internal forces and boundary-regional image forces, and is implemented via level sets.

The proposed region force can be generated from any image segmentation technique. This means that while RAGS is independent of any particular segmentation technique, it is dependent on the quality of the regions produced. However, we show a good degree of tolerance to (reasonable) segmentation quality, and that our snake indeed acts as a refinement of the results of the initial region segmentation. Later in Section 10.7, we introduce the mean shift segmentation technique presented by Comaniciu *et al.* in [12, 13] which is a very elegant method to generate region maps for this work. Results will be presented based on region maps obtained from both the under-segmentation and over-segmentation options of the software from Comaniciu and Meer's study.

⁴There are notable exceptions to this, e.g. [11].

10.4.1 Gradient Flow Force: A Summary

As mentioned earlier, the gradient flows impose local constraints while the region force contributes global constraints. Within a homogeneous region of an image, measured by region segmentation, the snake evolves mainly according to gradient flows. The first gradient flow is the weighted length gradient flow, which is given by (10.7). It is composed of two terms. The first is the weighted curvature term, $g(|\nabla I|)\kappa\tilde{\mathcal{N}}$, which smooths the active contour and also shrinks it. The second term, $(\nabla g(|\nabla I|) \cdot \tilde{\mathcal{N}})\tilde{\mathcal{N}}$, is on the normal factor of the gradient of the weighting function. Unlike the curvature, the vector field $\nabla g(|\nabla I|)$ is static. The direction and strength of this field depend on position only, and is independent of time and contour.

The second gradient flow, $g(|\nabla I|)c\tilde{\mathcal{N}}$, is introduced by constant motion which locally minimizes area (see [14] for proof). It helps the snake shrink or expand toward object boundaries and accelerates its convergence speed.

For all these forces, the weighting function g can be defined as any decreasing function of the image I edge map f such that $g \rightarrow 0$ as $f \rightarrow \infty$. When dealing with gray level images, the solution (as used in this work) is straightforward:

$$f = |\nabla(\text{Gauss} * I)| \quad \text{and} \quad g = \frac{1}{1 + f}. \quad (10.9)$$

This monotonically decreasing nature is illustrated in Fig. 10.2. As for color images, the edge function f becomes a little more intricate (an example function will be presented in Section 10.6). However, the derivation of the decreasing function g can remain the same.

10.4.2 Diffused Region Force

The aim of generating a region force is to empower the snake with a global view of image features. A typical region segmentation method splits an image into several regions, giving the segmentation map S . From this, the region map R is generated by computing the gradient of S . The gradient computation is the same as the edge computation stage for generating gradient forces. Then, we compute the gradient ∇R of this region map, resulting in region constraints in the vicinity of the region boundaries. Having slithered across a homogeneous region reliant on the gradient flow forces, if the snake tries to step from one region into another, it must concur with the region force in ∇R since it breaks the region

criteria, which probably indicates a leakage. The force field ∇R has vectors pointing toward the center of the region boundaries. The capture area of this pure region force is quite small: only immediate areas close to region boundaries. The vectors need to be diffused further away from the region boundaries to create a larger capture field. To achieve this, we can diffuse ∇R resulting in region forces with a larger capture area along the region boundaries. Hence, the region force vector field $[\tilde{R}(z) = (u(z), v(z)), z = (x, y)]$ is obtained by solving the following equations:

$$\begin{cases} p(|\nabla R|)\nabla^2 u - q(|\nabla R|)(u - \nabla R_u) = 0 \\ p(|\nabla R|)\nabla^2 v - q(|\nabla R|)(v - \nabla R_v) = 0 \end{cases}, \quad (10.10)$$

where ∇^2 is the Laplacian operator with dimensions u and v , $p(\cdot)$ and $q(\cdot)$ are weighting functions that control the amount of diffusion, and ∇R_u and ∇R_v are the components of vector field ∇R along the u and v directions⁵. The weighting functions are selected such that $p(\cdot)$ gets smaller as $q(\cdot)$ becomes larger with the desirable result that in the *proximity* of large gradients, there will be very little smoothing and the vector field will be nearly equal to the gradient of the region map. We use the following functions for diffusing the region gradient vectors:

$$\begin{cases} p(|\nabla R|) = e^{-(|\nabla R|/K)} \\ q(|\nabla R|) = 1 - p(|\nabla R|) \end{cases}, \quad (10.11)$$

where K is a constant and acts as a trade-off between field smoothness and gradient conformity. The solution of (10.10) is the equilibrium state of the following partial differential equations:

$$\begin{cases} u_t = p(|\nabla R|)\nabla^2 u - q(|\nabla R|)(u - \nabla R_u) \\ v_t = p(|\nabla R|)\nabla^2 v - q(|\nabla R|)(v - \nabla R_v) \end{cases}, \quad (10.12)$$

where u and v are treated as functions of time. These partial differential equations can be implemented using an explicit finite difference scheme. An iterative process can be set up and guaranteed to converge with the following constraint

$$\Delta t \leq \frac{\Delta x \Delta y}{4p_{\max}}, \quad (10.13)$$

⁵Theoretically, ∇R can be diffused in any two orthogonal directions, u and v , within the image domain. However, practically we will only choose x and y directions corresponding to image plane coordinates. Thus ∇R_u and ∇R_v are equal to $\frac{\delta R}{\delta x}$ and $\frac{\delta R}{\delta y}$ respectively.

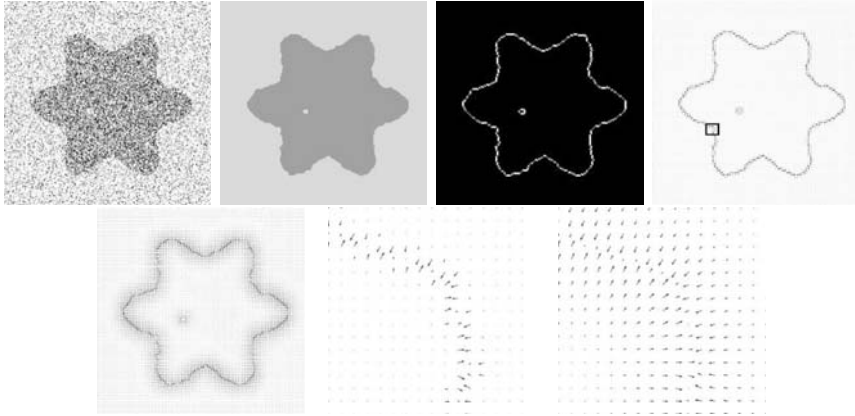


Figure 10.12: Region force diffusion—top row: A synthetic image with additive Gaussian noise, region segmentation map, region boundary map R , and gradient of the region map R (and a small selected area)—bottom row: diffused region vector field, and close-up views in the small selected area of the vectors in the gradient of region map and the diffused region vector field respectively.

where Δx and Δy are the spatial sample intervals, $p_{\max}$ is the maximum of $p(\cdot)$, and Δt is time step, the interval between time t_n and time t_{n+1} when iteratively solving (10.12).

From (10.11) and (10.12) we note that within a homogeneous region, based on the criteria of region segmentation, $p(\cdot)$ equals 1 while $q(\cdot)$ equals 0. Thus (10.12) is only left with the first term (as the second term vanishes). This effectively smooths the vector field. However, at the region boundaries, $p(\cdot) \rightarrow 0$ and $q(\cdot) \rightarrow 1$. The smoothing term imposes less and the region vectors are close to the gradient of the region map R . Thus the diffused region vector field provides the evolving snake with an attracting force in a sufficiently large range near the region boundaries, and also allows the snake to evolve solely under other gradient forces.

Figure 10.12 illustrates an example of region force diffusion, including close-up views of pre- and post-diffusion vector field.

10.4.3 Region-Aided Snake Formulation

Next, we can derive the region-aided geometric snake formulation. The standard geometric snake is given by (10.8). In the traditional sense, the snake forces fall

into two types, internal forces and external forces. The internal forces impose regularity on the curve and control the elasticity and rigidity of the snake. The external forces pull the snake toward salient image features such as object boundaries. Thus, the internal and external forces in (10.8) can be written as

$$\begin{cases} F_{\text{int}} = g(|\nabla I|)\kappa\vec{\mathcal{N}} \\ F_{\text{ext}} = g(|\nabla I|)c\vec{\mathcal{N}} - \nabla g(|\nabla I|) \end{cases}, \quad (10.14)$$

where $g(\cdot)$ is the stopping function as before. The first term of the external forces is a constant shrink or expand force in the normal direction of the snake. It can be separated from other external forces in the sense that it is not spatially static in the image domain as other external forces and needs different numerical schemes. However, considering the previous definition of snake forces and that the constant force alone can push the snake toward boundaries, we keep it in the external term.

The diffused region force is a feature driven force and spatially static. So we can add the diffused region force to the external term:

$$\begin{cases} F_{\text{int}} = g(|\nabla I|)\kappa\vec{\mathcal{N}} \\ F_{\text{ext}} = \alpha g(|\nabla I|)\vec{\mathcal{N}} + \beta\vec{R} - \nabla g(|\nabla I|) \end{cases}, \quad (10.15)$$

where $\vec{R}$ is the region force vector field obtained in (10.10) and α is a new constant incorporating c . Constants α and β act as a trade-off between gradient forces and region forces. In practice, β is a constant from 0 to 1 for most nonhighly textured images. If good segmentation results are available, β should be set close to 1.

The snake evolves under all the internal and external forces. However, only the forces in the normal direction of the evolving contours can change the geometry. The forces tangential to the contours can only change the parameterization of the contours. Thus, a geometric snake evolving under internal and external forces can be interpolated as

$$C_t = [(F_{\text{int}} + F_{\text{ext}}) \cdot \vec{\mathcal{N}}]\vec{\mathcal{N}}. \quad (10.16)$$

Finally, by substituting (10.15) into (10.16), the region-aided geometric snake formulation becomes

$$C_t = [g(|\nabla I|)(\kappa + \alpha) - \nabla g(|\nabla I|) \cdot \vec{\mathcal{N}} + \beta\vec{R} \cdot \vec{\mathcal{N}}]\vec{\mathcal{N}}. \quad (10.17)$$

10.4.4 Level Set Representation

In this section, we outline the level set representation for the region-aided geometric snake. Level sets describe a moving front in an implicit function and are the basis for the numerical algorithm for curve evolution according to functions of curvature, introduced by Osher *et al.* [15, 16]. In the application to active contours, the evolving contour is embedded into a higher dimensional surface as a zero level set. The entire surface, the level sets, is an implicit representation of the embedded contour. As shown in Fig. 10.13, the snake is initially built in a three-dimensional surface, which later evolves according to underlying forces. Finally, the converged snake is extracted from the level sets by cutting it at zero height.

Let C be a level set of a function of $\phi : [0, a] \times [0, b] \rightarrow \Re$. That is, C is embedded into the zero level set with ϕ being an implicit representation of the curve C . This representation is parameter free and intrinsic. Given a planar curve that evolves according to $C_t = \mathbb{F}\vec{N}$ for a given function $\mathbb{F}$, the embedding function should deform according to $\phi_t = \mathbb{F}|\nabla\phi|$, where $\mathbb{F}$ is computed on the level sets. By embedding the evolution of C in that of ϕ , topological changes

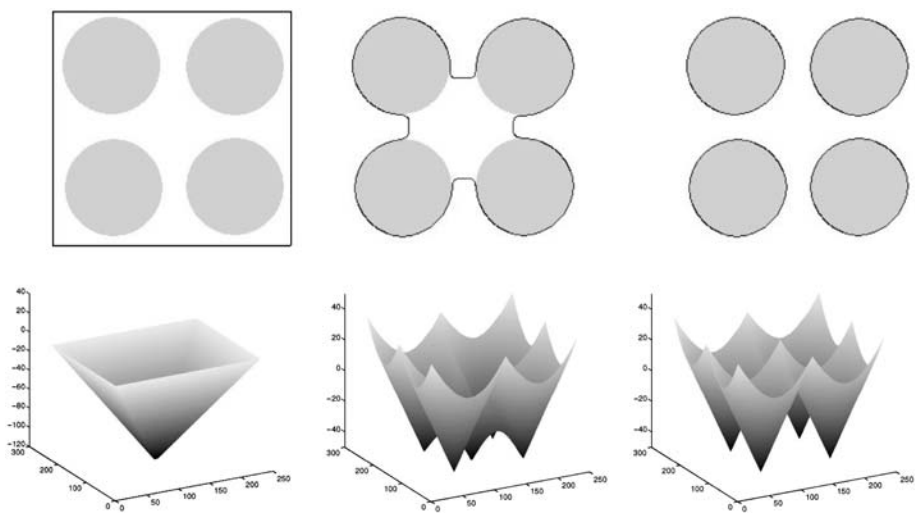


Figure 10.13: Level sets evolution for an embedded snake. Top row: initial snake on test image, evolving contour, and final converged snake. Bottom row: corresponding evolving level sets. The snake is tracked at zero height.

of C are handled automatically and accuracy and stability are achieved using numerically stable computations.

The internal curvature and external pressure terms of the RAGS formulation in (10.17) can be easily transferred to a level set representation:

$$\begin{cases} C_t = g(|\nabla I|)\kappa\tilde{\mathcal{N}} \rightarrow \phi_t = g(|\nabla I|)\kappa|\nabla\phi| \\ C_t = g(|\nabla I|)\alpha\tilde{\mathcal{N}} \rightarrow \phi_t = g(|\nabla I|)\alpha|\nabla\phi| \end{cases}, \quad (10.18)$$

The other external forces in (10.17) are static vector fields derived from image data which do not change as the active contour deforms. Static force fields are defined on the spatial positions rather than the active contour itself. Since $\tilde{\mathcal{N}}$ is the inward normal, the level set representation of the inward unit normal is given by

$$\tilde{\mathcal{N}} = -\frac{\nabla\phi}{|\nabla\phi|}. \quad (10.19)$$

Then, we have

$$\mathbb{F} \cdot \tilde{\mathcal{N}} = -\frac{1}{|\nabla\phi|}(\mathbb{F} \cdot \nabla\phi). \quad (10.20)$$

Combining (10.18) with (10.20) where $\mathbb{F}$ takes on the static force fields, the level set representation of RAGS is given by

$$\phi_t = g(|\nabla I|)(\kappa + \alpha)|\nabla\phi| + \nabla g(|\nabla I|) \cdot \nabla\phi - \beta\tilde{\mathcal{R}} \cdot \nabla\phi, \quad (10.21)$$

where $g(\cdot)$ is the stopping function as before. The expression for the curvature of the zero level set assigned to the interface itself is given by

$$\kappa = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_y\phi_x\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \quad (10.22)$$

10.5 Numerical Solutions

The numerical solution for region force diffusion is discussed in detail in Section 10.5.1, but the detailed numerical solutions for RAGS level set representation are only presented in Appendix A as they are not critical to the understanding of the concepts underlying RAGS. In fact, the whole of this section can be skipped without loss of continuity.

10.5.1 Numerical Solutions for Region Force Diffusion for RAGS

Initially, a mesh grid needs to be selected, with final accuracy directly dependent on its resolution. However, due to the nature of a digital image, the grid resolution is constrained to the pixel level. It was shown in Section 10.4.2 that the steady solution of (10.10) can be achieved by computing the equilibrium state of (10.12). The initial state of the region force vector field $\tilde{R}$ is given by the gradient of the region map R . Simple central differences can be used to approximate ∇R , resulting in vectors that are then diffused. Let Δx and Δy be the grid spacing, Δt be the time step, and i, j , and n represent the spatial position and time. The partial derivative of time can be approximated by forward difference as

$$u_t = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}. \quad (10.23)$$

The spatial partial derivatives can be solved using central differences approximation given by

$$\nabla^2 u = \frac{u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j}}{\Delta x \Delta y}. \quad (10.24)$$

The solutions to partial derivatives of $v(x, y, t)$ are similar to those of $u(x, y, t)$. The weighting functions given in (10.11) can be easily computed. Thus, substituting the partial derivatives into (10.12) gives the following iterative solution:

$$\begin{cases} u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \Lambda \\ v_{i,j}^{n+1} = v_{i,j}^n + \Delta t \Omega \end{cases}, \quad (10.25)$$

where,

$$\Lambda = \frac{p(\cdot)_{i,j}}{\Delta x \Delta y} (u_{i+1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n + u_{i-1,j}^n - 4u_{i,j}^n) - q(\cdot)_{i,j} (u_{i,j}^n - R_{x,ij})$$

and

$$\Omega = \frac{p(\cdot)_{i,j}}{\Delta x \Delta y} (v_{i+1,j}^n + v_{i,j+1}^n + v_{i,j-1}^n + v_{i-1,j}^n - 4v_{i,j}^n) - q(\cdot)_{i,j} (v_{i,j}^n - R_{y,ij}),$$

where $R_{x,ij}$ and $R_{y,ij}$ are partial derivatives of R . They can also be approximated by central differences as

$$\begin{cases} R_{x,ij} = \frac{R_{i+1,j} - R_{i-1,j}}{2\Delta x} \\ R_{y,ij} = \frac{R_{i,j+1} - R_{i,j-1}}{2\Delta y} \end{cases}. \quad (10.26)$$

The convergence is guaranteed with the time step restriction of (10.13).

10.5.2 Numerical Solution for the Level Set Implementation of RAGS

As in the numerical solution for vector diffusion, a computational grid is required. Once the grid is chosen, the initial level sets $\phi(x, t) = 0$ can be defined with the property that the zero level set corresponds to the initial contours of the snake. The signed-distance transform can be used to build the initial level sets. A brute-force Euclidean distance transform would be computationally infeasible. Practically, accuracy is required only near the initial contours, and discrete values based on grid distance can suffice further away. A positive sign is given to the points outside the contours, and a negative sign is applied to the points inside.

As shown in (10.17), the snake evolves according to four forces that can be categorized into three types based on the nature of their motions:

1. The first motion is of a smoothing and collapsing nature with speed proportional to its curvature as shown in Fig. 10.1. It can be numerically approximated using central differences, because the curvature is only dependent on the contour. It is independent of time and spatial position.
2. The second is expanding or shrinking with a spatially constant speed, characterized by $\alpha g(\cdot)$ in the normal direction of the curve. However, when the constant term exists, the normals can collide with each other while evolving. Thus shocks, or corners, will form and once a shock has developed, some information will be lost as it evolves. This means that shocks cause irreversibility; information cannot be recovered by tracing 'backwards' in time. Generally, no new information can be created while evolving, which is referred to as an *entropy condition*. Central difference approximation cannot be used to approximate the gradient in this case, as it suffers from shocks where the entropy condition is invoked. An *upwind scheme* can be used, as an entropy-satisfying scheme, that engages information upwind of the direction of its propagation. In other words, in order to achieve a stable numerical scheme, the numerical domain of dependence should contain the mathematical domain of dependence. Thus, in order to approximate the gradient of the constant term, it is important to first know which way the speed function points, and whether it is negative or positive. Then we can choose proper backward or forward difference approximations.

3. The third type of motion in (10.17) is contributed by the underlying static velocity field, the direction and strength of which are based on spatial position. It is independent of the shape and position of the snake. The motion of contours under this velocity field can be numerically approximated through *upwind schemes* by checking the sign of each component of the velocity field and constructing one-sided upwind differences in the appropriate direction. For a positive speed component, backward difference approximation is used, otherwise forward difference approximation should be applied.

By using these approximation schemes, (10.17) can be numerically implemented. The detailed numerical solutions for RAGS are presented in Appendix A. For general numerical solution to level sets, including concepts such as *entropy condition* and *upwind scheme*, the interested reader is referred to works by Sethian [16, 17] and by Osher *et al.* [18].

10.6 Region-Aided Geometric Snake on Vector-Valued Images

The theory of boundary detection by the geometric or geodesic snake can be applied to any general “edge detector” function. The stopping function g should tend to zero when reaching edges.

When dealing with gray level images, the decreasing function g can be easily derived from the edge detector f , as shown in (10.9). We use a similar stopping function for edges obtained directly from vector-valued images such as a color image.

A consistent extension of scalar gradients based on a solid theoretical foundation has been presented by di Zenzo [19]. This extension has been applied in the active contour literature to both geometric and parametric snakes.

In a vector-valued image the vector edge is considered as the largest difference between eigenvalues in the tensor metric. Let $\Theta(u_1, u_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^m$ be an m -band image for $i = 1, 2, \dots, m$. For color images, $m = 3$. A point in the image is considered as a vector in $\mathbb{R}^m$. The distance between two points, $P = (u_1^0, u_2^0)$ and $Q = (u_1^1, u_2^1)$, is given by $\Delta\Theta = \Theta(P) - \Theta(Q)$. When this distance tends to be infinitesimal, the difference becomes the differential $d\Theta = \sum_{i=1}^2 \frac{\partial\Theta}{\partial u_i} du_i$ with

its squared norm given by

$$d\Theta^2 = \sum_{i=1}^2 \sum_{j=1}^2 \frac{\partial \Theta}{\partial u_i} \frac{\partial \Theta}{\partial u_j} du_i du_j. \quad (10.27)$$

Using standard Riemannian geometry notation, let $s_{ij} = \frac{\partial \Theta}{\partial u_i} \cdot \frac{\partial \Theta}{\partial u_j}$, such that

$$d\Theta^2 = \sum_{i=1}^2 \sum_{j=1}^2 s_{ij} du_i du_j = \begin{bmatrix} du_1 \\ du_2 \end{bmatrix}^T \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} du_1 \\ du_2 \end{bmatrix}. \quad (10.28)$$

For a unit vector $v = (\cos \theta, \sin \theta)$, then $d\Theta^2(v)$ indicates the rate of change of the image in the direction of v . The extrema of the quadratic form are obtained in the directions of the eigenvectors of the metric tensor s_{ij} , and the corresponding eigenvalues are

$$\lambda_{\pm} = \frac{s_{11} + s_{22} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}}{2} \quad (10.29)$$

with eigenvectors $(\cos \theta_{\pm}, \sin \theta_{\pm})$ where the angles $\theta_{\pm}$ are given by

$$\begin{cases} \theta_+ = \frac{1}{2} \arctan \frac{2s_{12}}{s_{11} - s_{22}} \\ \theta_- = \theta_+ + \frac{\pi}{2} \end{cases}. \quad (10.30)$$

The maximal and minimal rates of change are the λ_+ and λ_- eigenvalues respectively, with corresponding directions of change being θ_+ and θ_- . The strength of an edge in a vector-valued case is not given simply by the rate of maximal change λ_+ , but by the difference between the extrema. Hence, a good approximation function for the vector edge magnitude should be based on $f = f(\lambda_+, \lambda_-)$. Now RAGS can be extended to the region-aided geometric color snake by selecting an appropriate edge function f_{col} . The edge stopping function g_{col} is defined such that it tends to 0 as $f_{\text{col}} \rightarrow \infty$. The following functions can be used (cf. (10.9)):

$$f_{\text{col}} = \lambda_+ - \lambda_- \quad \text{and} \quad g_{\text{col}} = \frac{1}{1 + f_{\text{col}}}. \quad (10.31)$$

Then replacing $g_{\text{col}}(\cdot)$ for the edge stopping term $g(\cdot)$ in (10.17), we have the color RAGS snake:

$$C_t = [g_{\text{col}}(|\nabla I|)(\kappa + \alpha) - \nabla g_{\text{col}}(|\nabla I|) \cdot \vec{\mathcal{N}} + \beta \vec{R} \cdot \vec{\mathcal{N}}] \vec{\mathcal{N}}. \quad (10.32)$$

Finally, its level set representation is also given by replacing $g_{\text{col}}(\cdot)$ for $g(\cdot)$ in (10.21):

$$\phi_t = g_{\text{col}}(|\nabla I|)(\kappa + \alpha)|\nabla \phi| + \nabla g_{\text{col}}(|\nabla I|) \cdot \nabla \phi - \beta \tilde{R} \cdot \nabla \phi. \quad (10.33)$$

10.7 The Mean Shift Algorithm

This section can be skipped without loss of continuity. Its topic is the process of generating the image region segmentation map S which is then used as described in Section 10.4.2. The reader can assume it is available and skip to the next section.

An essential requisite for RAGS is a segmentation map of the image. This means that RAGS is independent of any particular segmentation technique as long as a region map is produced; however, it is dependent on its representational quality. In this section, the mean shift algorithm is reviewed as a robust feature space analysis method which is then applied to image segmentation. It provides very reasonable segmentation maps and has extremely few parameters that require tuning.

The concept underlying the nonparametric mean shift technique is to analyze the density of a feature space generated from some input data. It aims to delineate dense regions in the feature space by determining the modes of the unknown density, i.e. first the data is represented by local maxima of an empirical probability density function in the feature space and then its modes are sought. The denser regions are regarded as significant clusters. Comaniciu *et al.* [13, 20] have recently provided a detailed analysis of the mean shift approach, including the review below, and presented several applications of it in computer vision, e.g. for color image segmentation.

We now briefly present the process of density gradient estimation. Consider a set of n data points $\{x_i\}_{i=1, \dots, n}$ in the d -dimensional Euclidean space R^d . Also consider the Epanechnikov kernel, an optimum kernel yielding minimum mean integrated square error:

$$K(x) = \begin{cases} \frac{1}{2Z_d}(d+2)(1-x^T x), & \text{if } x^T x < 1 \\ 0, & \text{otherwise} \end{cases}, \quad (10.34)$$

where Z_d is the volume of the unit d -dimensional sphere. Using $K(x)$ and window radius h , the *multivariate kernel density estimate* on the point x is

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right). \quad (10.35)$$

The estimate of the density gradient can be defined as the gradient of the kernel density estimate since a differentiable kernel is used:

$$\hat{\nabla} f(x) \equiv \nabla \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n \nabla K\left(\frac{x - x_i}{h}\right). \quad (10.36)$$

Applying (10.34) to (10.36), we obtain

$$\hat{\nabla} f(x) = \frac{n_x}{n(h^d Z_d)} \frac{d+2}{h^2} \left(\frac{1}{n_x} \sum_{x_i \in H_h(x)} [x_i - x] \right), \quad (10.37)$$

where the region $H_h(x)$ is a hypersphere of radius h and volume $h^d Z_d$, centered on x , and containing n_x data points. The *sample mean shift* is the last term in (10.37)

$$M_h(x) \equiv \frac{1}{n_x} \sum_{x_i \in H_h(x)} [x_i - x]. \quad (10.38)$$

The quantity $\frac{n_x}{n(h^d Z_d)}$ is the kernel density estimate $\hat{f}(x)$ computed with the hypersphere $H_h(x)$, and thus (10.37) can be rewritten as

$$\hat{\nabla} f(x) = \hat{f}(x) \frac{d+2}{h^2} M_h(x), \quad (10.39)$$

which can be rearranged as

$$M_h(x) = \frac{h^2}{d+2} \frac{\hat{\nabla} f(x)}{\hat{f}(x)}. \quad (10.40)$$

Using (10.40), the mean shift vector provides the direction of the gradient of the density estimate at x which always points toward the direction of the maximum increase (in the density). Hence, it converges along a path leading to a mode of the density.

In [13], Comaniciu *et al.* performed the mean shift procedure for image segmentation in a joint domain, the image (*spatial*) domain, and color space (*range*) domain. The spatial constraints were then inherent in the mode searching procedure. The window radius is the only significant parameter in their segmentation scheme. A small window radius results in oversegmentation (i.e. larger number of clusters), and a large radius produces undersegmentation (yielding a smaller

number of clusters). In this work, the performance of RAGS will be demonstrated on both the undersegmentation and oversegmentation resolutions of Comaniciu and Meer's work. In either case, the result of the mean shift procedure is the region segmentation map S which is passed to RAGS for generating the diffused region boundary map $\tilde{R}$.

10.8 A Summary of the RAGS Algorithm

The color RAGS algorithm is now reviewed with the aid of Fig. 10.14. Given the input color image, two streams of processing can begin concurrently.

- In the first stream, the vector gradient is computed to provide the edge function f , which is then used in (10.9) to yield the decreasing function g , followed by ∇g . Function g will act as spatial weights for the snake curvature force and constant force, and ∇g will contribute to the underlying doublet attraction force.
- In the second stream, a region segmentation map S is produced by applying any reasonable segmentation technique, e.g. the mean shift algorithm. From it, region map R can then be generated using vector gradients. Gradient of the region map R provides ∇R , which imposes region forces immediate to region boundaries. These region forces are then diffused by solving (10.10), resulting in a region force vector field $\tilde{R}$.

Thus, all the underlying velocity fields and the weighting function g are ready and prepared. Then we can generate initial level sets based on an initial snake using the distance transform and evolve the level sets according to all force fields (rightmost part of Fig. 10.14). The curvature force and constant force adaptively change with the level set snake. Along with the static forces, they are numerically solved using the principles described in Section 10.5.2 with the solutions given in Appendix A. After the level set evolves to a steady state, the final snake is easily obtained by extracting the zero level set.

10.9 Experiments and Results

In this section we present results that show improvements over either the standard geometric snake or the geometric GGVF snake or both, and mainly in

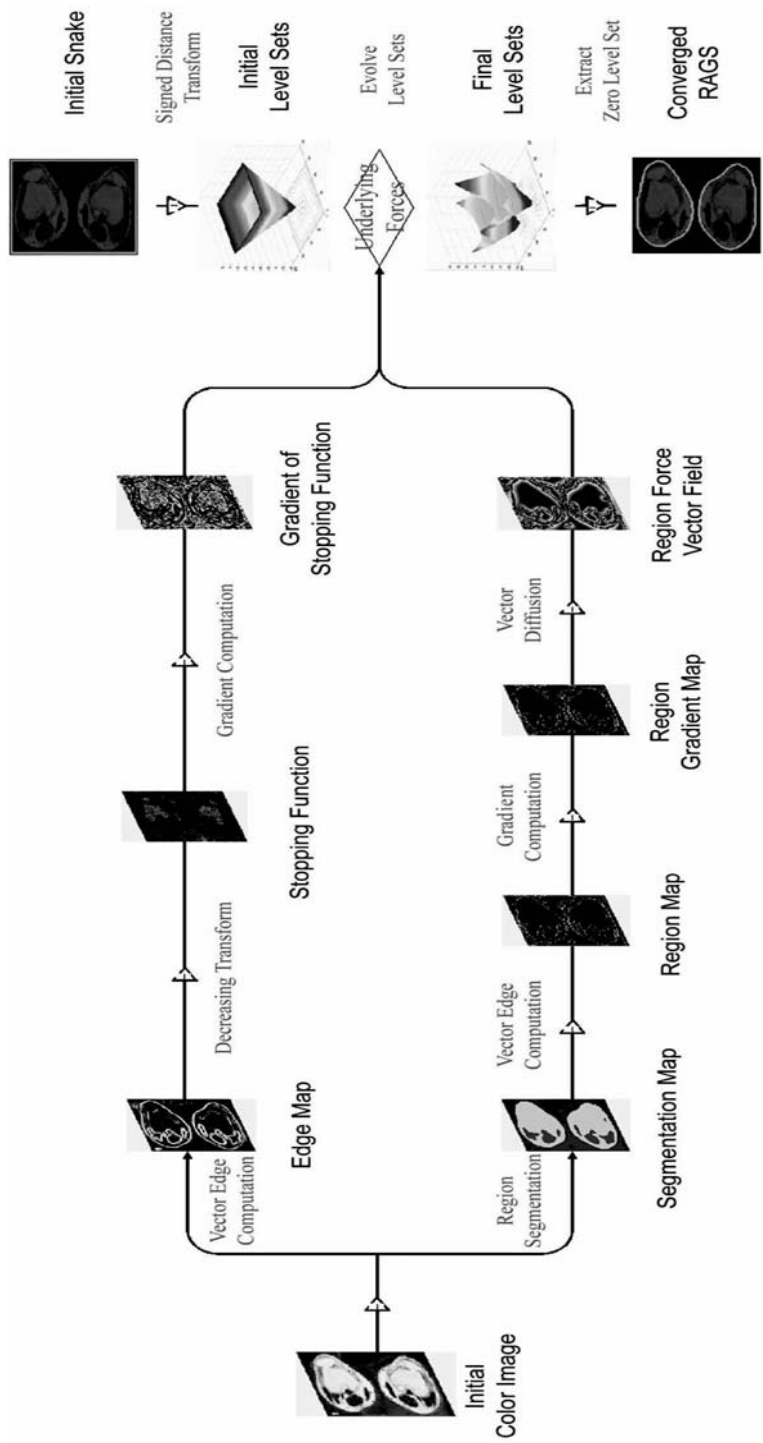


Figure 10.14: RAGS processing schema (color slide).

images where there are weak edges or noisy regions preventing the aforementioned snakes to perform at their best. Although GGVPs have been reported only using gray level image gradients, we can also apply them to “color” gradients (obtained as described in Section 10.6), which allows direct comparison with the color RAGS. It must also be noted that the GGVP can sometimes perform better than we have shown in some of the following examples as long as it is initialized differently, i.e. much closer to the desired boundary. In all the experiments, we have initiated the geometric, GGVP, and RAGS snakes at the same starting position, unless specifically stated.

10.9.1 Preventing Weak-Edge Leakage

We first illustrate the way weak-edge leakage is handled on a synthetic image. The test object is a circular shape with a small blurred area on the upper right boundary as shown in Fig. 10.15.

The standard geometric snake steps through the weak edge because the intensity changes so gradually that there is no clear boundary indication in the edge map. The RAGS snake converges to the correct boundary since the extra diffused region force delivers useful global information about the object boundary and helps prevent the snake from stepping through. Figure 10.16 shows, for the test object in Fig. 10.15, the edge map, the stopping function $g(\cdot)$, its gradient magnitude $|\nabla g(\cdot)|$, the region segmentation map S , and the vector map of the diffused region force $\tilde{R}$.

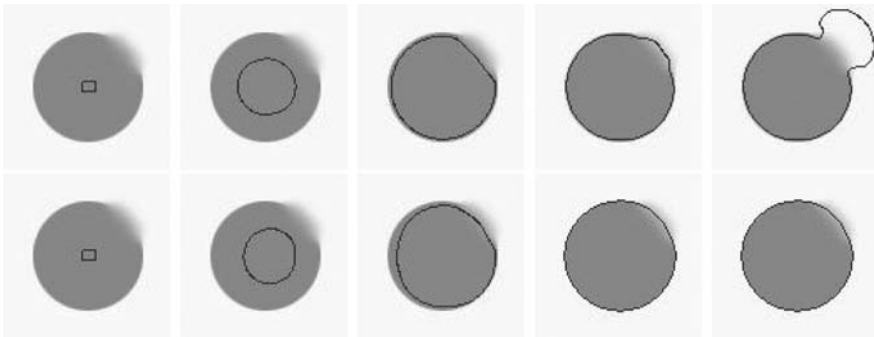


Figure 10.15: Weak-edge leakage testing on a synthetic image. Top row: geodesic snake steps through. Bottom row: RAGS snake converges properly using its extra region force.



Figure 10.16: Diffused region force on weak edge. From left: the edge map, the stopping function $g(\cdot)$ of edge map, the magnitude of its gradient $\nabla g(\cdot)$, the region segmentation map, and the vector map of the diffused region force $\tilde{R}$.

10.9.2 Neighboring Weak/Strong Edges

The next experiment is designed to demonstrate that both the standard geometric snake and the GGVF snake readily step through a weak edge to reach a neighboring strong edge. The test object in Fig. 10.17 contains a prominent circle inside a faint one. The presence of the weaker edge at the outer boundary is detected only by the RAGS snake. The geodesic snake fails because the weaker outer boundary allows the whole snake to leak through (similar to but in the opposite direction of propagation in Fig. 10.15). The GGVF snake fails due to the strong gradient vector force caused by the inner object boundary. Practical examples of this can also be observed in most of the real images shown later, such as Figs 10.20 and 10.26.

10.9.3 Testing on Noisy Images

We also performed comparative tests to examine and quantify the tolerance to noise for the standard geometric, the geometric GGVF, and the RAGS snakes. For this a harmonic shape was used as shown in Fig. 10.18. It was generated



Figure 10.17: Strong neighboring edge leakage. From left: initial snake, geodesic snake steps through weak edge in top right of outer boundary, GGVF is attracted by the stronger inner edge, and RAGS snake converges properly using extra region force.



Figure 10.18: A shape and its boundary (a harmonic curve).

using

$$r = a + b \cos(m\theta + c), \quad (10.41)$$

where r is the length from any edge point to the center of the shape, a , b , and c remain constant, and m can be used to produce different numbers of ‘bumps’; in this case $m = 6$. We added varying amounts of noise and measured the accuracy of fit (i.e. boundary description) after convergence. The accuracy was computed using maximum radial error (MRE), i.e. the maximum distance in the radial direction between the true boundary and each active contour.

Impulse noise was added to the original image from 10% to 60% as shown in the first column of Fig. 10.19. The region segmentation data used for RAGS is in the second column (without any post-processing to close gaps, etc.). The third, fourth, and fifth columns show the converged snake for the standard geometric, the GGVF, and RAGS snakes respectively. A simple subjective examination clearly demonstrates the superior segmentation quality of the proposed snake. The initial state for the standard geometric and RAGS snakes is a square at the edge of the image, while for the GGVF it is set close to the true boundary to ensure better convergence. At low percentages of noise, all snakes could find the boundary accurately enough. However, at increasing noise levels ($>20\%$), more and more local maxima appear in the gradient flow force field, which prevent the standard geometric and GGVF snakes from converging to the true boundaries. The RAGS snake has a global view of the noisy image and the underlying region force pushes it toward the boundary. The MRE results are shown in Table 10.1. These verify RAGS error values to be consistently and significantly lower than the other two snake types for noise levels $>10\%$.

10.9.4 Results on gray level images

Figures 10.20–10.22 demonstrate RAGS in comparison to the standard geometric and GGVF snakes on various gray level images. Figure (10.20) shows a good example of weak-edge leakage on the lower side of the object of interest. While

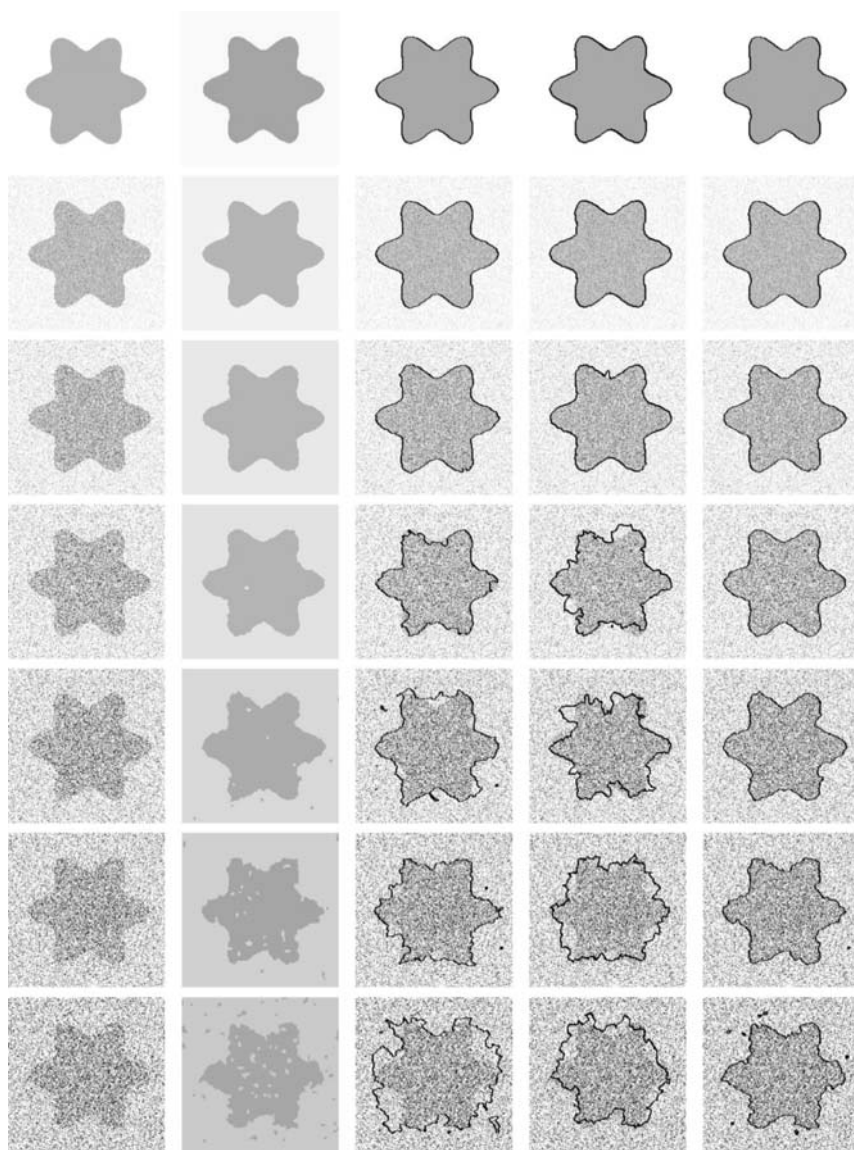


Figure 10.19: Shape recovery in noisy images. (Column 1) original image with various levels of added Gaussian noise [0%, 10%, ..., 60%], (column 2) the region maps later diffused by RAGS, (column 3) standard geometric snake results, (column 4) GGVF snake results, and (column 5) RAGS results.

Table 10.1: MRE comparison for the harmonic shapes in Fig. 10.19

% noise	Standard geometric snake error	GGVF snake error	RAGS snake error
0	2.00	2.00	2.00
10	2.23	2.24	2.00
20	5.00	7.07	4.03
30	10.00	16.03	3.41
40	16.16	21.31	5.22
50	15.81	21.00	5.38
60	28.17	20.10	5.83

RAGS does extremely well here, the geometric snake leaks through and the GGVF snake leaks and fails to progress at all in the narrow object. In Fig. 10.21, RAGS achieves a much better overall fit than the other snakes, particularly in the lower regions of the right-hand snake and the upper-right regions of the left-hand snake. In Fig. 10.22, again RAGS manages to segment the desired region much better than the standard geometric and the GGVF snakes. Note the stan-

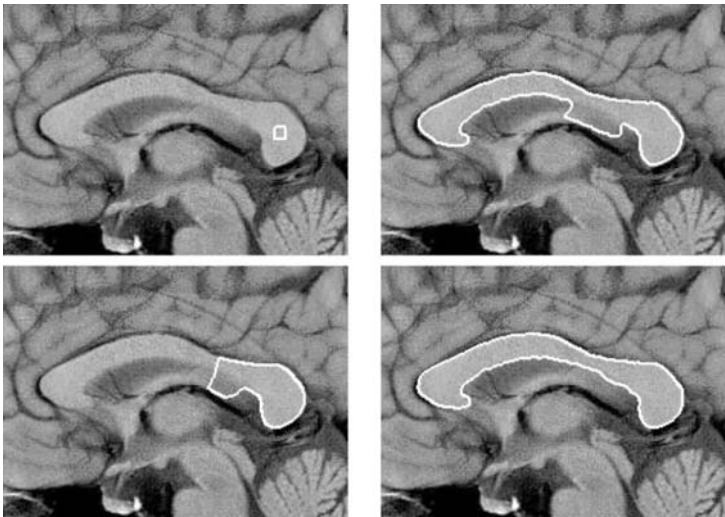


Figure 10.20: Brain MRI (*corpus callosum*) image. Top row: initial snake, standard geometric snake. Bottom row: GGVF snake and RAGS snake (original image courtesy of GE Medical Systems).

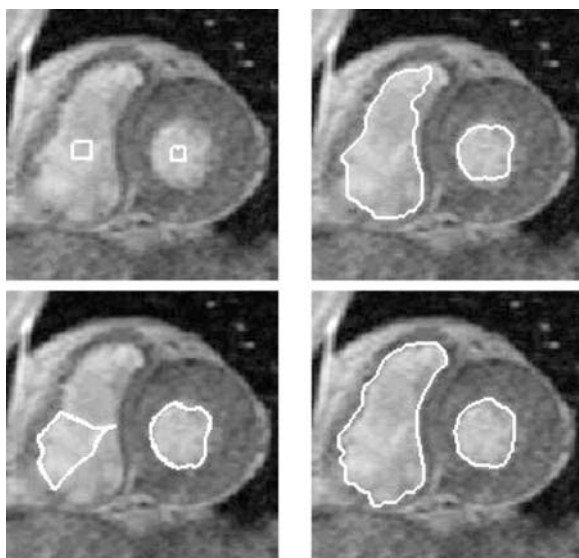


Figure 10.21: Heart MRI image. Top row: initial snakes and standard geometric snakes. Bottom row: GGVF snakes and final RAGS snakes showing improvement on the top right of the left snake and the lower region of the right snake.

standard snake leaks out of the object, similar to the effect demonstrated with the synthetic image in Fig. 10.15.

10.9.5 Results on Color Images

We now consider the performance of the RAGS snake on color images. In Fig. 10.23 we can see a cell image with both strong and fuzzy region boundaries. Note how the fuzzy boundaries to the right of the cell “dilute” gradually into the background. So the results in the top-right image again demonstrate an example of weak-edge leakage, similar to the example in Fig. 10.22, where the standard geometric snake fails to converge on the outer boundary. The middle and bottom rows show the converged RAGS snake using the oversegmentation and undersegmentation color region maps produced by the mean shift algorithm.

A very similar example is demonstrated in Fig. 10.24 in application to images of the optic disk in which the blood vessels have been removed using color mathematical morphology techniques. Again, the failing performance of the standard

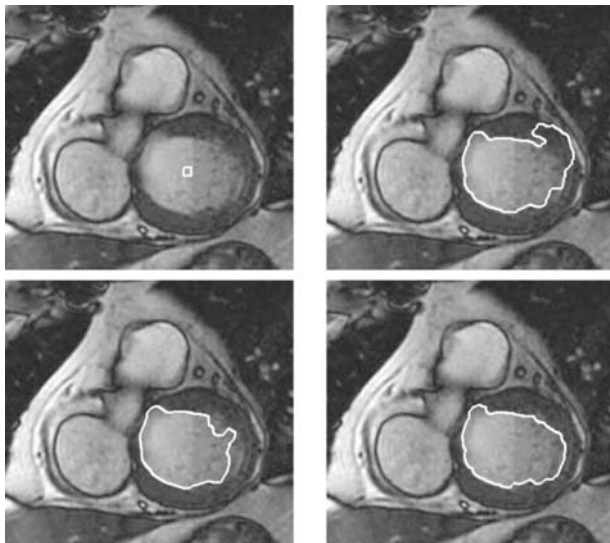


Figure 10.22: Heart MRI image. Top row: initial snake, and standard geometric snake. Bottom row: GGVF snake and final RAGS snake showing better convergence and no leakage (original image courtesy of GE Medical Systems).

snake is shown along with the RAGS results on both oversegmentation and undersegmentation regions.

In Fig. 10.25, a full application of RAGS is presented where the resulting regions from the RAGS snake are quantitatively evaluated against those hand-labeled by an expert ophthalmologist. The first column represents these groundtruth boundaries. The second column shows the position of the starting RAGS snakes. The boundary of the optic disk is quite fuzzy and well blended with the background. The region force helps the proposed snake stop at weak edges while the standard geometric snake leaks through (as shown in Fig. 10.24) and the accuracy of the GGVF snake is highly dependent on where the initial snake is placed (hence GGVF snake results are not provided). The last two columns illustrate the RAGS results using oversegmented and undersegmented regions of the mean shift algorithm respectively.

A simple measure of overlap is used to evaluate the performance of the RAGS snake against its corresponding groundtruth:

$$M = \frac{n(A \cap B)}{n(A \cup B)}$$

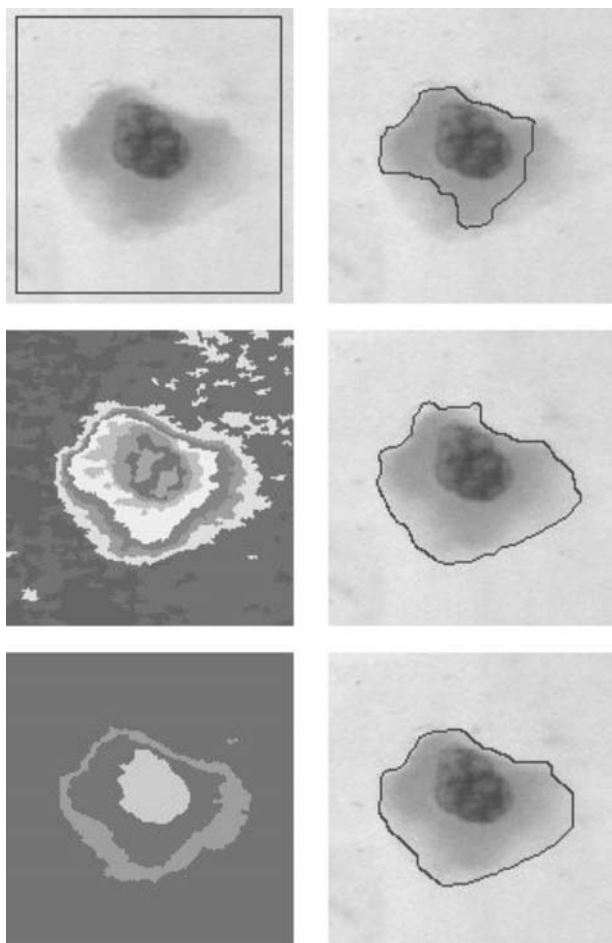


Figure 10.23: Weak-edge leakage testing. Top row: original image with starting contour and geodesic snake which steps through. Middle row: oversegmentation color region map and converged RAGS snake. Bottom row: undersegmentation color region map and converged RAGS snake (original image courtesy of Bristol Biomedical Image Archive, Bristol University, UK) (color slide).

where A and B correspond to ground-truth and RAGS localized optic disk regions respectively, and $n(\cdot)$ is the number of pixels in a region. Table 10.2 shows the result of measurement M demonstrating a 91.7% average performances for both over/undersegmentation RAGS respectively.

The final example in Fig. 10.26 shows a darker cell center compared to the cell outer region, but more significantly the object of interest is surrounded by

Table 10.2: Quantitative evaluation of RAGS snake on the optic disks in 10.25

Image	1	2	3	4	5	6	Average
% RAGS (over)	91.4	90.0	91.9	93.1	93.1	90.5	91.7
% RAGS (under)	90.7	89.5	93.1	91.3	93.0	92.7	91.7

other strong features. The standard geometric snake splits and converges unsatisfactorily and the GGVF snake is pulled in and out by the stronger inner cell nucleus and neighboring cells respectively, while the RAGS snake converges well to the outer cell boundary without leaking through.

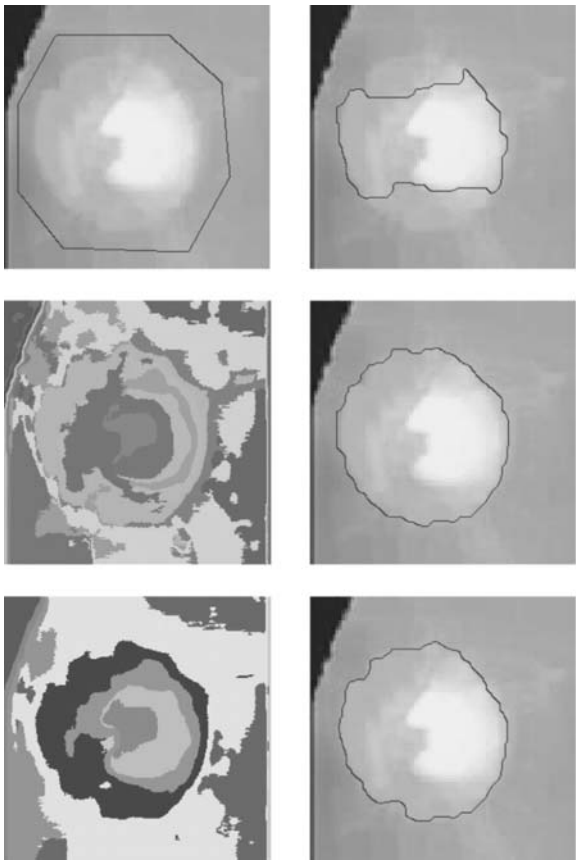


Figure 10.24: Optic disk localization. Top row: initial contour and geodesic snake which steps through to the stronger central region. Middle row: oversegmentation color region map and final RAGS snake. Bottom row: undersegmentation color region map and final RAGS snake (color slide).

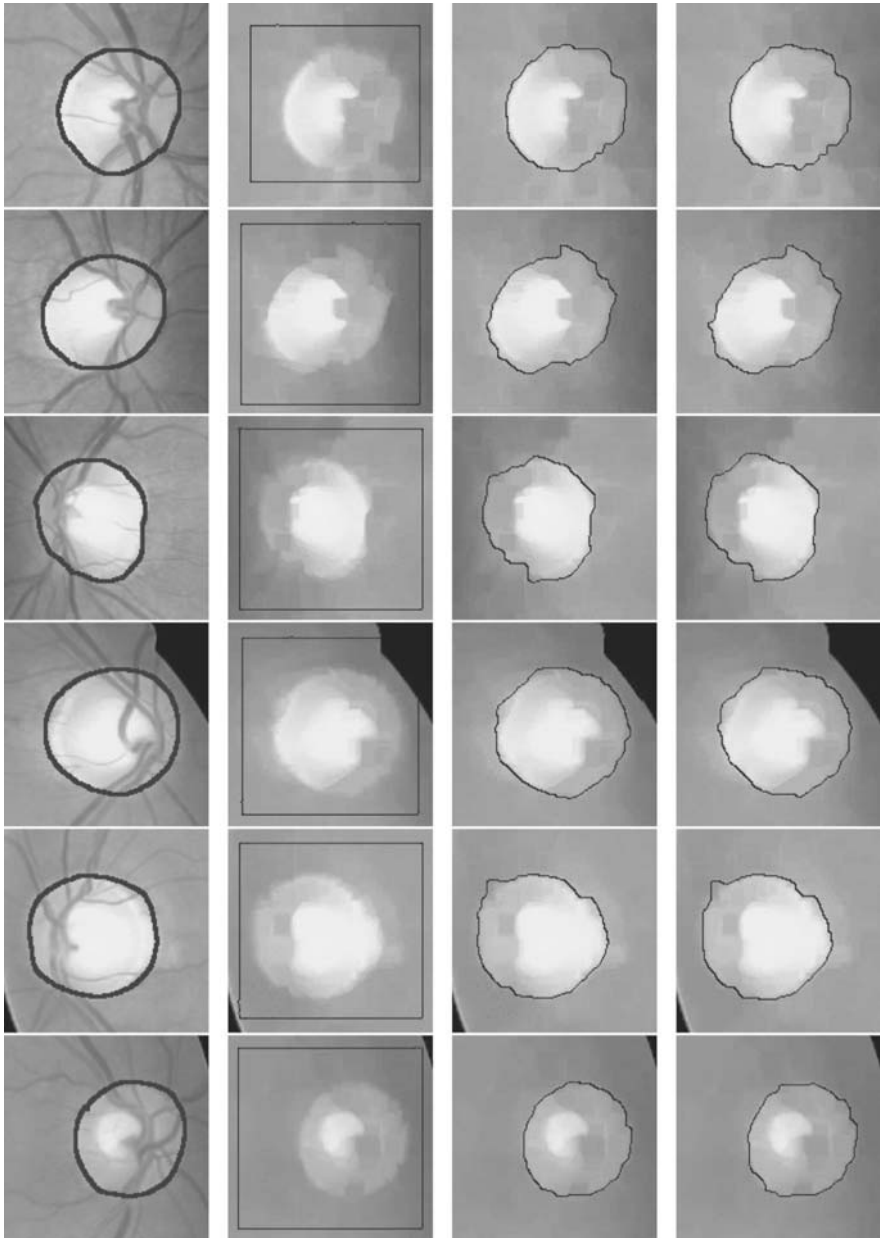


Figure 10.25: RAGS segmentation comparison with ground-truth. (Column 1) ground-truth, (column 2) initial snakes, (column 3) RAGS results with over-segmentation, and (column 4) RAGS results with undersegmentation (color slide).

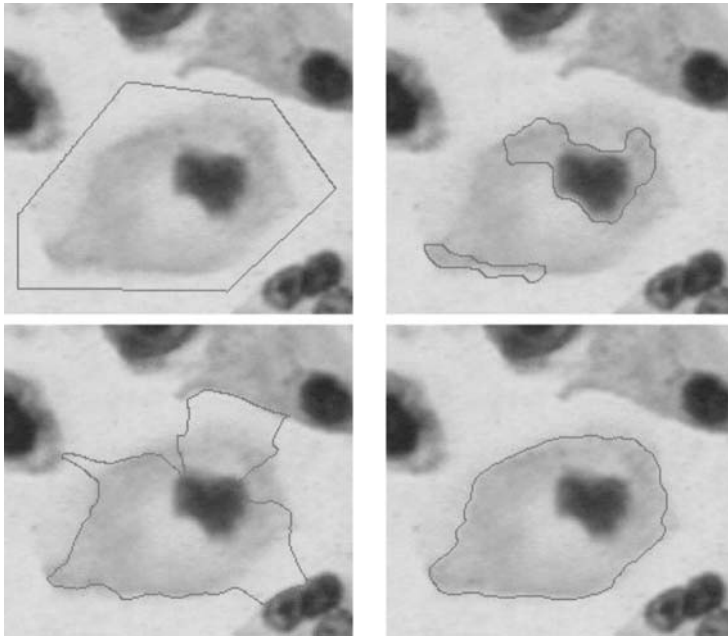


Figure 10.26: Cell with strong nucleus feature. Top row: initial snake and standard geometric snake. Bottom row: GGVF snake and RAGS snake showing how the stronger inner edge in the cell nucleus does not cause it to lose the outer weaker edge (original image courtesy of Bristol Biomedical Image Archive, Bristol University, UK) (color slide).

All the examples shown here illustrate the resilience of RAGS to weak edges and noise. However, the RAGS snake does suffer from some shortcomings. As with the standard geometric snake, or the geometric GGVF snake, it will not perform well in highly textured regions in which the gradient flow forces may be hampered by multitudes of texture edge information. It is also dependent on a reasonable segmentation stage, although this was shown to be quite flexible using a popular method of image segmentation.

10.10 Conclusions

A novel method, the region-aided geometric snake or RAGS, has been proposed. It integrates the gradient flow forces with region constraints, composed of the image region vector flow forces obtained through the diffusion of the region

map. The theory behind RAGS is standalone and hence the region force can be generated starting from any reasonable segmentation technique. We also showed its simple extension to color gradients. We demonstrated the performance of RAGS, against the standard geometric snake and the geometric GGVF snake, on weak edges and noisy images as well as on a number of other examples.

The experimental results have shown that the region-aided snake is much more robust toward weak edges. Also, it has better convergence quality compared with both the standard geometric snake and the geometric GGVF snake. The weak-edge leakage problem is usually caused by inconclusive edge values at the boundaries, which makes it difficult for gradient-based techniques to define a good edge. The gradual changes do not provide sufficient minima for the stopping function to prevent the level set accumulating in that area. The diffused region map gives the snake an extra underlying force at the boundaries. It also makes the snake more tolerable to noise as shown by the harmonic shape recovery experiment and many of the real images. The noise in the image introduces local minima in the stopping function preventing the standard geometric snake to converge to the true boundary. However, for RAGS the diffused region forces give a better global idea of the object boundary in the noise clutter and help the snake step closer and converge to the global minima.

10.11 Further Reading

Deformable contour models are commonly used in image processing and computer vision, for example for shape description [21], object localization [22], and visual tracking [23].

A good starting point to learn about *parametric* active contours is [24]. These snakes have undergone significant improvements since their conception, for example see the GVF snake in [7,9]. Region-based parametric snake frameworks have also been reported in [25–27]

The geometric model of active contours was simultaneously proposed by Caselles *et al.* [1] and Malladi *et al.* [2]. Geometric snakes are based on the theory of curve evolution in time according to intrinsic geometric measures of the image. They are numerically implemented via level sets, the theory of which can be sought in [15, 16].

There has been a number of works based on the geometric snake and level set framework. Siddiqi *et al.* [14] augmented the performance of the standard geometric snake that minimizes a modified length functional by combining it with a weighted area functional. Xu *et al.* extended their parametric GVF snake [7] into the generalized GVF snake, the GGVF, in [9]. Later, they also established an equivalence model between parametric and geometric active contours [10] using the GGVF. A geometric GGVF snake enhanced with simple region-based information was presented in [10]. Paragios *et al.* [28, 29] presented a boundary and region unifying geometric snake framework which integrates a region segmentation technique with the geometric snake. In [30], Yezzi *et al.* developed coupled curve evolution equations and combined them with image statistics for images of a known number of region types, with every pixel contributing to the statistics of the regions inside and outside an evolving curve. Using color edge gradients, Sapiro [6] extended the standard geometric snake for use with color images (also see Fig. 10.6). In [11], Chan *et al.* described a region-segmentation-based active contour that does not use the geometric snake's gradient flow to halt the curve at object boundaries. Instead, this was modeled as an energy minimization of a Mumford–Shah-based minimal partition problem and implemented via level sets. Their use of a segmented region map is similar to the concept we have explored here.

Level set methods can be computationally expensive. A number of fast implementations for geometric snakes have been proposed. The *narrow band* technique, initially proposed by Chop [31], only deals with pixels that are close to the evolving zero level set to save computation. Later, Adalsterinsson *et al.* [32] analyzed and optimized this approach. Sethian [33, 34] also proposed the fast marching method to reduce the computations, but it requires the contours to monotonically shrink or expand. Some effort has been expended in combining these two methods. In [35], Paragios *et al.* showed this combination could be efficient in application to motion tracking. Adaptive mesh techniques [36] can also be used to speed up the convergence of PDEs. More recently, additive operative splitting (AOS) schemes were introduced by Weickert *et al.* [37] as an unconditionally stable numerical scheme for nonlinear diffusion in image processing. The basic idea is to decompose a multidimensional problem into one-dimensional ones. AOS schemes can be easily applied in implementing level set propagation [38].

The mean shift algorithm is a nonparametric technique for estimation of the density gradient, which was first proposed by Fukunaga *et al.* [39]. The idea was later generalized by Cheng [40]. The technique was extended to various applications, amongst them color image segmentation, by Comaniciu *et al.* [12, 13, 20].

10.12 Appendix. Numerical Solution for the Level Set Implementation of RAGS

Let $\phi_{i,j}$ denote the value of ϕ at the grid position of $x_i = i\Delta x$, $y_i = j\Delta y$, where Δx and Δy are the grid steps along x and y directions respectively. Denote $\phi(x_i, y_j, t_n)$ by $\phi_{i,j}^n$, the time derivative ϕ_t at (i, j, t_n) is approximated by forward difference as $\phi_t(i, j, t_n) = (\phi_{i,j}^{n+1} - \phi_{i,j}^n)/\Delta t$, where Δt is a small time interval. As given in (10.17), the snake evolves according to four forces. However, they can be categorized into three types based on the nature of their motions.

The first motion is a collapsing one with speed proportional to its curvature. It is a parabolic contribution to the equation of motion and it can be approximated with central differences. The curvature κ is only dependent on contours; it is independent of time and spatial position, hence it can also be solved using central difference approximations. The curvature motion at time t is approximated as

$$(g(\cdot)\kappa|\nabla\phi|)_{i,j}^n = g(\cdot)_{i,j}K_{i,j}^n(D_{i,j}^{0x2} + D_{i,j}^{0y2})^{1/2}, \quad (10.42)$$

where $D_{i,j}^{0x} = (\phi_{i+1,j}^n - \phi_{i-1,j}^n)/2\Delta x$, $D_{i,j}^{0y} = (\phi_{i,j+1}^n - \phi_{i,j-1}^n)/2\Delta y$, and $K_{i,j}^n$ is the central difference approximation to the curvature expression given in (10.22):

$$(\phi_x)_{i,j}^n = D_{i,j}^{0x}, (\phi_y)_{i,j}^n = D_{i,j}^{0y}, \quad (10.43)$$

$$(\phi_{xx})_{i,j}^n = \frac{\phi_{i+1,j}^n - 2\phi_{i,j}^n + \phi_{i-1,j}^n}{\Delta x^2}, \quad (\phi_{yy})_{i,j}^n = \frac{\phi_{i,j+1}^n - 2\phi_{i,j}^n + \phi_{i,j-1}^n}{\Delta y^2}, \quad (10.44)$$

$$(\phi_{xy})_{i,j}^n = \frac{\phi_{i+1,j+1}^n - \phi_{i-1,j+1}^n - \phi_{i+1,j-1}^n + \phi_{i-1,j-1}^n}{4\Delta x\Delta y}. \quad (10.45)$$

The second motion is expanding or shrinking with a spatially constant speed in its normal direction. It must be approximated through *entropy-satisfying schemes* [16]. Let V_0 be the constant speed function regarding $\alpha g(\cdot)$. Following

Sethian's *upwinding* finite difference scheme, the solution is given by

$$\begin{cases} (V_0|\phi|)_{i,j}^n = V_{0i,j}[\max(D_{i,j}^{-x}, 0)^2 + \min(D_{i,j}^{+x}, 0)^2 \\ \quad + \max(D_{i,j}^{-y}, 0)^2 + \min(D_{i,j}^{+y}, 0)^2]^{1/2} & \text{if } V_{0i,j} \geq 0 \\ (V_0|\phi|)_{i,j}^n = V_{0i,j}[\max(D_{i,j}^{+x}, 0)^2 + \min(D_{i,j}^{-x}, 0)^2 \\ \quad + \max(D_{i,j}^{+y}, 0)^2 + \min(D_{i,j}^{-y}, 0)^2]^{1/2} & \text{otherwise} \end{cases}, \quad (10.46)$$

where $D_{i,j}^{+x} = (\phi_{i+1,j}^n - \phi_{i,j}^n)/\Delta x$, $D_{i,j}^{+y} = (\phi_{i,j+1}^n - \phi_{i,j}^n)/\Delta y$ and $D_{i,j}^{-x} = (\phi_{i,j}^n - \phi_{i-1,j}^n)/\Delta x$, $D_{i,j}^{-y} = (\phi_{i,j}^n - \phi_{i,j-1}^n)/\Delta y$ are the forward and backward differences, respectively.

The external forces left in (10.17) contribute the third underlying static velocity field for snake evolution. Their direction and strength are based on spatial position, but not on the snake. This motion can be numerically approximated as follows. Let $\vec{U}(x, y, t)$ denote the underlying static velocity field according to $\beta \vec{R} - \nabla g(\cdot)$. We check the sign of each component of $\vec{U}$ and construct one-sided upwind differences in the appropriate (upwind) direction [16]:

$$\begin{aligned} (\vec{U} \cdot \nabla \phi)_{i,j}^n &= \max(u_{i,j}^n, 0)D_{i,j}^{-x} + \min(u_{i,j}^n, 0)D_{i,j}^{+x} \\ &\quad + \max(v_{i,j}^n, 0)D_{i,j}^{-y} + \min(v_{i,j}^n, 0)D_{i,j}^{+y}, \end{aligned} \quad (10.47)$$

where $\vec{U} = (u, v)$. Thus, (10.17) is numerically solved using the schemes described above.

Questions

1. What are the advantages of geometric snakes over their parametric counterparts?
2. Which are some of the key papers on the geometric snake?
3. How do I diffuse the region segmentation map?
4. Describe how weighting functions $p(\cdot)$ and $q(\cdot)$ behave in vector diffusion?
5. What are the parameters in RAGS?
6. How do I choose the parameter values?
7. What are some of the disadvantages of RAGS?

8. *What is a good source of information for learning about level sets?*
9. *How are level sets initialized?*
10. *Is the geometric snake computationally efficient? Are there any ways to speed up the convergence?*
11. *How do I find out more about the GVF and GGVF snakes?*
12. *Describe the mean shift process.*
13. *How do I find out more about the mean shift segmentation method?*
14. *Who else has applied di Zenzo's method of vector gradients in the active contour literature?*
15. *How do I find out more about the optic disk application from the Results section?*

Bibliography

- [1] Caselles, V., Catte, F., Coll, T., and Dibos, F., A geometric model for active contours, *Numer. Math.*, Vol. 66, pp. 1–31, 1993.
- [2] Malladi, R., Sethian, J., and Vemuri, B., Evolutionary fronts for topology independent shape modeling and recovery, In: *Proceedings of the 3rd European Conference on Computer Vision*, pp. 3–13, 1994.
- [3] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., Gradient flows and geometric active contour models, In: *Proceedings of the 5th IEEE International Conference on Computer Vision*, pp. 810–815, 1995.
- [4] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic active contour, *International J. Comput. Vis.*, Vol. 22, No. 1, pp. 61–79, 1997.
- [5] Malladi, R., Sethian, J., and Vemuri, B., Shape modeling with front propagation: A level set approach, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 2, pp. 158–175, 1995.
- [6] Sapiro, G., Color Snakes, *Comput. Vis. Image Underst.*, Vol. 68, No. 2, pp. 247–253, 1997.
- [7] Xu, C. and Prince, J., Snakes, Shapes, and Gradient Vector Flow, *IEEE Trans. Image Process.*, Vol. 7, No. 3, pp. 359–369, 1998.
- [8] Xu, C. and Prince, J., Gradient vector flow: A new external force for snakes, In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pp. 66–71, 1997.
- [9] Xu, C. and Prince, J., generalized gradient vector flow external forces for active contours, *Signal Process.*, Vol. 71, No. 2, pp. 131–139, 1998.
- [10] Xu, C., Yezzi, J., and Prince, J., On the relationship between parametric and geometric active contours, In: *Proceedings of the 34th Asilomar Conference on Signal, Systems, and Computers*, pp. 483–489, 2000.

- [11] Chan, T. and Vese, L., Active contours without edges, *IEEE Trans. Image Process.*, Vol. 10, No. 2, pp. 266–277, 2001.
- [12] Comaniciu, D. and Meer, P., Mean shift analysis and applications, In: *Proceedings of the 7th IEEE International Conference on Computer Vision*, pp. 1197–1203, 1999.
- [13] Comaniciu, D. and Meer, P., Mean shift: A robust approach toward feature space analysis, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 24, No. 5, pp. 603–619, 2002.
- [14] Siddiqi, K., Lauziere, Y., Tannenbaum, A., and Zucker, S., Area and length minimizing flows for shape segmentation, *IEEE Trans. Image Process.*, Vol. 7, No. 3, pp. 433–443, 1998.
- [15] Osher, S. and Sethian, J., Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.*, Vol. 79, pp. 12–49, 1988.
- [16] Sethian, J., *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, 1996.
- [17] Sethian, J., Curvature and the evolution of fronts, *Commun. Math. Phys.*, Vol. 101, pp. 487–499, 1985.
- [18] Osher, S. and Fedkiw, R., *Level Sets and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2002.
- [19] di Zenzo, S., A note on the gradient of a multi-image, *Comput. Vis., Graph. Image Process.*, Vol. 33, No. 1, pp. 116–125, 1986.
- [20] Comaniciu, D. and Meer, P., Robust analysis of feature spaces: Color image segmentation, In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pp. 750–755, 1997.
- [21] Cootes, T., Taylor, C., Cooper, D., and Graham, J., Active shape models—their training and application, *Comput. Vis. Image Underst.*, Vol. 61, No. 1, pp. 38–59, 1995.

- [22] Osareh, A., Mirmehdi, M., Thomas, B., and Markham, R., Colour morphology and snakes for optic disc localisation, In: *Proceedings of the 6th Conference on Medical Image Understanding and Analysis*, pp. 21–24, 2002.
- [23] Blake, A. and Isard, M., *Active Contours*, Springer, London, 1998.
- [24] Kass, M., Witkin, A., and Terzopoulos, D., Snakes: Active contour models, *Int. J. Comput. Vis.*, Vol. 1, pp. 321–331, 1988.
- [25] Ronfard, R., Region-based strategies for active contour models, *int. J. Comput. Vis.*, Vol. 13, No. 2, pp. 229–251, 1994.
- [26] Chakraborty, A., Staib, L., and Duncan, J., Deformable boundary finding in medical images by integrating gradient and region information, *IEEE Trans. Med. Imaging*, Vol. 15, No. 6, pp. 859–870, 1996.
- [27] Zhu, S. and Yuille, A., Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 18, No. 9, pp. 884–900, 1996.
- [28] Paragios, N. and Deriche, R., Coupled geodesic active regions for image segmentation: A level set approach, In: *Proceedings of the 6th European Conference on Computer Vision*, pp. 224–240, 2000.
- [29] Paragios, N. and Deriche, R., Geodesic active regions: A new framework to deal with frame partition problems in computer vision, *J. Vis. Commun. Image Represent.*, Vol. 13, No. 1–2, pp. 249–268, 2002.
- [30] Yezzi, A., Tsai, A., and Willsky, A., A fully global approach to image segmentation via coupled curve evolution equations, *J. Vis. Commun. Image Represent.*, Vol. 13, No. 1–2, pp. 195–216, 2002.
- [31] Chop, D., Computing minimal surfaces via level set curvature-flow, *J. Comput. Phys.*, Vol. 106, pp. 77–91, 1993.
- [32] Adalsterinsson, D. and Sethian, J., A fast level set method for propagating interfaces, *J. Comput. Phys.*, Vol. 118, pp. 269–277, 1995.

- [33] Sethian, J., Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces, *Acta Numer.*, Vol. 5, pp. 309–395, 1996.
- [34] Sethian, J., A fast marching level set method for monotonically advancing fronts, In: *Proceedings of the National Academy of Sciences*, Vol. 93, pp. 1591–1694, 1996.
- [35] Paragios, N. and Deriche, R., Geodesic active contour and level set for the detection and tracking of moving objects, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 22, No. 3, pp. 266–280, 2000.
- [36] Milne, R., An adaptive level-set method, Ph.D. Thesis, Department of Mathematics, University of California, Berkeley, 1995.
- [37] Weickert, J., ter Harr Romeny, B. M., and Viergener, M., Efficient and reliable scheme for non-linear diffusion and filtering, *IEEE Trans. Image Process.*, Vol. 7, pp. 398–410, 1998.
- [38] Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M., Fast geodesic active contours, *IEEE Trans. Image Process.*, Vol. 10, No. 10, pp. 1467–1475, 2001.
- [39] Fukunaga, K. and Hostetler, L., The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Trans. Inf. Theory*, Vol. IT-21, pp. 32–40, 1975.
- [40] Cheng, Y., Mean shift, mode seeking and clustering, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 8, pp. 790–799, 1995.
- [41] Danielsson, P., Euclidean distance mapping, *Comput. Graph. Image Process.*, Vol. 14, pp. 227–248, 1980.
- [42] Borgefors, G., Distance transformations in arbitrary dimensions, *Comput. Vis., Graph. Image Process.*, Vol. 27, pp. 321–345, 1984.
- [43] Eggers, H., Two fast Euclidean distance transformations in Z^2 based on sufficient propagation, *Comput. Vis. Image Underst.*, Vol. 69, No. 1, pp. 106–116, 1998.

- [44] Gevers, T., Ghebreab, S., and Smeulders, A., Color invariant snakes, In: Proceedings of the 9th British Machine Vision Conference, pp. 659–670, 1998.
- [45] Osareh, A., Mirmehdi, M., Thomas, B., and Markham, R., Identification of exudate pathologies and the optic disc in colour retinal images, Br. J. Ophthalmol., Vol. 87, pp. 1220–1223, 2003.

Chapter 11

Co-Volume Level Set Method in Subjective Surface Based Medical Image Segmentation

Karol Mikula,¹ Alessandro Sarti,² and Fiorella Sgallari³

11.1 Introduction

It is well known that the so-called *level set equation* [42, 43, 54, 55]

$$u_t = |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \quad (11.1)$$

for curvature-driven motion as well as its nontrivial generalizations are well suited to image processing applications and they are often used nowadays. In this chapter we deal with a specific equation of mean curvature flow type [48–50], namely,

$$u_t = \sqrt{\varepsilon^2 + |\nabla u|^2} \nabla \cdot \left(g(|\nabla G_\sigma * I^0|) \frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}} \right), \quad (11.2)$$

where $u(t, x)$ is an unknown (segmentation) function defined in $Q_T \equiv [0, T] \times \Omega$. $\Omega \subset \mathbb{R}^d$ is a bounded domain with a Lipschitz continuous boundary $\partial\Omega$, $[0, T]$ is a time interval, I^0 is a given image, and $\varepsilon > 0$ is a parameter. The equation is accompanied with zero Dirichlet boundary conditions and initial

¹Department of Mathematics, Slovak University of Technology, Radlinského 11, 813 68 Bratislava, Slovakia, E-mail: mikula@vox.svf.stuba.sk

²DEIS, University of Bologna, Via Risorgimento 2, 40136 Bologna, Italy, E-mail: asarti@deis.unibo.it

³Department of Mathematics, University of Bologna, Piazza di Porta S. Donato 5, 40127 Bologna, Italy, E-mail: sgallari@dm.unibo.it

condition

$$u(t, x) = u^D \quad \text{in } [0, T] \times \partial\Omega, \quad (11.3)$$

$$u(0, x) = u^0(x) \quad \text{in } \Omega. \quad (11.4)$$

Without loss of generality, we may assume $u^D = 0$. The Perona–Malik function $g : \mathbb{R}_0^+ \rightarrow \mathbb{R}^+$ is nonincreasing, $g(0) = 1$, admitting $g(s) \rightarrow 0$ for $s \rightarrow \infty$ [45]. Usually we use the function $g(s) = 1/(1 + Ks^2)$, $K \geq 0$. $G_\sigma \in C^\infty(\mathbb{R}^d)$ is a smoothing kernel, e.g. the Gauss function

$$G_\sigma(x) = \frac{1}{(4\pi\sigma)^{d/2}} e^{-|x|^2/4\sigma}, \quad (11.5)$$

which is used in presmoothing of image gradients by the convolution

$$\nabla G_\sigma * I^0 = \int_{\mathbb{R}^d} \nabla G_\sigma(x - \xi) \tilde{I}^0(\xi) d\xi, \quad (11.6)$$

with $\tilde{I}^0$ being the extension of I^0 to $\mathbb{R}^d$ given by periodic reflection through the boundary of image domain. The computational domain Ω is usually a subdomain of the image domain; it should include the segmented object. In fact, in most situations Ω corresponds to image domain itself. We assume that an initial state of the segmentation function is bounded, i.e. $u^0 \in L_\infty(\Omega)$. For shortening notations, we will use the abbreviation

$$g^0 = g(|\nabla G_\sigma * I^0|). \quad (11.7)$$

Due to smoothing properties of convolution, we always have $1 \geq g^0 \geq \nu_\sigma > 0$ [5, 27].

Equation (11.2) is a regularization, in the sense $|\nabla u| \approx |\nabla u|_\varepsilon = \sqrt{\varepsilon^2 + |\nabla u|^2}$ [19], of the segmentation equation suggested in [7–9, 30, 31], namely,

$$u_t = |\nabla u| \nabla \cdot \left(g^0 \frac{\nabla u}{|\nabla u|} \right). \quad (11.8)$$

However, while in [19] the ε -regularization was used just as a tool to prove the existence of a viscosity solution of the level set equation (see also [10, 12]), in our work ε is a modeling parameter. As we will see later, it can help in suitable denoising and completion of missing boundaries in images. Such regularization can be interpreted as a mean curvature flow of graphs with respect to a specific Riemann metric given by the image features [49].

The idea to use Riemannian mean curvature flow of graphs to compute the so-called subjective contours [29] originates in [48–50]. The subjective surfaces method, developed there, has been successfully used to complete missing boundaries of objects in digital 2D and 3D data sets and thus it is a powerful method for segmentation of highly noisy, e.g. medical, images. In this chapter we follow the same idea.

Initially, a “point-of-view” surface, given by an observer (user) chosen fixation point inside the image, is taken as u^0 (see e.g. Fig. 11.11 (top right)). Then this initial state of the segmentation function is evolved by Eq. (11.2), until the so-called subjective surface arises (see e.g. Fig. 11.11 (bottom) right or Fig. 11.14 (top row)). For small ε , the subjective surface closes gaps in image object boundaries and is stabilized, i.e. almost does not change by further evolution, so it is easy to stop the segmentation process. The idea to follow evolution of the graph of segmentation function [48–50] and not to follow evolution of a particular level set of u is new in comparison with other level set methods used in image segmentation (cf. [6–9, 30, 31, 36]). In standard level set approach, the redistancing [42, 55] is used to keep unit slope along the level set of interest (e.g. along segmentation curve). In such an approach the evolution of u itself is forgotten at every redistancing step. Such solution prevents steepening of u and one cannot obtain the subjective surfaces. In our computational method we do not impose any specific requirements (e.g., redistancing) to solution of the level set equation, the numerically computed segmentation function can naturally evolve to a “piecewise constant steady state” result of the segmentation process.

For numerical solution of the nonlinear diffusion equation (11.2), governing Riemannian mean curvature flow of graphs, we use semi-implicit complementary volume (called also co-volume or finite volume-element) method. Since (11.2) is regularization of (11.8), for the curvature driven level set flow (11.8) or for some other form of the level set equation (11.1), the method can be used as well (cf. [21, 25]).

For time discretization of nonlinear diffusion equations, there are basically three possibilities: implicit, semi-implicit, or explicit schemes. For spatial discretization usually finite difference, finite volume, or finite element method is used. The co-volume technique is a combination of finite element and finite volume methods. Implicit, i.e. nonlinear, time discretization, and co-volume techniques for solution of the level set equation were introduced in [56]. The efficient

co-volume level set method based on semi-implicit, i.e. linear, time discretization was given and studied in [25]. In [25], the method was applied to image smoothing nonlinear diffusion level set equation; here we apply the method to image segmentation and completion of missing boundaries.

Let us note that Eq. (11.8) can be rewritten into an advection–diffusion form as

$$u_t = g^0 |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \nabla g^0 \cdot \nabla u. \quad (11.9)$$

Various finite difference schemes [7–9, 30, 31, 48–50] are usually based on this form using upwinding in advection term and explicit time stepping. Our co-volume technique relies on discretization of the basic form (11.8), or more precisely on its regularization (11.2), and we use its integral (weak, variational) formulation. In such a way, the discretization scheme naturally respects a variational structure of the problem, it gives clear discrete form of local mass balance, and it naturally fulfills discrete minimum–maximum principle (L_∞ -stability). The semi-implicit discretization in time yields such stability property (i.e. no spurious oscillations appear in our solution) for any length of discrete time step. This is a main advantage in comparison with explicit time stepping, where the stability is often achieved only under severe time step restriction. Since in nonlinear diffusion problems (such as the level set equation), the coefficients depend on the solution itself and thus they must be recomputed in every discrete time update, an overall CPU time for explicit scheme can be tremendous. On the other hand, the implicit time stepping as in [56], although unconditionally stable, leads to solution of nonlinear systems in every discrete time update. For the level-set-like problems, there is no guarantee for convergence of a fast Newton solver, and fixed-point-like iterations are very slow [56]. From this point of view, the semi-implicit method seems to be optimal regarding stability and efficiency. In every time update we solve linear system of equations which can be done efficiently using, e.g., suitable preconditioned iterative linear solvers.

In Section 11.2 we discuss various curve evolution and level set models leading to segmentation Eqs. (11.8) and (11.2). In Section 11.3 we introduce our semi-implicit co-volume level set method for solving these equations and discuss some of its theoretical properties and implementation aspects. In Section 11.4 we discuss numerical experiments.

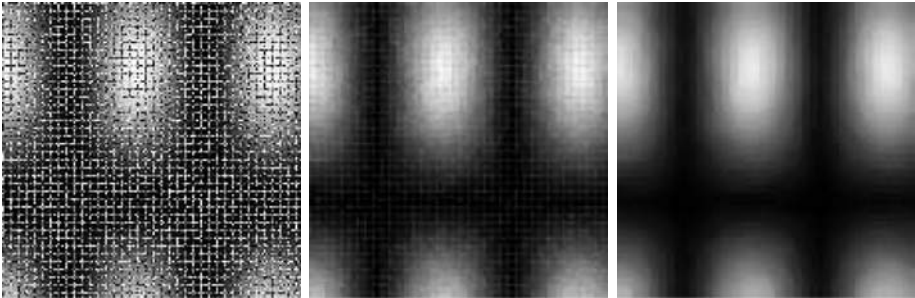


Figure 11.1: Image corrupted by a structural noise (left), and result of filtering by level set equation after two (middle) and ten (right) discrete scale steps.

11.2 Discussion on Related Mathematical Models

The level set equation (11.1) has great significance in axiomatization of image processing and computer vision [1]. It fulfills the so-called *morphological principle*: If u is a solution then, for any nondecreasing function φ , $\varphi(u)$ is a solution as well. It means that level sets of a solution u move independently of each other, or in other words, they diffuse only intrinsically (in tangential direction) and there is no diffusion across level sets in the normal direction. In that sense it provides a directional smoothing of the image along its level lines. We illustrate the smoothing effect of the level set equation in Figs. 11.1 (removing structural noise) and 11.2 (removing salt and pepper noise) [25].

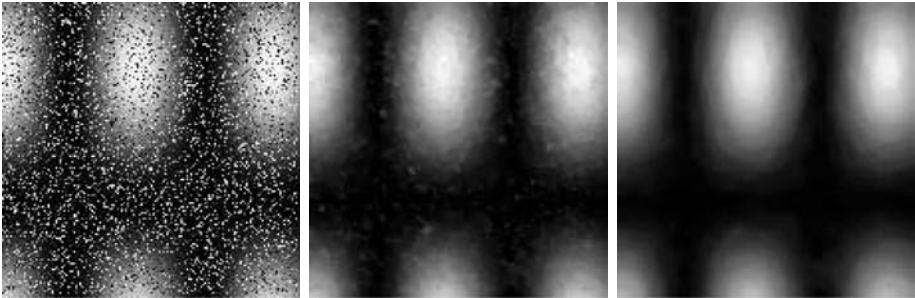


Figure 11.2: Initial image corrupted by salt and pepper noise (left), and result of filtering by level set equation after two (middle) and ten (right) discrete scale steps.

In image filtration, the initial condition for the level set equation (11.1) is given by the image gray-level intensity I^0 itself, i.e., $u^0 = I^0$ and usually zero Neumann boundary conditions are used. The solution $u(t, x)$ gives a family of *scaled* (filtered, smoothed) versions of $I^0(x)$. The parameter t is understood as *scale*, and the process of nonlinear selective smoothing is called image multiscale analysis [1]. In [25], the linear semi-implicit co-volume method to solve image selective smoothing equation [2]

$$u_t = g(|\nabla G_\sigma * u|)|\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \quad (11.10)$$

has been suggested and studied. Equation (11.10) can be used for edge-preserving smoothing in a similar way as the so-called Perona–Malik equation [1, 2, 5, 24–28, 37, 38, 41, 45], see Fig. 11.3.

The aim of segmentation is to find boundaries of a distinguished object of an image. In generic situation these boundaries correspond to edges. However, in

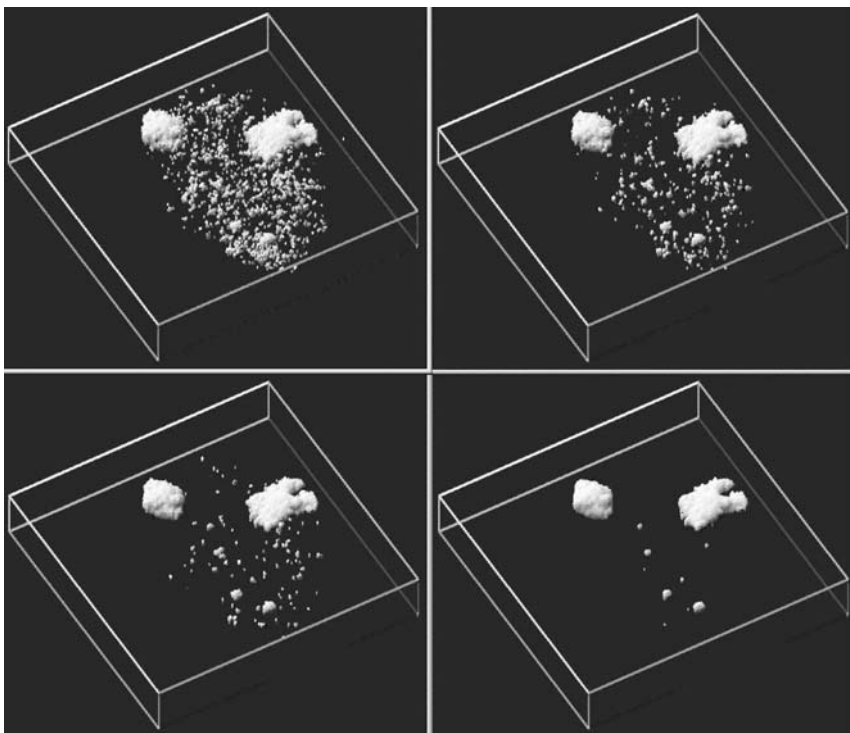


Figure 11.3: Extraction of two chromosomes in a human cell using geometrical diffusion (11.10) [24].

the presence of noise or in images with occlusions or subjective contours, these edges can be very irregular or even interrupted. Then the analysis of the scene and segmentation of objects become a difficult task.

In the so-called active contour models [32], an evolving family of curves converging to an edge is constructed. A simple approach (similar to various discrete region-growing algorithms) is to put small seed, e.g. small circular curve, inside the object and then evolve the curve to find automatically the object boundary. For such moving curves the level set models have been introduced in the last decade. A basic idea is that moving curve corresponds to a specific level line of the level set function which solves some reasonable generalization of Eq. (11.1). The level set methods have several advantages among which independence of dimension of the image and topology of objects are probably the most important. However, a reader can be interested also in the so-called direct (Lagrangian) approaches to curve and surface evolution (see e.g. [16–18, 39, 40]).

First simple level set model with the speed of segmentation curve modulated by $g(|\nabla I^0(x)|)$ (or more precisely by $g(|\nabla G_\sigma * I^0|)$), where g is a smooth edge detector function, e.g. $g(s) = 1/(1 + Ks^2)$, has been given in [6] and [36]. In such a model, “steady state” of a particular level set (level line in 2D image) corresponds to boundary of a segmented object. Due to the shape of the Perona–Malik function g , the moving segmentation curve is strongly slowed down in a neighborhood of an edge, leading to a segmentation result. However, if an edge is crossed during evolution (which is not a rare event in noisy images), there is no mechanism to go back. Moreover, if there is a missing part of the object boundary, the algorithm is completely useless (as any other simple region-growing method).

Later on, the curve evolution and the level set models for segmentation have been significantly improved by finding a proper driving force in the form $-\nabla g(|\nabla I^0(x)|)$ [7–9, 30, 31]. The vector field $-\nabla g(|\nabla I^0(x)|)$ has an important geometric property: It points toward regions where the norm of the gradient ∇I^0 is large (see Figs. 11.4 and 11.5). Thus if an initial curve belongs to a neighborhood of an edge, then it is driven toward this edge by this proper velocity field. Such motion can also be interpreted as a flow of the curve on surface $g(|\nabla I^0(x)|)$ subject to gravitational-like force driving the curve down to the narrow valley corresponding to the edge (see Fig. 11.6, [40]).

However, as one can see from Figs. 11.7 and 11.8, the situation is much more complicated in the case of noisy images. The advection process alone is

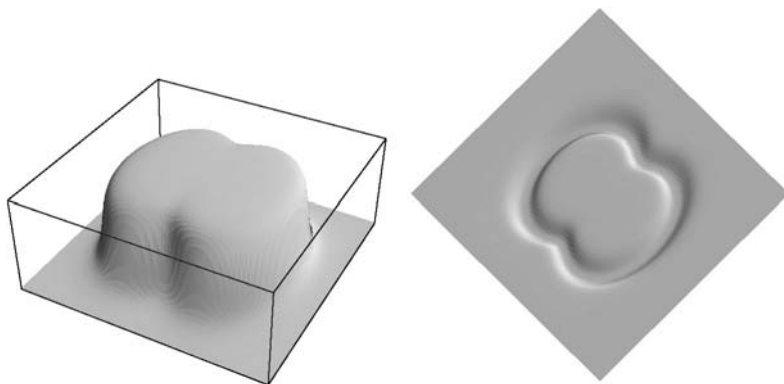


Figure 11.4: A graph of the image intensity function $I^0(x)$ corresponding to a “dumbbell” image (left, see also Fig. 11.5) and a graph of the function $g(|\nabla I^0(x)|)$ (right) where a narrow valley along the edge can be observed (color slide).

not sufficient. In a noisy environment, the evolving level set can behave very irregularly, it can be attracted to spurious edges and no reasonably convergent process can be observed. This phenomenon is documented in Fig. 11.8 left. To prevent such a situation, one has to regularize the evolution. A helpful regularization is to add a curvature dependence to the level set flow. If evolution

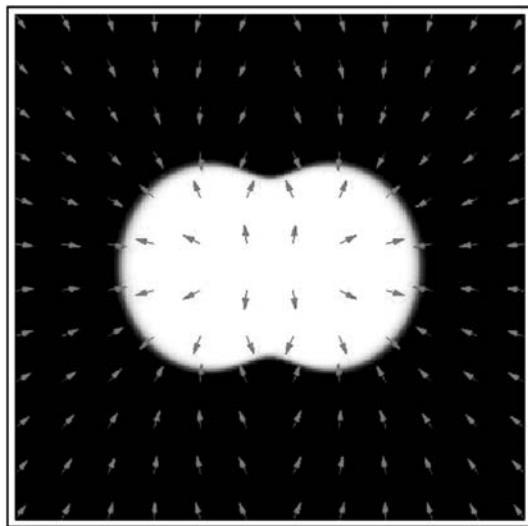


Figure 11.5: Image given by the intensity $I^0(x)$ from Fig. 11.4 (left) and the arrows representing the vector field $-\nabla g(|\nabla I^0(x)|)$ (color slide).



Figure 11.6: An initial ellipse driven by the vector field $-\nabla g(|\nabla I^0(x)|)$ down to the valley to find the edge in the image I^0 (color slide).

of a curve in the normal direction depends on its curvature k , then the sharp irregularities are smoothed. Such motion can be interpreted as an intrinsic diffusion of the curve. A reasonable regularization term is given by $g^0 k$, where the amount of curve intrinsic diffusion is small in the vicinity of unspurious edges. In Fig. 11.8 (right), we present initial ellipse evolution to successful segmentation result using such advection–(intrinsic) diffusion model, which was computed by the direct method from [40]. The level set formulation of such

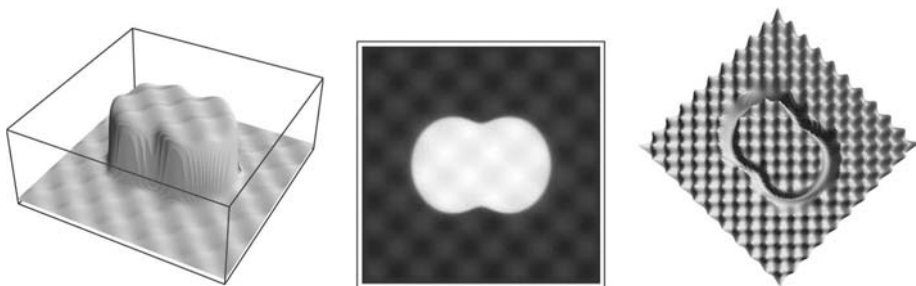


Figure 11.7: The situation is more complicated in the case of a “noisy” image (middle); we also plot a graph of its intensity $I^0(x)$ (left) and the corresponding surface $g(|\nabla I^0(x)|)$ (right) (color slide).

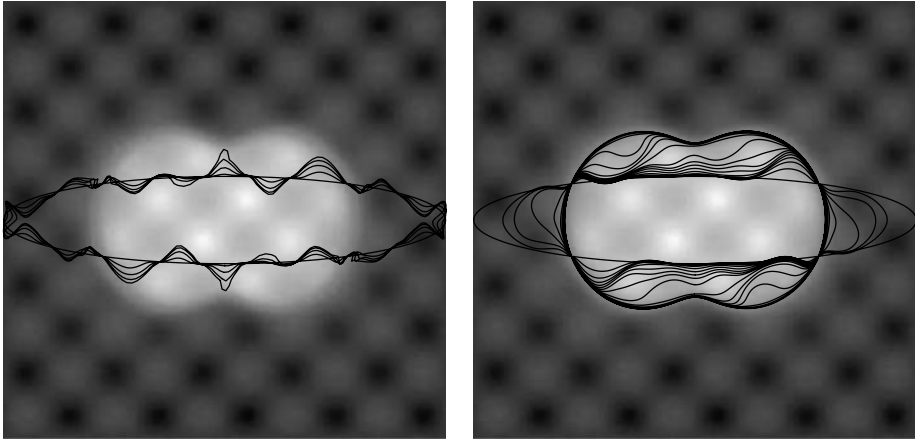


Figure 11.8: The evolution only by advection leads to attracting a curve (initial ellipse) to spurious edges, the evolution must be stopped without any reasonable segmentation result (left). By adding regularization term related to curvature of evolving curve, the edge is found smoothly (right).

curve evolution is given by Eq. (11.9), which is, of course, only another form of Eq. (11.8).

Although model (11.8) behaves very well if we are in the vicinity of an edge, it is sometimes difficult to drive the segmentation curve there. If we start with a small circular seed, it has large curvature and diffusion dominates advection so the seed disappears (curve shrinks to a point [22,23]). Then some constant speed must be added to dominate diffusion at the beginning of the process, but it is not clear at all when to switch off this driving force to have just the mechanism of the model (11.8). Moreover, in the case of missing boundaries of image objects, there is no criterion for such a switch, so the segmentation curve cannot be well localized to complete the missing boundaries.

An important observation now is that Eq. (11.8) moves not only one particular level line (segmentation curve) but all level lines by the above mentioned advection–diffusion mechanism. So, in spite of all previously mentioned segmentation approaches, we may start to think not on evolution of one particular level set but on evolution of the whole surface composed of those level sets. This idea to look on the solution u itself, i.e. on the behavior of our segmentation function, can help significantly.

Let us look on a simple numerical experiment presented in Fig. 11.10 representing extraction of the solid circle depicted in Fig. 11.9. The starting

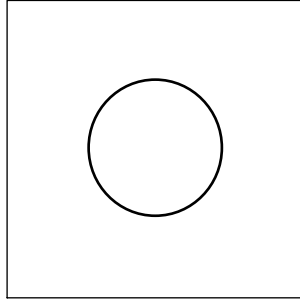


Figure 11.9: Image of a solid circle.

point-of-view surface u^0 is plotted on the top left. The subsequent evolution is depicted in the next subfigures. First, isolines which are close to the edge, i.e. in the neighborhood of the solid circle where the advection term is nonzero, are attracted from both sides to this edge. A small shock (steep gradient) is formed due to accumulation of these level lines (see Fig. 11.10 (top right)). In the regions outside the neighborhood of the circle, the advection term is vanishing and $g^0 \equiv 1$, so only intrinsic diffusion of level sets plays a role. This means that all inside level sets are shrinking and finally they disappear. Such a process is nothing else but a decrease of the maximum of our segmentation function until the upper level of the shock is achieved. It is clear that a flat region in the profile of segmentation function inside the circle is formed. Outside of the circle, level sets are also shrinking until they are attracted by nonzero velocity field and then they contribute to the shock. In the bottom left of Fig. 11.10, we see the shape of segmentation function u after such evolution, in the bottom right there are isocontours of such function accumulated on the edges. It is very easy to use one of them, e.g., $(\max(u) + \min(u))/2$, to get the circle.

The situation is not so straightforward for the highly nonconvex image depicted in Fig. 11.11. Our numerical observation leads to formation of steps in subsequent evolution of the segmentation function, which is understandable, because very different level sets of initial surface u^0 are attracted to different parts of the boundary of “batman.” Fortunately, we are a bit free in choosing the precise form of diffusion term in the segmentation model. After expansion of divergence, Eqs. (11.2) and (11.8) give the same advection term, $\nabla g^0 \cdot \nabla u$ (cf. Eq. (11.9)), so important advection mechanism which accumulates segmentation function along the shock is the same. However, diffusion mechanisms are a

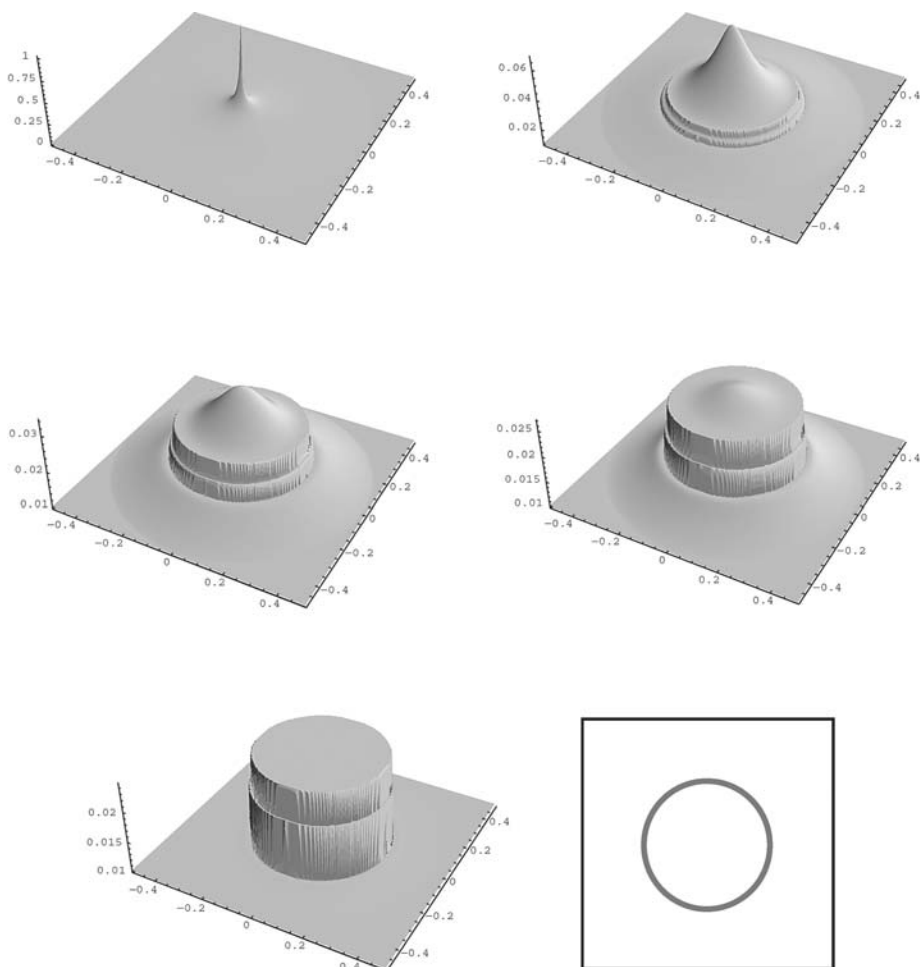


Figure 11.10: Subjective surface based segmentation of solid circle. We plot numerically computed time steps 0, 2, 10, 20, and 100. In the bottom right we see accumulation of level lines of segmentation function on the edges. In this experiment $\varepsilon = 10^{-10}$, so we are very close to level set flow equation (11.8) (color slide).

bit different. Eq. (11.2), in the case $\varepsilon = 1$, gives diffusion which is known as mean curvature flow of graphs. It means that no level sets of segmentation function move in the normal direction proportionally to curvature, but the graph of segmentation function moves (as 2D surface in 3D space) in the normal direction proportionally to the mean curvature. The large variations in the graph of segmentation function are then smoothed due to large mean curvature. Of course,

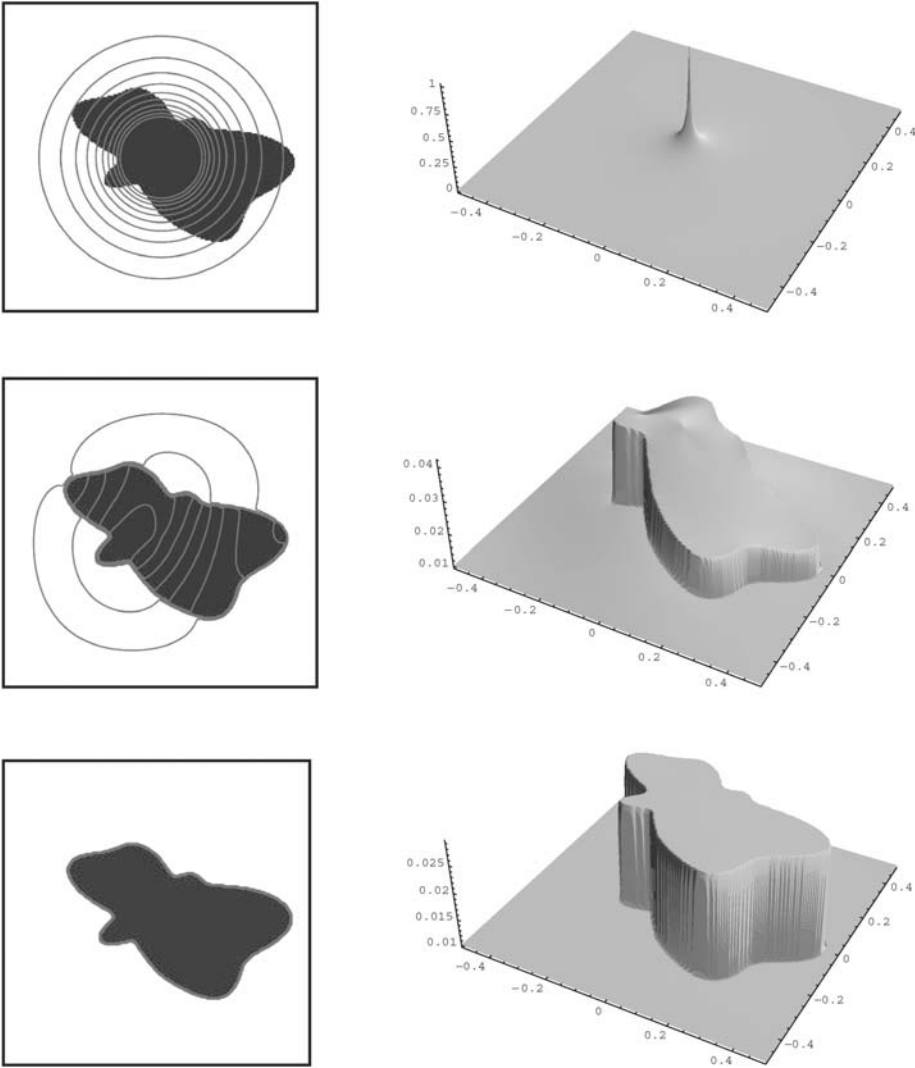


Figure 11.11: Subjective surface based segmentation of a “batman” image. In the left column we plot the black and white images to be segmented together with isolines of the segmentation function. In the right column there are shapes of the segmentation function. The rows correspond to time steps 0, 1, and 10, which gives the final result $\varepsilon = 1$ (color slide).

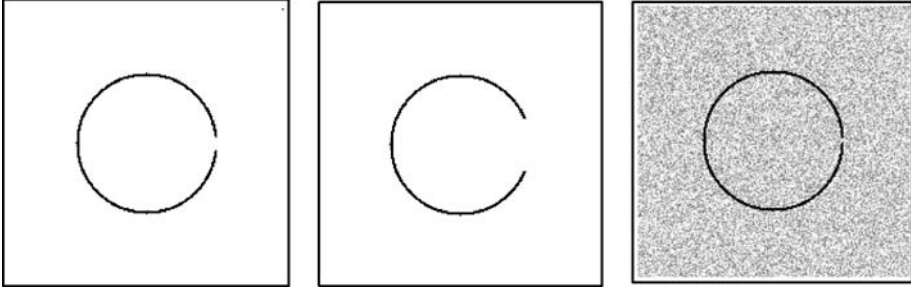


Figure 11.12: Three testing images. Circle with a smaller (left) and a big (middle) gap, and noisy circle with a gap.

the smoothing is applied only outside the edges. On the edges the advection dominates, since the mean curvature term is multiplied by a small value of g^0 . In Fig. 11.11 (bottom) we may see formation of a piecewise flat profile of the segmentation function, which can be again very simply used for extraction of “batman,” although, due to Dirichlet boundary data and $\varepsilon = 1$, this profile moves slowly downwards in subsequent evolution. In this (academic) example, the only goal was to smooth (flatten) the segmentation function inside and outside the edge, so the choice $\varepsilon = 1$ was really satisfactory. In the case $\varepsilon = 1$, Eq. (11.2) can be interpreted as a time relaxation for the minimization of the weighted area functional

$$A_{g^0} = \int_{\Omega} g^0 \sqrt{1 + |\nabla u|^2} dx,$$

or as the mean curvature motion of a graph in Riemann space with metric $g^0 \delta_{ij}$ [48].

In the next three testing images plotted in Fig. 11.12 we illustrate the role of the regularization parameter ε . The same choice, $\varepsilon = 1$, as in the previous image with complete edge, is clearly not appropriate for image object with a gap (Fig. 11.12 (left)), as seen in Fig. 11.13. We see that minimal-surface-like diffusion closes the gap with a smoothly varying “waterfall” like shape. Although this shape is in a sense stable (it moves downwards in a “self-similar form”), it is not appropriate for segmentation purposes. However, decreasing ε , i.e., if we stay closer to the curvature-driven level set flow (11.8), or in other words, if we stretch the Riemannian metric $g^0 \delta_{ij}$ in the vertical z direction [49], we get

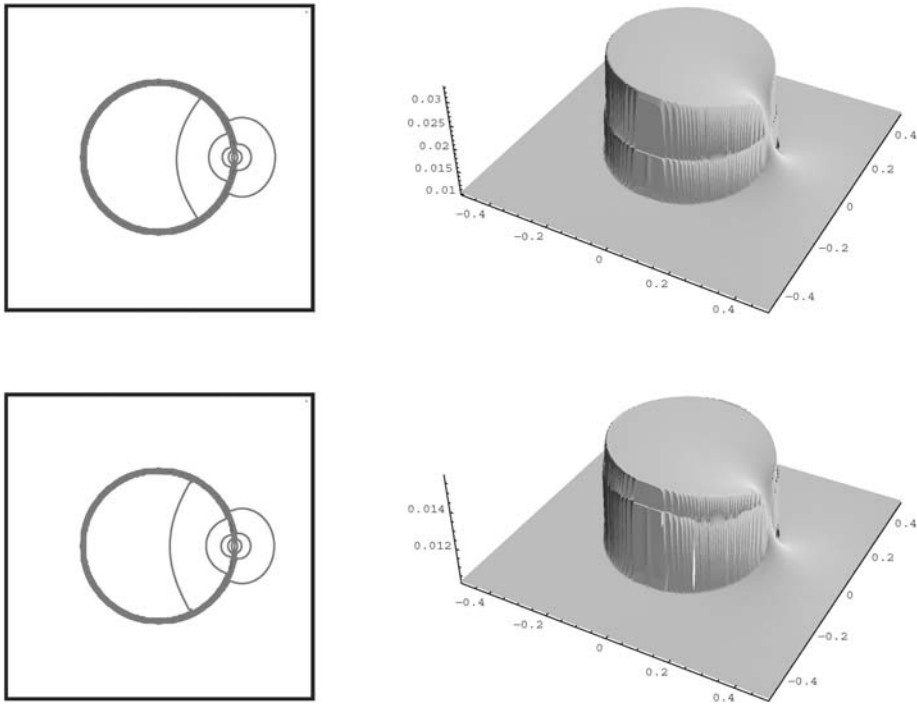


Figure 11.13: Experiment on testing image plotted in Fig. 11.12 (left). The results of evolution of the segmentation function (in the left its isolines, in the right its graphs) after 10 (top row) and 100 (bottom row) time steps. In this case, $\varepsilon = 1$, the shape is stable, but moving downwards in a “self-similar” form, so it is not utilizable as the segmentation result.

very good segmentation results as presented in Fig. 11.14. Of course, smaller ε is needed to close larger gaps (see Fig. 11.15).

If there is a noisy image as in Fig. 11.12 (right), the motion of level lines to shock is more irregular, but finally the segmentation function is smoothed as well (see Figs. 11.16 and 11.17). If the regularization parameter ε is small, then piecewise flat profile of the segmentation function will move very slowly downwards, so it is easy to stop the evolution and get the result of segmentation process.

In the presented experiments, we have seen that the solution of Eq. (11.2) is well suited to finding and completing edges in (noisy) images. Its advection–diffusion mechanism leads to promising results. In the next section we give an efficient and robust computational method for its solution.

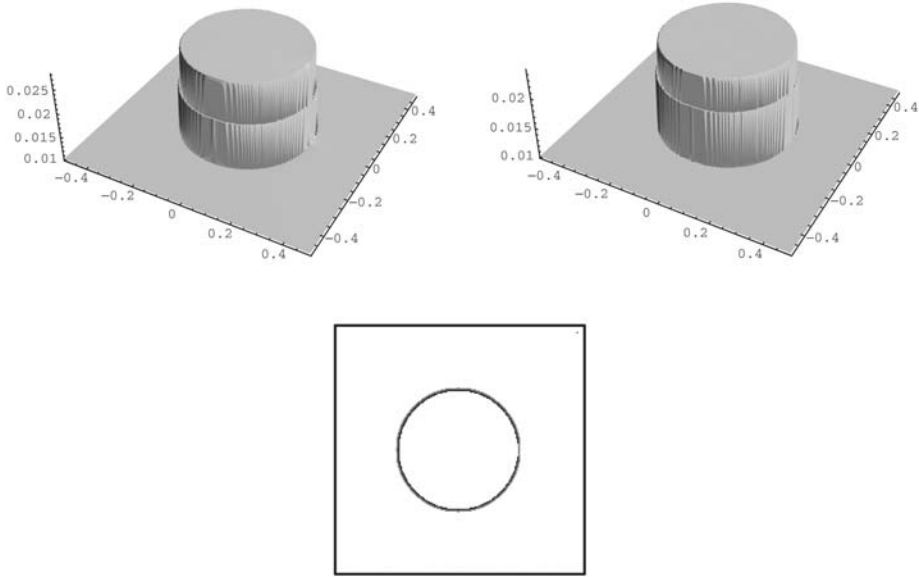


Figure 11.14: Results of the segmentation process for testing image plotted in Fig. 11.12 (left) using $\varepsilon = 10^{-2}$ (top left) and $\varepsilon = 10^{-5}$ (top right). The isline $(\max(u) + \min(u))/2$ well represents the segmented circle (bottom red line). For large range of ε , we get satisfactory results (color slide).

11.3 Semi-implicit Co-Volume Scheme

We present our method in discretization of Eq. (11.8), although we always use its ε -regularization (11.2) with a specific $\varepsilon > 0$. The notation is simpler in the case of (11.8) and it will be clear where regularization appears in the numerical scheme.

First we choose a uniform discrete time step τ and a variance σ of the smoothing kernel G_σ . Then we replace time derivative in (11.8) by backward difference. The nonlinear terms of the equation are treated from the previous time step while the linear ones are considered on the current time level, this means semi-implicitness of the time discretization. In the last decade, semi-implicit schemes have become a powerful tool in image processing, we refer e.g. to [3, 4, 25–27, 33, 37, 51, 57, 58].

Semi-implicit in time discretization. Let τ and σ be fixed numbers, I^0 be a given image, and u^0 be a given initial segmentation function. Then, for

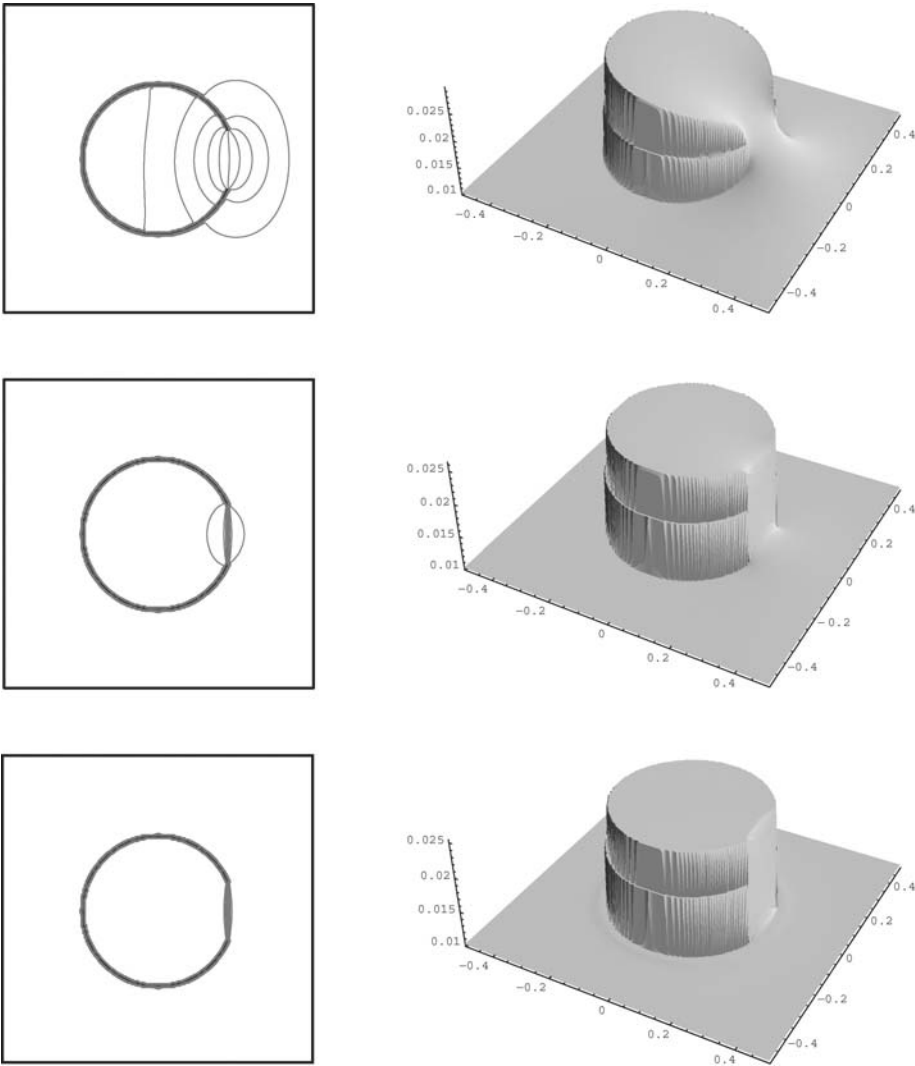


Figure 11.15: Segmentation of the circle with a big gap (Fig. 11.12 (middle)) using $\epsilon = 1$ (top), $\epsilon = 10^{-2}$ (middle), and $\epsilon = 10^{-5}$ (bottom). For bigger missing part a smaller ϵ is desirable. In the left column we see how close to the edges the isolines are accumulating and closing the gap, and in the right we see how steep the segmentation function is along the gap (color slide).

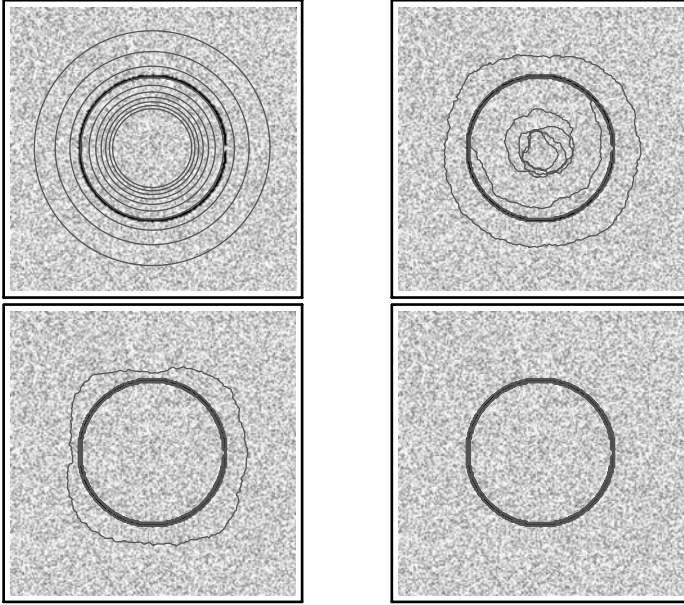


Figure 11.16: Isolines of the segmentation function in the segmentation of the noisy circle (Fig. 11.12 (right)) are shown in time steps 0, 50, 100, and 200. Since the gap is not so big we have chosen $\varepsilon = 10^{-1}$ (color slide).

$n = 1, \dots, N$, we look for a function w^n , solution of the equation,

$$\frac{1}{|\nabla w^{n-1}|} \frac{w^n - w^{n-1}}{\tau} = \nabla \cdot \left(g^0 \frac{\nabla w^n}{|\nabla w^{n-1}|} \right). \quad (11.11)$$

A digital image is given on a structure of pixels with rectangular shape, in general (red rectangles in Fig. 11.18). Since discrete values of I^0 are given in pixels and they influence the model, we will relate spatially discrete approximations of the segmentation function u also to image pixels, more precisely, to their centers (red points in Fig. 11.18). In every discrete time step of the method (11.11), we have to evaluate gradient of the segmentation function at the previous step $|\nabla w^{n-1}|$. For that goal, it is reasonable to put a triangulation (dashed black lines in Fig. 11.18) inside the pixel structure and take a piecewise linear approximation of the segmentation function on this triangulation. Such an approach will give a constant value of the gradient per triangle, allowing simple and clear construction of fully discrete system of equations. This is the main feature of the co-volume [25, 56] and finite element [13–15] methods in solving mean curvature flow in the level set formulation.

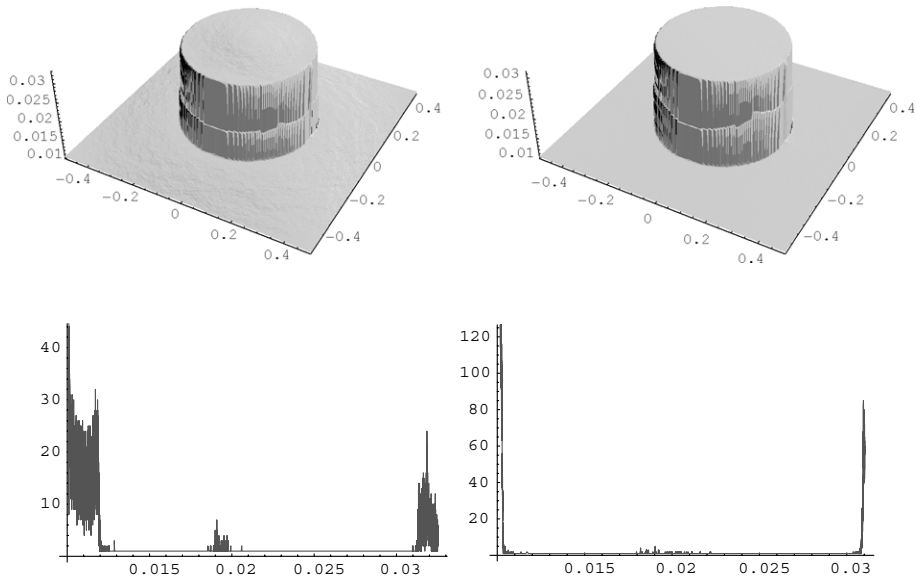


Figure 11.17: The graph of the segmentation function and its histograms in time steps 100 and 200 for the same experiment as presented in Fig. 11.16. The histograms give a practical advise to shorten the segmentation process in case of noisy images. For a noisy image, the formation of completely piecewise flat subjective surface takes longer time. However, the gaps in histogram of the segmentation function are developed soon. It allows to take any level inside these gaps and to visualize the corresponding level line to get desirable segmentation result (color slide).

As can be seen in Fig. 11.18, in our method the centers of pixels are connected by a new rectangular mesh and every new rectangle is splitted into four triangles. The centers of pixels will be called degree of freedom (DF) nodes. By this procedure we also get further nodes (at crossing of red lines in Fig. 11.18) which, however, will not represent degrees of freedom. We will call them non-degree of freedom (NDF) nodes. Let a function u be given by discrete values in the pixel centers, i.e. in DF nodes. Then in additional NDF nodes we take the average value of the neighboring DF nodal values. By such defined values in NDF nodes, a piecewise linear approximation u_h of u on the triangulation can be built. Let us note that we restrict further considerations in this chapter only to this type of grids. For triangulation $\mathcal{T}_h$, given by the previous construction, we construct a co-volume (dual) mesh. We modify a basic

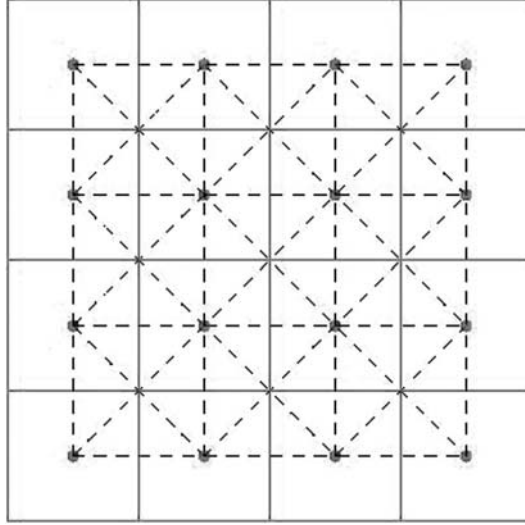


Figure 11.18: The image pixels (red solid lines) corresponding to co-volume mesh. Triangulation (black dashed lines) for the co-volume method with degree of freedom nodes (red round points) corresponding to centers of pixels (color slide).

approach given in [25, 56] in such a way that our co-volume mesh will consist of cells p associated only with DF nodes p of $\mathcal{T}_h$, say $p = 1, \dots, M$. Since there will be one-to-one correspondence between co-volumes and DF nodes, without any confusion, we use the same notation for them. In this way we have excluded the boundary nodes (due to Dirichlet boundary data) and NDF nodes.

For each DF node p of $\mathcal{T}_h$, let C_p denote the set of all DF nodes q connected to the node p by an edge. This edge will be denoted by σ_{pq} and its length by h_{pq} . Then every *co-volume* p is bounded by the lines (co-edges) e_{pq} that bisect and are perpendicular to the edges σ_{pq} , $q \in C_p$. By this construction, the co-volume mesh corresponds exactly to the pixel structure of the image inside the computational domain Ω where the segmentation is provided. We denote by $\mathcal{E}_{pq}$ the set of triangles having σ_{pq} as an edge. In a situation depicted in Fig. 11.18, every $\mathcal{E}_{pq}$ consists of two triangles. For each $T \in \mathcal{E}_{pq}$ let c_{pq}^T be the length of the portion of e_{pq} that is in T , i.e., $c_{pq}^T = m(e_{pq} \cap T)$, where m is a measure in $\mathbb{R}^{d-1}$. Let $\mathcal{N}_p$ be the set of triangles that have DF node p as a vertex. Let u_h be a piecewise linear function on triangulation $\mathcal{T}_h$. We will denote a constant value

of $|\nabla u_h|$ on $T \in \mathcal{T}_h$ by $|\nabla u_T|$ and define regularized gradients by

$$|\nabla u_T|_\varepsilon = \sqrt{\varepsilon^2 + |\nabla u_T|^2}. \quad (11.12)$$

We will use the notation $u_p = u_h(x_p)$, where x_p is the coordinate of the node p of triangulation $\mathcal{T}_h$.

With these notations, we are ready to derive co-volume spatial discretization. As is usual in finite volume methods [20, 34, 44], we integrate (11.11) over every co-volume p , $i = 1, \dots, M$. We get

$$\int_p \frac{1}{|\nabla u^{n-1}|} \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla \cdot \left(g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|} \right) dx. \quad (11.13)$$

For the right-hand side of (11.13), using divergence theorem we get

$$\begin{aligned} \int_p \nabla \cdot \left(g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|} \right) dx &= \int_{\partial p} \frac{g^0}{|\nabla u^{n-1}|} \frac{\partial u^n}{\partial \nu} ds \\ &= \sum_{q \in C_p} \int_{e_{pq}} \frac{g^0}{|\nabla u^{n-1}|} \frac{\partial u^n}{\partial \nu} ds. \end{aligned}$$

So we have an integral formulation of (11.11)

$$\int_p \frac{1}{|\nabla u^{n-1}|} \frac{u^n - u^{n-1}}{\tau} dx = \sum_{q \in C_p} \int_{e_{pq}} \frac{g^0}{|\nabla u^{n-1}|} \frac{\partial u^n}{\partial \nu} ds \quad (11.14)$$

expressing a “local mass balance” property of the scheme. Now the exact “fluxes” on the right-hand side and “capacity function” $\frac{1}{|\nabla u^{n-1}|}$ on the left-hand side (see e.g. [34]) will be approximated numerically using piecewise linear reconstruction of u^{n-1} on triangulation $\mathcal{T}_h$. If we denote g_T^0 approximation of g^0 on a triangle $T \in \mathcal{T}_h$, then for the approximation of the right-hand side of (11.14), we get

$$\sum_{q \in C_p} \left(\sum_{T \in \mathcal{E}_{pq}} c_{pq}^T \frac{g_T^0}{|\nabla u_T^{n-1}|} \right) \frac{u_q^n - u_p^n}{h_{pq}}, \quad (11.15)$$

and the left-hand side of (11.14) is approximated by

$$M_p m(p) \frac{u_p^n - u_p^{n-1}}{\tau}, \quad (11.16)$$

where $m(p)$ is a measure in $\mathbb{R}^d$ of co-volume p and either

$$M_p = \frac{1}{|\nabla u_p^{n-1}|}, \quad |\nabla u_p^{n-1}| = \sum_{T \in \mathcal{N}_p} \frac{m(T \cap p)}{m(p)} |\nabla u_T^{n-1}| \quad (11.17)$$

or

$$M_p = \sum_{T \in \mathcal{N}_p} \frac{m(T \cap p)}{m(p)} \frac{1}{|\nabla u_T^{n-1}|}. \quad (11.18)$$

The averaging of the gradients (11.17) has been used in [25, 56], and the approximation (11.18) is new and we have found it very useful regarding good convergence properties in solving the linear systems (see below) iteratively for $\varepsilon \ll 1$. Regularizations of both the approximations of the capacity function are as follows: either

$$M_p^\varepsilon = \frac{1}{|\nabla u_p^{n-1}|_\varepsilon} \quad (11.19)$$

or

$$M_p^\varepsilon = \sum_{T \in \mathcal{N}_p} \frac{m(T \cap p)}{m(p)} \frac{1}{|\nabla u_T^{n-1}|_\varepsilon}. \quad (11.20)$$

Now we can define coefficients, where the ε -regularization is taken into account, namely,

$$b_p^{n-1} = M_p^\varepsilon m(p), \quad (11.21)$$

$$a_{pq}^{n-1} = \frac{1}{h_{pq}} \sum_{T \in \mathcal{E}_{pq}} c_{pq}^T \frac{g_T^0}{|\nabla u_T^{n-1}|_\varepsilon}, \quad (11.22)$$

which together with (11.15) and (11.16) give the following.

Fully-discrete semi-implicit co-volume scheme. Let u_p^0 , $p = 1, \dots, M$, be given discrete initial values of the segmentation function. Then, for $n = 1, \dots, N$ we look for u_p^n , $p = 1, \dots, M$, satisfying

$$b_p^{n-1} u_p^n + \tau \sum_{q \in C_p} a_{pq}^{n-1} (u_p^n - u_q^n) = b_p^{n-1} u_p^{n-1}. \quad (11.23)$$

Theorem. There exists a unique solution $(u_1^n, \dots, u_M^n)$ of the scheme (11.23) for any $\tau > 0$, $\varepsilon > 0$ and for every $n = 1, \dots, N$. Moreover, for any $\tau > 0$, $\varepsilon > 0$ the following stability estimate holds

$$\min_p u_p^0 \leq \min_p u_p^n \leq \max_p u_p^n \leq \max_p u_p^0, \quad 1 \leq n \leq N. \quad (11.24)$$

Proof. The system (11.23) can be rewritten in the form

$$\left(b_p^{n-1} + \tau \sum_{q \in C_p} a_{pq}^{n-1} \right) u_p^n - \tau \sum_{q \in C_p} a_{pq}^{n-1} u_q^n = b_p^{n-1} u_p^{n-1}. \quad (11.25)$$

Applying Dirichlet boundary conditions, it gives the system of linear equations with a matrix, the off diagonal elements of which are symmetric and negative. Diagonal elements are positive and dominate the sum of absolute values of the nondiagonal elements in every row. Thus, the matrix of the system is symmetric and diagonally dominant M-matrix which imply that it always has a unique solution. The M-matrix property gives us the minimum–maximum principle, which can be seen by the following simple trick. We may temporarily rewrite (11.23) in the equivalent form

$$u_p^n + \frac{\tau}{b_p^{n-1}} \sum_{q \in C_p} a_{pq}^{n-1} (u_p^n - u_q^n) = u_p^{n-1} \quad (11.26)$$

and let $\max(u_1^n, \dots, u_M^n)$ be achieved in the node p . Then the second term on the left-hand side is non-negative and thus $\max(u_1^n, \dots, u_M^n) = u_p^n \leq u_p^{n-1} \leq \max(u_1^{n-1}, \dots, u_M^{n-1})$. In the same way we can prove the relation for minimum and together we have

$$\min_p u_p^{n-1} \leq \min_p u_p^n \leq \max_p u_p^n \leq \max_p u_p^{n-1}, \quad 1 \leq n \leq N, \quad (11.27)$$

which by recursion imply the desired stability estimate (11.24). $\square$

So far, we have said nothing about evaluation of g_T^0 included in coefficients (11.22). Since image is piecewise constant on pixels, we may replace the convolution by the weighted average to get $I_\sigma^0 := G_\sigma * I^0$ (see e.g. [37]) and then relate discrete values of I_σ^0 to pixel centers. Then, as above, we may construct its piecewise linear representation on triangulation and in such way we get constant value of ∇I_σ^0 on every triangle $T \in \mathcal{T}_h$. Another possibility is to solve numerically a linear heat equation for time t corresponding to variance σ with initial datum given by I^0 (see e.g. [3]). The convolution represents a preliminary smoothing of the data. It is also a theoretical tool to have bounded gradients and thus a strictly positive weighting coefficient g^0 . In practice, the evaluation of gradients on discrete grid (e.g., on triangulation described above) always gives bounded values. So, working on discrete grid, one can also avoid the convolution, especially if preliminary denoising is not needed or not desirable. Then it is possible to work directly with gradients of piecewise linear representation of I^0 in the evaluation of g_T^0 .

Our co-volume scheme in this paper is designed for the specific mesh (see Fig. 11.18) given by the rectangular pixel structure of 2D image. For simplicity of implementation and for the reader's convenience, we will write the

co-volume scheme in a “finite-difference notation.” As is usual for 2D rectangular grids, we associate co-volume p and its corresponding center (DF node) with a couple (i, j) , i will represent the vertical direction and j the horizontal direction. If Ω is a rectangular subdomain of the image domain where n_1 and n_2 are number of pixels in the vertical and horizontal directions, respectively, then $i = 1, \dots, m_1$, $j = 1, \dots, m_2$, $m_1 \leq n_1 - 2$, $m_2 \leq n_2 - 2$ and $M = m_1 m_2$. Similarly, the unknown value u^n_p is associated with $u^n_{i,j}$. For every co-volume p , the set $\mathcal{N}_p$ consists of eight triangles (see Fig. 11.18). In every discrete time step $n = 1, \dots, N$, and for every $i = 1, \dots, m_1$, $j = 1, \dots, m_2$, we compute absolute value of gradient on these eight triangles denoted by $G_{i,j}^k$, $k = 1, \dots, 8$. For that goal, using discrete values of u from the previous time step, we use the following expressions (we omit upper index $n - 1$ on u):

$$\begin{aligned} G_{i,j}^1 &= \sqrt{\left(\frac{0.5(u_{i,j+1}+u_{i+1,j+1}-u_{i,j}-u_{i+1,j})}{h}\right)^2 + \left(\frac{u_{i+1,j}-u_{i,j}}{h}\right)^2}, \\ G_{i,j}^2 &= \sqrt{\left(\frac{0.5(u_{i,j}+u_{i+1,j}-u_{i,j-1}-u_{i+1,j-1})}{h}\right)^2 + \left(\frac{u_{i+1,j}-u_{i,j}}{h}\right)^2}, \\ G_{i,j}^3 &= \sqrt{\left(\frac{0.5(u_{i+1,j-1}+u_{i+1,j}-u_{i,j-1}-u_{i,j})}{h}\right)^2 + \left(\frac{u_{i,j}-u_{i,j-1}}{h}\right)^2}, \\ G_{i,j}^4 &= \sqrt{\left(\frac{0.5(u_{i,j-1}+u_{i,j}-u_{i-1,j-1}-u_{i-1,j})}{h}\right)^2 + \left(\frac{u_{i,j}-u_{i,j-1}}{h}\right)^2}, \\ G_{i,j}^5 &= \sqrt{\left(\frac{0.5(u_{i,j}+u_{i-1,j}-u_{i,j-1}-u_{i-1,j-1})}{h}\right)^2 + \left(\frac{u_{i,j}-u_{i-1,j}}{h}\right)^2}, \\ G_{i,j}^6 &= \sqrt{\left(\frac{0.5(u_{i,j+1}+u_{i-1,j+1}-u_{i,j}-u_{i-1,j})}{h}\right)^2 + \left(\frac{u_{i,j}-u_{i-1,j}}{h}\right)^2}, \\ G_{i,j}^7 &= \sqrt{\left(\frac{0.5(u_{i,j}+u_{i,j+1}-u_{i-1,j}-u_{i-1,j+1})}{h}\right)^2 + \left(\frac{u_{i,j+1}-u_{i,j}}{h}\right)^2}, \\ G_{i,j}^8 &= \sqrt{\left(\frac{0.5(u_{i+1,j}+u_{i+1,j+1}-u_{i,j}-u_{i,j+1})}{h}\right)^2 + \left(\frac{u_{i,j+1}-u_{i,j}}{h}\right)^2}. \end{aligned}$$

In the same way, but only in the beginning of the algorithm, we compute values $G_{i,j}^{\sigma,k}$, $k = 1, \dots, 8$, changing u by I_σ^0 in the previous expressions, where I_σ^0 is a smoothed image as explained in the paragraph above. Then for every $i = 1, \dots, m_1$, $j = 1, \dots, m_2$ we construct (north, west, south, and east)

coefficients

$$\begin{aligned} n_{i,j} &= \tau \frac{1}{2} \sum_{k=1}^2 \frac{g(G_{i,j}^{\sigma,k})}{\sqrt{\varepsilon^2 + (G_{i,j}^k)^2}}, & w_{i,j} &= \tau \frac{1}{2} \sum_{k=3}^4 \frac{g(G_{i,j}^{\sigma,k})}{\sqrt{\varepsilon^2 + (G_{i,j}^k)^2}}, \\ s_{i,j} &= \tau \frac{1}{2} \sum_{k=5}^6 \frac{g(G_{i,j}^{\sigma,k})}{\sqrt{\varepsilon^2 + (G_{i,j}^k)^2}}, & e_{i,j} &= \tau \frac{1}{2} \sum_{k=7}^8 \frac{g(G_{i,j}^{\sigma,k})}{\sqrt{\varepsilon^2 + (G_{i,j}^k)^2}} \end{aligned}$$

and we use either (cf. (11.17))

$$m_{i,j} = \frac{1}{\sqrt{\varepsilon^2 + \left(\frac{1}{8} \sum_{k=1}^8 G_{i,j}^k\right)^2}}$$

or (cf. (11.18))

$$m_{i,j} = \frac{1}{8} \sum_{k=1}^8 \frac{1}{\sqrt{\varepsilon^2 + (G_{i,j}^k)^2}}$$

to define diagonal coefficients

$$c_{i,j} = n_{i,j} + w_{i,j} + s_{i,j} + e_{i,j} + m_{i,j} h^2.$$

If we define right-hand sides at the n th discrete time step by

$$r_{i,j} = m_{i,j} h^2 u_{i,j}^{n-1},$$

then for DF node corresponding to couple (i, j) we get the equation

$$c_{i,j} u_{i,j}^n - n_{i,j} u_{i+1,j}^n - w_{i,j} u_{i,j-1}^n - s_{i,j} u_{i-1,j}^n - e_{i,j} u_{i,j+1}^n = r_{i,j}. \quad (11.28)$$

Collecting these equations for all DF nodes and taking into account Dirichlet boundary conditions, we get the linear system to be solved.

We solve this system by the so-called SOR (successive over relaxation) iterative method, which is a modification of the basic Gauss–Seidel algorithm (see e.g. [46]). At the n th discrete time step we start the iterations by setting $u_{i,j}^{(0)} = u_{i,j}^{n-1}$, $i = 1, \dots, m_1$, $j = 1, \dots, m_2$. Then in every iteration $l = 1, \dots$ and for every $i = 1, \dots, m_1$, $j = 1, \dots, m_2$, we use the following two-step procedure:

$$\begin{aligned} Y &= (s_{i,j} u_{i-1,j}^{(l)} + w_{i,j} u_{i,j-1}^{(l)} + e_{i,j} u_{i,j+1}^{(l-1)} + n_{i,j} u_{i+1,j}^{(l-1)} + r_{i,j}) / c_{i,j} \\ u_{i,j}^{(l)} &= u_{i,j}^{(l-1)} + \omega(Y - u_{i,j}^{(l-1)}). \end{aligned}$$

We define squared L_2 norm of residuum at current iteration by

$$R^{(l)} = \sum_{i,j} (c_{i,j} u_{i,j}^{n(l)} - n_{i,j} u_{i+1,j}^{n(l)} - w_{i,j} u_{i,j-1}^{n(l)} - s_{i,j} u_{i-1,j}^{n(l)} - e_{i,j} u_{i,j+1}^{n(l)} - r_{i,j})^2.$$

The iterative process is stopped if $R^{(l)} < \text{TOL } R^{(0)}$. Since the computing of residuum is time consuming itself, we check it, e.g., after every ten iterations. The relaxation parameter ω is chosen by a user to improve convergence rate of the method; we have very good experience with $\omega = 1.85$ for this type of problems. Of course, the number of iterations depends on the chosen precision TOL, length of time step τ , and a value of the regularization parameter ε also plays a role. If one wants to weaken this dependence, more sophisticated approaches can be recommended (see e.g. [25, 35, 46] and paragraph below) but their implementation needs more programming effort. The semi-implicit co-volume method as presented above can be implemented in tens of lines.

We also outline shortly further approaches for solving the linear systems given in every discrete time step by (11.23). The system matrix has known (penta-diagonal) structure and moreover it is symmetric and diagonally dominant M-matrix. One could apply direct methods as Gaussian elimination, but this approach would lead to an immense storage requirements and computational effort. On the contrary, iterative methods can be applied in a very efficient way. In the previous paragraph we have already presented one of the most popular iterative methods, namely SOR. This method does not need additional storage, the matrix elements are used only to multiply the old solution values and convergence can be guaranteed for our special structure and properties of the system matrix. However, if the convergence is slow due to condition number of the system matrix (which increases with number of unknowns and for increasing τ and decreasing ε), faster iterative methods can be used. For example, the preconditioned conjugate gradient methods allow fast convergence, although they need more storage. If the storage requirements are reduced, then they can be very efficient and robust [25, 35]. For details of implementation of the efficient preconditioned iterative solvers for co-volume level set method, we refer to [25], cf. also [51]. Also an alternative direct approach based on operating splitting schemes can be recommended [57, 58].

In the next section, comparing CPU times, we will show that semi-implicit scheme is much more efficient and robust than explicit scheme for this type of problems. The explicit scheme combined with finite differences in space is

usually based on formulations such as (11.9) [7–9, 30, 31, 48–50] where all derivatives are expanded to get curvature and advection terms. Then, e.g., Eq. (11.2) for $\varepsilon = 1$ is written in the form

$$u_t = g^0 \frac{(1 + u_{x_2}^2)u_{x_1 x_1} - 2u_{x_1} u_{x_2} u_{x_1 x_2} + (1 + u_{x_1}^2)u_{x_2 x_2}}{1 + u_{x_1}^2 + u_{x_2}^2} + g_{x_1}^0 u_{x_1} + g_{x_2}^0 u_{x_2},$$

where u_s means partial derivative of a function u with respect to a variable s and x_1 and x_2 are spatial coordinates in the plane. In this form, it is not clear (reader may try) which terms to take from previous and which on the current time level, having in mind the unconditional stability of the method. Fully implicit time stepping would lead to a difficult nonlinear system solution, so the explicit approach is the one straightforwardly utilizable. In spite of that, the basic formulation (11.2) leads naturally to convenient semi-implicit time discretization.

Let us recall the usual criterion on numerical schemes for solving partial differential equations: numerical domain of dependence should contain physical domain of dependence. In diffusion processes, in spite of advection, a value of solution at any point is influenced by any other value of solution in a computational domain. This is naturally fulfilled by the semi-implicit scheme. We solve linear system of equations at every time step which, at every discrete point, takes into account contribution of all other discrete values in computational domain.

11.4 Discussion on Numerical Results

This section is devoted to the discussion on further numerical experiments computed by the semi-implicit co-volume level set method. In Section 11.2 we already discussed some examples which have been used mainly to illustrate the advection–diffusion mechanism of the segmentation equation (11.2) and the role of parameter ε in closing the gaps. In the sequel we will discuss the role of additional model parameters as well as all aspects of our implementation. We also compare the method with different approaches to confirm efficiency of our numerical scheme.

For a given discrete image I^0 with n_1, n_2 , the number of pixels in the vertical and horizontal directions, respectively, we define space discretization step $h = \frac{1}{n_1}$. It means, we embed the image into a rectangle $[-0.5 \frac{n_2}{n_1}, 0.5 \frac{n_2}{n_1}] \times [-0.5, 0.5]$. If one wants to use $h = 1$ (which would correspond to pixel size equals to 1),

all considerations can be changed accordingly. We prefer the above definition of spatial discretization step, because it is closer to standard approaches to numerical solution of PDEs.

First we give some CPU times overview of the method. Since we are interested in finding a “steady state” (see discussion in Section 11.2) of the evolution in order to stop the segmentation process, the important properties are the number of time steps needed to come to this “equilibrium” and a CPU time for every discrete time step. We discuss CPU times in the experiment related to segmentation of the circle with a gap given in Fig. 11.12 (left), computed using $\varepsilon = 10^{-2}$ (see Fig. 11.14 (top left)). The testing image has 200×200 pixels and the computational domain Ω corresponds to the whole image domain. Since for the boundary nodes we prescribe Dirichlet boundary conditions, we have $M = 198 \times 198$ degrees of freedom. As the criterion to recognize the “steady state,” we use a change in L_2 norm of solution between subsequent time steps, i.e., we check whether

$$\sqrt{\sum_p h^2 (u_p^n - u_p^{n-1})^2} < \delta$$

with a prescribed threshold δ . For the semi-implicit scheme and small ε (then the downwards motion of the “steady state” is very slow) a good choice of threshold is $\delta = 10^{-5}$.

Reasonable time steps for our semi-implicit method are of order $(10h)^2$, e.g., for the discussed example very good results regarding CPU times and precision have been obtained for $\tau \in [0.001, 0.01]$. Since by a classical criterion the precision of numerical schemes for parabolic equations is optimal for $\tau \approx h^2$, we have also computed such a case. But, no significant difference due to precision has been observed, only much longer CPU time was necessary. In our example $\tau = 5 \times 10^{-3}$ and 20 time steps yield the segmentation result (using threshold $\delta = 10^{-5}$). On 2.4 GHz Linux PC, the overall CPU time for this segmentation was 4.93 sec (i.e., approximately 0.25 sec for one time step including construction of coefficients and solving the linear system). This CPU time was obtained with $\text{TOL} = 10^{-3}$. Since we are mainly interested in “equilibrium,” one can also decide that such precision is not necessary in every discrete time step. With increasing TOL fewer numbers of SOR iterations are needed. Another way is to prescribe a fixed number (but not too small) of iterations in every time step, e.g., ten

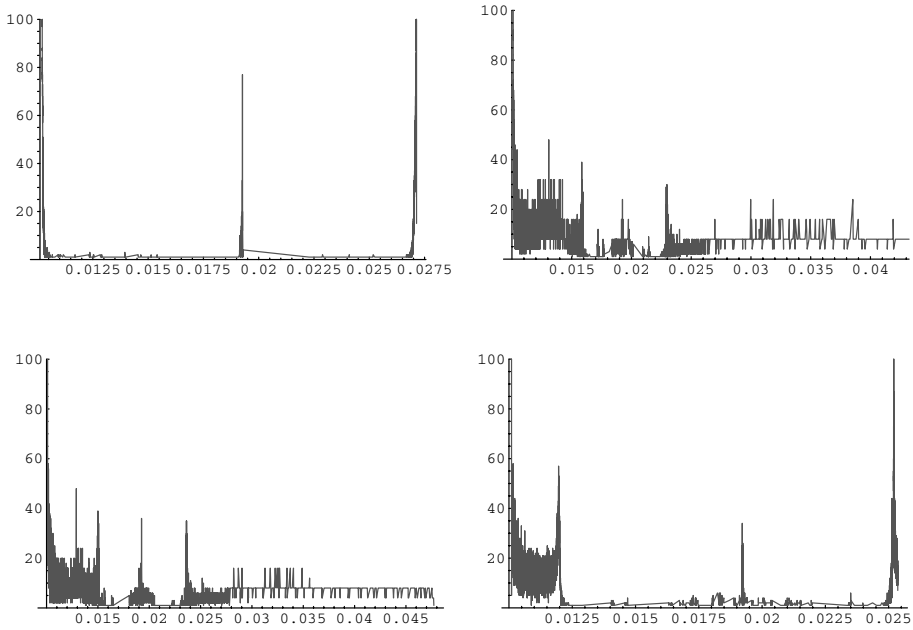


Figure 11.19: Histogram of the segmentation result given by semi-implicit scheme after 20 time steps (top left). Histograms of the segmentation function given by the explicit scheme after 500 (top right), 1000 (bottom left), and 5000 (bottom right) time steps (color slide).

prescribed SOR iterations lead to comparable segmentation with twice faster CPU time as mentioned above.

Now, let us look at the behavior of the explicit scheme in this example. We use the explicit version of the scheme (11.23) where also the second term on the left-hand side is taken from the $(n - 1)$ th time step. Then, due to stability reasons, we have to choose $\tau = 5 \times 10^{-6}$. Although one explicit time step takes just 0.05 sec (including construction of coefficients and explicit time update of the solution), to get a segmentation result comparable with the semi-implicit scheme we need about 10 000 time steps. In Fig. 11.19 we present histograms of the segmentation function, where the plotted range $[0, 100]$ in the vertical direction has been chosen for visualization. We compare histograms, because one cannot use the same threshold δ for explicit and semi-implicit schemes due to very small change in the solution between time steps in explicit scheme. In the top left, there is a histogram of the segmentation result given by semi-implicit scheme after 20 time steps. The shocks in solution (corresponding to outer and

inner edges of the circle) are given by two large gaps in histogram. In the top right there is a histogram of the segmentation function given by the explicit scheme after 500 time steps, and then after 1000 (bottom left) and 5000 (bottom right) time steps. We see that, due to necessity of small time step, the formation of the piecewise flat solution is very slow for explicit scheme. Although after 1000 time steps one can see the formation of two gaps which could be already used for detection of “final” segmentation result, the CPU time for 1000 steps of explicit scheme is 49.5 sec, which is ten times longer than for semi-implicit scheme. If we would like to obtain a similar histogram as plotted in the top left using an explicit scheme, we would need 100 times longer CPU time as in the case of semi-implicit scheme.

In all computations presented above, we have used $g(s) = \frac{1}{1+Ks^2}$, $K = 1$. In experiments without noise there is no significant difference by changing K . We get the same behavior of the method changing K from 0.1 to 10. It is understandable because the function g plays a role only along edges and its more ($K > 1$) or less ($K < 1$) quickly decreasing profile governs only speed by which level sets of solution are attracted to the edge from a small neighborhood. Everywhere else only pure mean curvature motion is considered ($g = 1$).

The situation is different for noisy images, e.g., depicted in Fig. 11.12 (right) and Figs. 11.16 and 11.17. The extraction of the circle in noisy environment takes a longer time (200 steps with $\tau = 0.01$ and $K = 1$) and it is even worse for $K = 10$. However, decreasing the parameter K gives stronger weight to mean curvature flow in noisy regions, so we can extract the circle fast, in only 20 steps with the same $\tau = 0.01$. In the case of noisy images, also the convolution plays a role. For example, if we switch off the convolution, the process is slower. But decreasing K can again improve the speed of segmentation process. In our computations we either do not apply convolution to I^0 or we use image presmoothing by $m \times m$ pixel mask with weights given by the Gauss function normalized to unit sum.

We start all computations with initial function given as a peak centered in a “focus point” inside the segmented object, as plotted, e.g., in Fig. 11.10 (top left). Such a function can be described for a circle with center s and radius R by $u^0(x) = \frac{1}{|x-s|+v}$, where s is the focus point and $\frac{1}{v}$ gives maximum of u^0 . Outside the circle we take value u^0 equal to $\frac{1}{R+v}$. If one needs zero Dirichlet boundary data, e.g., due to some theoretical reasons (cf. [11, 49]), the value $\frac{1}{R+v}$ can be subtracted from the peak-like profile. If the computational domain Ω corresponds to image domain, we use $R = 0.5$. For small objects a smaller R

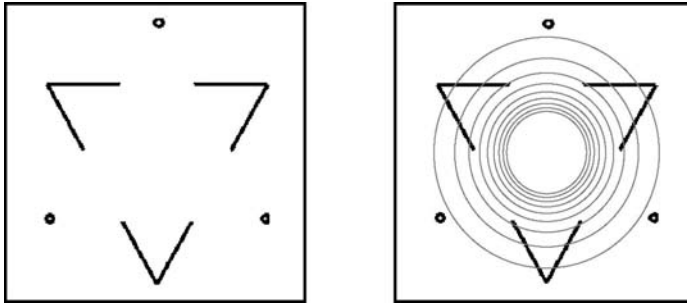


Figure 11.20: Image with subjective contours: double-Kanizsa triangle (left), and image together with isolines of initial segmentation function (right) (color slide).

can be used to speed up computations. Our choice of peak-like initial function is motivated by its nearly flat profile near the boundary of computational domain. However, other choices, e.g., $u^0(x) = 1 - \frac{|x-s|}{R}$, are also possible. If we put the focus point s not too far from the center of mass of the segmented object, we get only slightly different evolution of the segmentation function and same segmentation result.

Now we will discuss some further segmentation examples. In Fig. 11.20 we present image (234×227 pixels) with subjective contours of the classic triangle of Kanizsa. The phenomenon of contours that appear in the absence of physical gradients has attracted considerable interest among psychologists and computer vision scientists. Psychologists suggested a number of images that strongly require image completion to detect the objects. In Fig. 11.20 (left), two solid triangles appear to have well defined contours, even in completely homogeneous areas. Kanizsa called the contours without gradient “subjective contours” [29], because the missed boundaries are provided by the visual system of the subject. We apply our algorithm in order to extract the solid triangle and complete the boundaries. In Figs. 11.21 and 11.22 we present evolution of the segmentation function together with plots of level lines accumulating along edges and closing subjective contours areas. In this experiment we used $\varepsilon = 10^{-5}$, $K = 1$, $v = 0.5$, $\tau = 0.001$, $\text{TOL} = 10^{-3}$. For long time periods (from 60th to 300th time step) we can also easily detect subjective contours of the second triangle. The first one, given by closing of the solid interrupted lines, is presented in Fig. 11.22 (bottom), visualizing level line $(\min(u) + \max(u))/2$. Interestingly, for bigger ε the second triangle has not been detected.

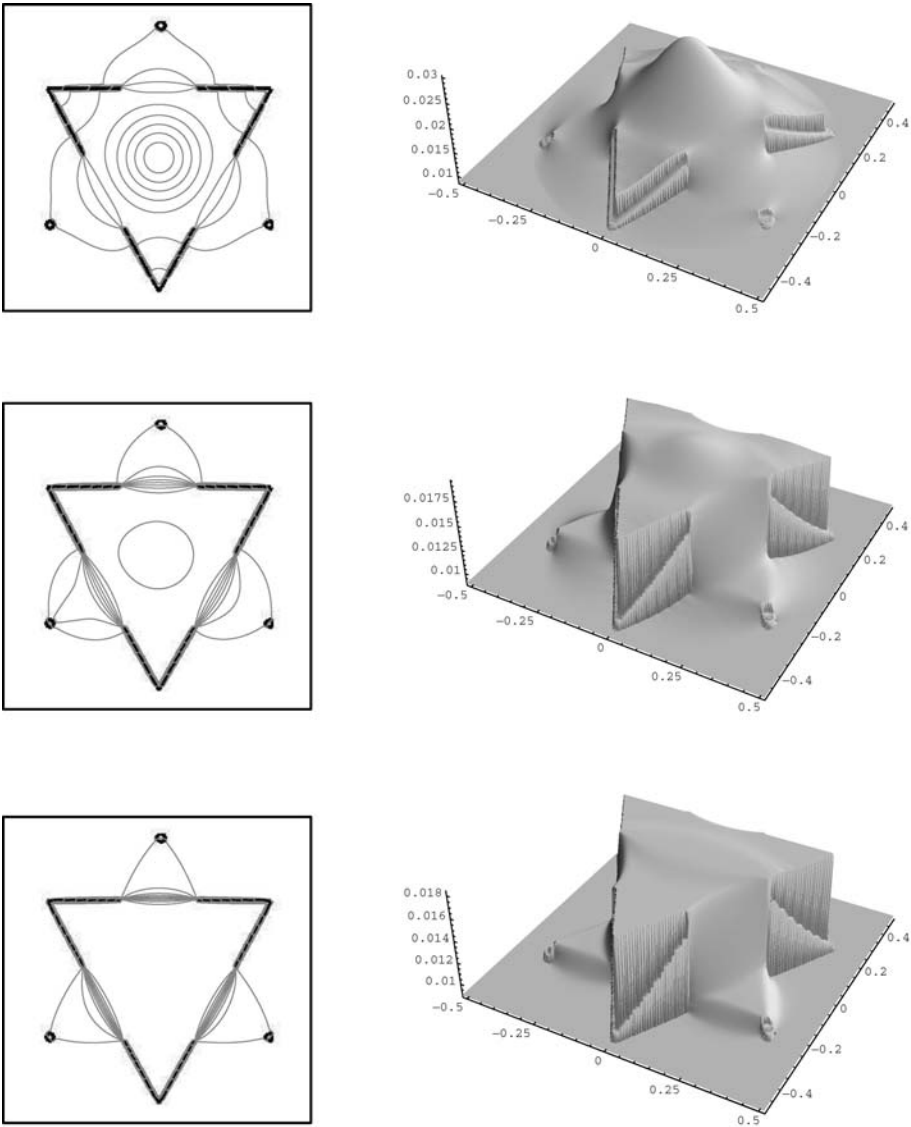


Figure 11.21: Level lines (left) and graphs of the segmentation function (right) in time steps 10, 30, and 60 (color slide).

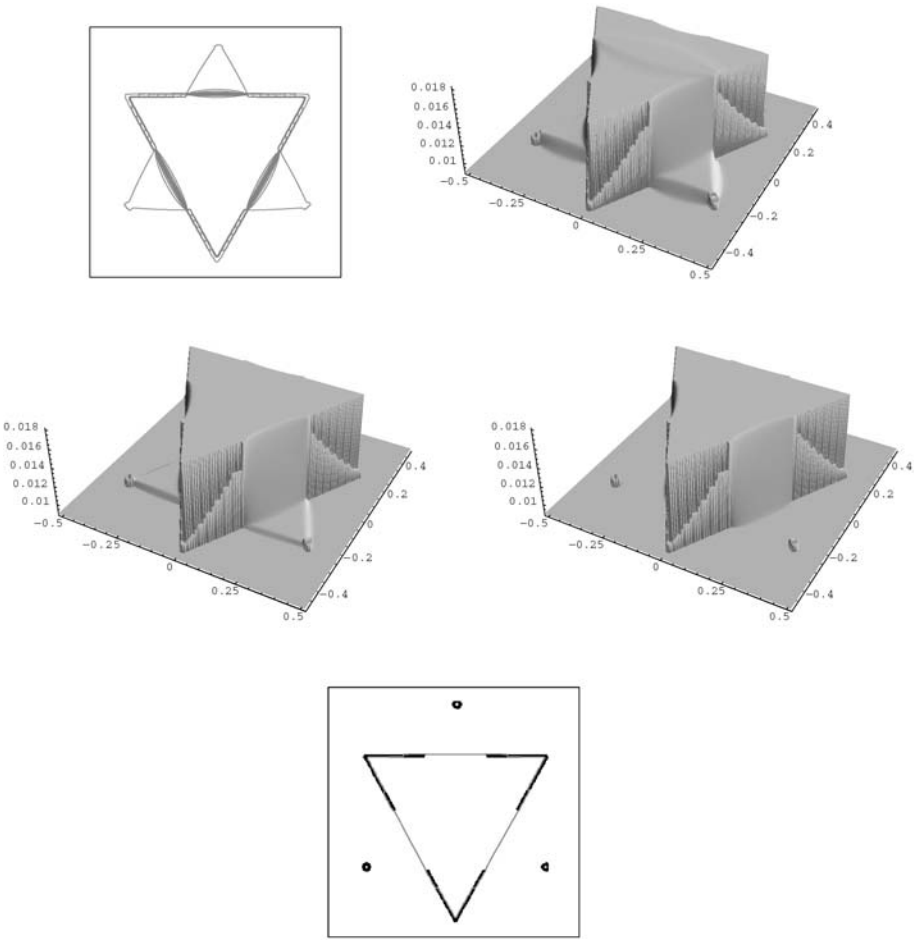


Figure 11.22: Level lines and graph of the segmentation function in time step 100 (top row). Then we show graphs of segmentation function after 300 and 800 steps (middle row). In the bottom row we plot the segmented Kanizsa triangle (color slide).

The next examples are related to medical image segmentation. First we process a 2D echocardiography (165×175 pixels) with high level of noise and gaps in ventricular and atrium boundaries (see Fig. 11.23).

In Fig. 11.24 we present segmentation of the left atrium. We start with peak-like segmentation function, $v = 1$, and we use $\varepsilon = 10^{-2}$, $K = 0.1$, $\tau = 0.001$, $\text{TOL} = 10^{-3}$, and $\delta = 10^{-5}$. In the top row of the figure we present the result of segmentation with no presmoothing of the given echocardiography. In such a case 68 time steps, with overall CPU time of 6.54 sec, were needed for threshold δ .

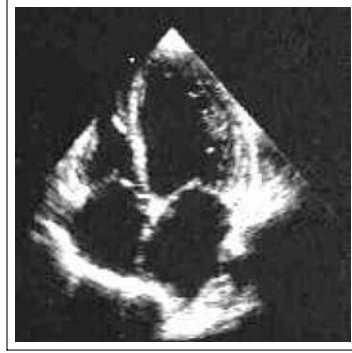


Figure 11.23: Echocardiographic image with high level of noise and gaps.

In the top right we see a graph of the final segmentation function. In the middle row we see its histogram (left) and zoom of the histogram around $\max(u)$ (right). By that we take level 0.057 for visualization of the boundary of segmented object (top left). In the bottom row we present the result of segmentation using 5×5 convolution mask. Such a result is a bit smoother and 59 time steps (CPU time = 5.65 sec) were used.

For visualization of the segmentation level line in further figures, we use the same strategy as above, i.e. the value of u just below the last peak of histogram (corresponding to upper “flat region”) is chosen. In segmentation of the right atrium, presented in Fig. 11.25, we took the same parameters as above and no presmoothing was applied. CPU time for 79 time steps was 7.59 sec. In segmentation of the left and right ventricles, with more destroyed boundaries, we use $K = 0.5$ and we apply 5×5 convolution mask (other parameters were same as above). Moreover, for the left ventricle we use double-peak-like initial function (see Fig. 11.26 (top)) to speed up the process for such highly irregular object. In that case 150 time steps (CPU time = 14.5 sec) were used. For the right ventricle, 67 time steps (CPU time = 6.57 sec) were necessary to get segmentation result, see Fig. 11.27.

In the last example given in Fig. 11.28, we present segmentation of the mammography (165×307 pixels). Without presmoothing of the given image and with parameters $\varepsilon = 10^{-1}$, $K = 0.1$, $\tau = 0.0001$, $v = 1$, $\text{TOL} = 10^{-3}$, and $\delta = 10^{-5}$ we get the segmentation after 72 time steps. Since there are no big gaps, we take larger ε and since the object is small (found in a shorter time) we use smaller time step τ .

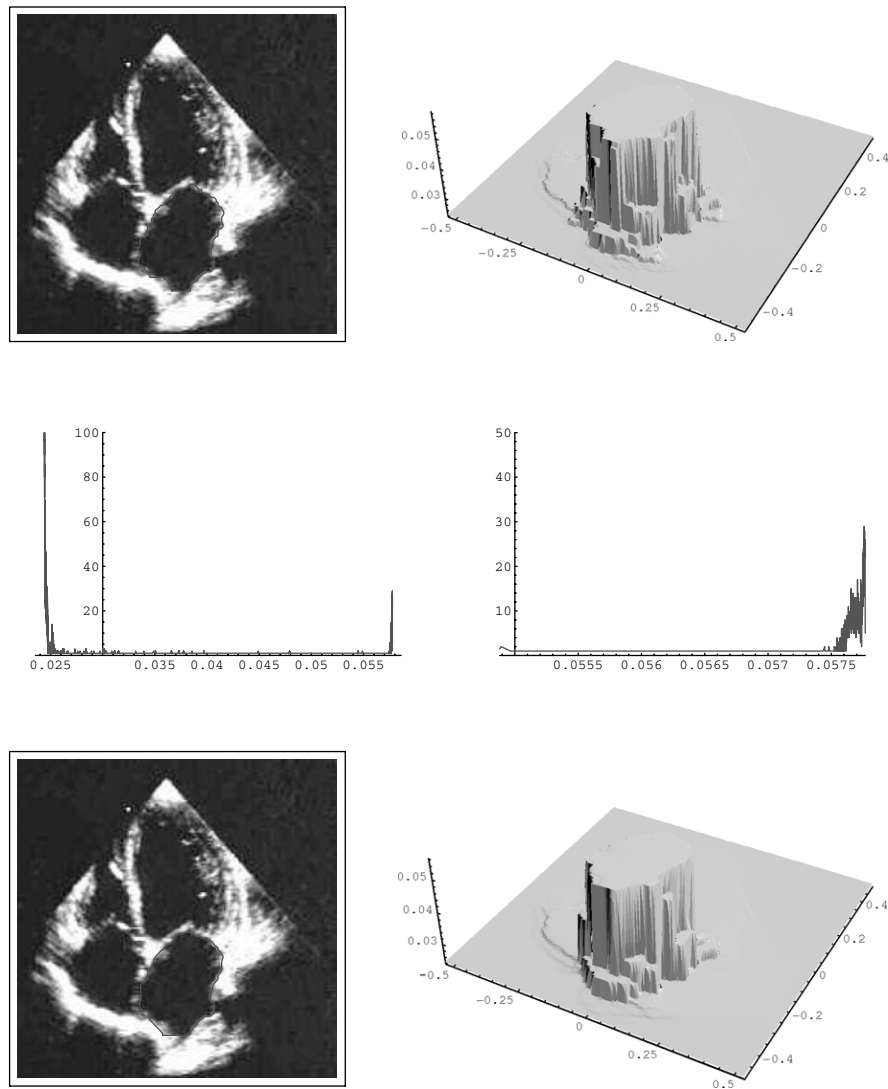


Figure 11.24: Segmentation level line and graph of the segmentation function for computation without convolution (top row) and histogram of the segmentation function and its zoom (middle row). Segmentation level line and graph of the segmentation function for computation with convolution (bottom row) (color slide).

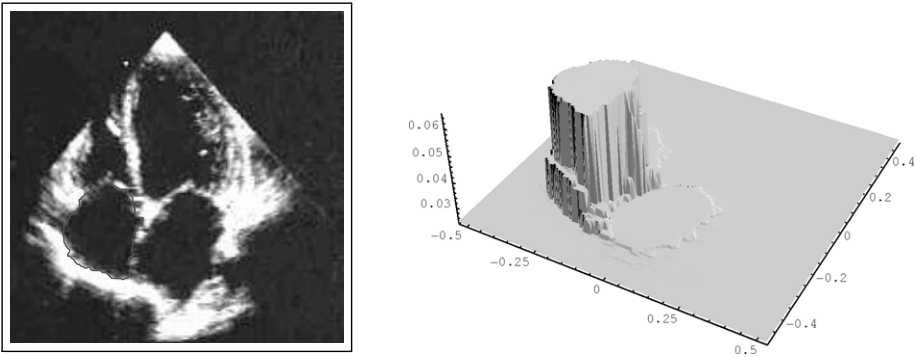


Figure 11.25: Segmentation level line and graph of the segmentation function for the right atrium (color slide).

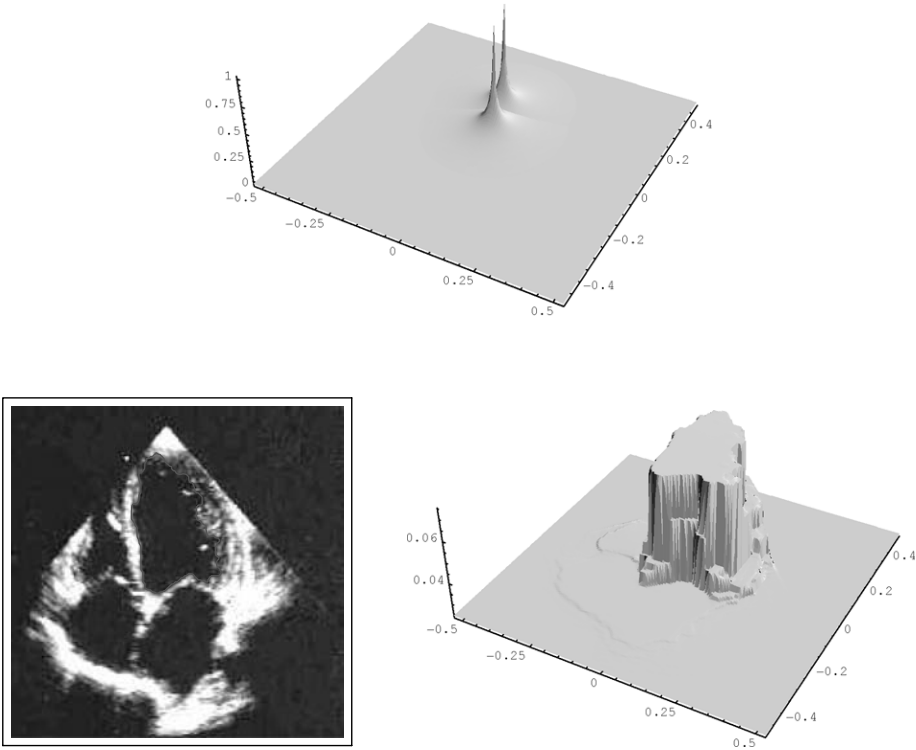


Figure 11.26: Initial double-peak segmentation function (top) and segmentation level line and graph of the segmentation function for the left ventricle (color slide).

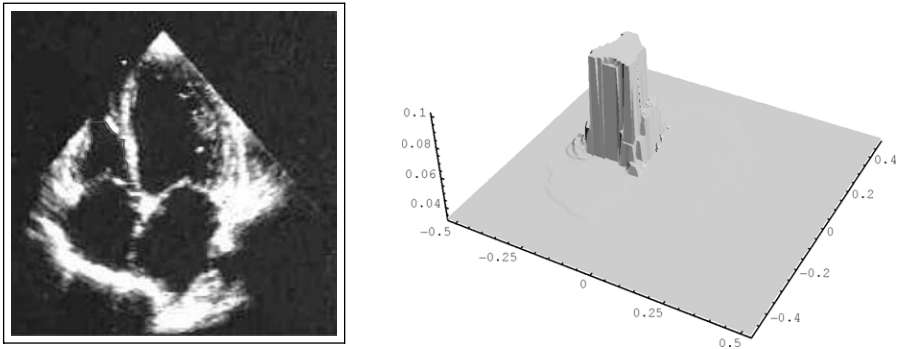


Figure 11.27: Segmentation level line and graph of the segmentation function for the right ventricle (color slide).

11.5 Conclusions

In this chapter we introduced the semi-implicit co-volume level set method for solving the segmentation equation given by the Riemannian mean curvature flow of graphs. We discussed basic properties of the model and the role of model parameters and gave all details for computer implementation of the numerical algorithm. We also showed unconditional stability of our method and its high efficiency for this type of problems. The computational results related to

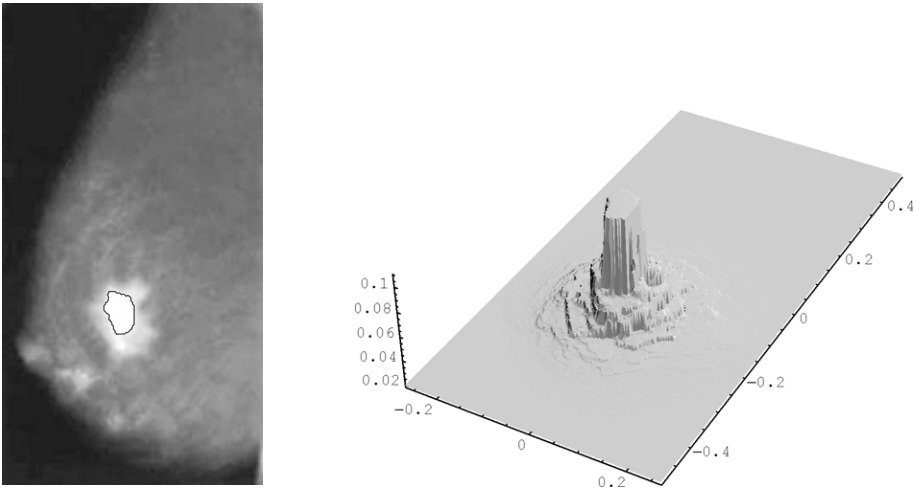


Figure 11.28: Segmentation level line and graph of the segmentation function for extraction of tumor in mammography (color slide).

medical image segmentation with partly missing boundaries and subjective contour extraction were discussed. The method was presented for 2D image segmentation. However, as is common in level set methods, the extension to 3D case is straightforward and can be done easily using ideas of this chapter.

11.6 Acknowledgements

This work was supported by NATO Collaborative Linkage Grant No. PST.CLG. 979123. The work of the first author was supported by the grant VEGA 1/0313/03, and by the European project “Visual Modeling” in Stefan Banach International Mathematical Centre PAN and ICM, Warsaw University. This work was partially supported by MIUR, grant number 2002013422-005.

Questions

1. *Outline the level set segmentation models used in the last decade. What is an advection–diffusion mechanism in such models?*
2. *What is the difference between previous level set segmentation models and Riemannian mean curvature flow of graphs discussed in this chapter?*
3. *What are the main principles and advantages of the semi-implicit time discretization?*
4. *How is the segmentation partial differential equation (11.2) discretized by the co-volume method?*
5. *What are the differences between semi-implicit co-volume method and explicit finite difference method?*
6. *What are the properties of the system matrix given by the semi-implicit co-volume scheme?*
7. *How can you get unconditional stability of the semi-implicit co-volume level set method?*
8. *What are the efficient methods for solving linear systems arising in the semi-implicit co-volume level set method?*

Bibliography

- [1] Alvarez, L., Guichard, F., Lions, P.L., and Morel, J.M., Axioms and fundamental equations of image processing, *Arch. Rat. Mech. Anal.*, Vol. 123, pp. 200–257, 1993.
- [2] Alvarez, L., Lions, P. L., and Morel, J. M., Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numer. Anal.*, Vol. 29, pp. 845–866, 1992.
- [3] Bänsch, E. and Mikula, K., A coarsening finite element strategy in image selective smoothing, *Comput. Vis. Sci.*, Vol. 1, No. 1, pp. 53–61, 1997.
- [4] Bänsch, E. and Mikula, K., Adaptivity in 3D image processing, *Comput. Vis. Sci.*, Vol. 4, No. 1, pp. 21–30, 2001.
- [5] Catté, F., Lions, P. L., Morel, J. M., and Coll, T., Image selective smoothing and edge detection by nonlinear diffusion, *SIAM J. Numer. Anal.*, Vol. 29, pp. 182–193, 1992.
- [6] Caselles, V., Catté, F., Coll, T., and Dibos, F., A geometric model for active contours in image processing, *Numer. Math.*, Vol. 66, pp. 1–31, 1993.
- [7] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic active contours, In: *Proceedings International Conference on Computer Vision'95*, Boston, pp. 694–699, 1995.
- [8] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic active contours, *Int. J. Comput. Vis.*, Vol. 22, pp. 61–79, 1997.
- [9] Caselles, V., Kimmel, R., Sapiro, G., and Sbert, C., Minimal surfaces: A geometric three dimensional segmentation approach, *Numer. Math.*, Vol. 77, pp. 423–451, 1997.
- [10] Chen, Y.-G., Giga, Y., and Goto, S., Uniqueness and existence of viscosity solutions of generalized mean curvature flow equation, *J. Diff. Geom.*, Vol. 33, pp. 749–786, 1991.
- [11] Citti, G. and Manfredini, M., Long time behavior of Riemannian mean curvature flow of graphs, *J. Math. Anal. Appl.*, Vol. 273, No. 2, pp. 353–369, 2002.

- [12] Crandall, M. G., Ishii, H., and Lions, P. L., User's guide to viscosity solutions of second order partial differential equations, *Bull. (NS) Amer. Math. Soc.*, Vol. 27, pp. 1–67, 1992.
- [13] Deckelnick, K. and Dziuk, G., Convergence of a finite element method for non-parametric mean curvature flow, *Numer. Math.*, Vol. 72, pp. 197–222, 1995.
- [14] Deckelnick, K. and Dziuk, G., Error estimates for a semi implicit fully discrete finite element scheme for the mean curvature flow of graphs, *Interfaces Free Bound.*, Vol. 2, No. 4, pp. 341–359, 2000.
- [15] Deckelnick, K. and Dziuk, G., A fully discrete numerical scheme for weighted mean curvature flow, *Numer. Math.*, Vol. 91, pp. 423–452, 2002.
- [16] Dziuk, G., Algorithm for evolutionary surfaces, *Numer. Math.*, Vol. 58, pp. 603–611, 1991.
- [17] Dziuk, G., Convergence of a semi discrete scheme for the curve shortening flow, *Math. Models Methods Appl. Sci.*, Vol. 4, pp. 589–606, 1994.
- [18] Dziuk, G., Discrete anisotropic curve shortening flow, *SIAM J. Numer. Anal.*, Vol. 36, pp. 1808–1830, 1999.
- [19] Evans, L. C. and Spruck, J., Motion of level sets by mean curvature I, *J. Diff. Geom.*, Vol. 33, pp. 635–681, 1991.
- [20] Eymard, R., Gallouet, T., and Herbin, R., The finite volume method, In: *Handbook for Numerical Analysis*, Vol. 7, Ciarlet, Ph., and Lions, J. L., eds, Elsevier, Amsterdam, 2000.
- [21] Frolkovič, P. and Mikula, K., Flux-based level set method: A finite volume method for evolving interfaces, Preprint IWR/SFB 2003–15, Interdisciplinary Center for Scientific Computing, University of Heidelberg, 2003.
- [22] Gage, M. and Hamilton, R. S., The heat equation shrinking convex plane curves, *J. Diff. Geom.*, Vol. 23, pp. 69–96, 1986.
- [23] Grayson, M., The heat equation shrinks embedded plane curves to round points, *J. Diff. Geom.*, Vol. 26, pp. 285–314, 1987.

- [24] Handlovičová, A., Mikula, K., and Sarti, A., Numerical solution of parabolic equations related to level set formulation of mean curvature flow, *Comput. Vis. Sci.*, Vol. 1, No. 2, pp. 179–182, 1999.
- [25] Handlovičová, A., Mikula, K., and Sgallari, F., Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution, *Numer. Math.*, Vol. 93, pp. 675–695, 2003.
- [26] Handlovičová, A., Mikula, K., and Sgallari, F., Variational numerical methods for solving nonlinear diffusion equations arising in image processing, *J. Vis. Commun. Image Represent.*, Vol. 13, pp. 217–237, 2002.
- [27] Kačur, J. and Mikula, K., Solution of nonlinear diffusion appearing in image smoothing and edge detection, *Appl. Numer. Math.*, Vol. 17, pp. 47–59, 1995.
- [28] Kačur, J. and Mikula, K., Slow and fast diffusion effects in image processing, *Comput. Vis. Sci.*, Vol. 3, No. 4, pp. 185–195, 2001.
- [29] Kanizsa, G., *Organization in Vision*, Praeger, New York, 1979.
- [30] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., Gradient flows and geometric active contours models, In: *Proceedings International Conference on Computer Vision'95*, Boston, 1995.
- [31] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., Conformal curvature flows: From phase transitions to active vision, *Arch. Rat. Mech. Anal.*, Vol. 134, pp. 275–301, 1996.
- [32] Kass, M., Witkin, A., and Terzopoulos, D., Snakes: Active contour models, *Int. J. Comput. Vis.*, Vol. 1, pp. 321–331, 1987.
- [33] Krivá, Z. and Mikula, K., An adaptive finite volume scheme for solving nonlinear diffusion equations in image processing, *J. Vis. Commun. Image Represent.*, Vol. 13, pp. 22–35, 2002.
- [34] Le Veque, R., *Finite volume methods for hyperbolic problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002.
- [35] Lin, C. J. and Moré, J. J., Incomplete Cholesky factorizations with limited memory, *SIAM. J. Sci. Comput.*, Vol. 21, pp. 24–45, 1999.

- [36] Malladi, R., Sethian, J. A., and Vemuri, B., Shape modeling with front propagation: A level set approach, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, pp. 158–174, 1995.
- [37] Mikula, K. and Ramarosy, N., Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing, *Numer. Math.*, Vol. 89, No. 3, pp. 561–590, 2001.
- [38] Mikula, K., Sarti, A., and Lamberti, C., Geometrical diffusion in 3D-echocardiography, *Proceedings of ALGORITMY'97—Conference on Scientific Computing, West Tatra Mountains, Zuberec*, pp. 167–181, 1997.
- [39] Mikula, K. and Ševčovič, D., Evolution of plane curves driven by a nonlinear function of curvature and anisotropy, *SIAM J. Appl. Math.*, Vol. 61, pp. 1473–1501, 2001.
- [40] Mikula, K. and Ševčovič, D., Computational and qualitative aspects of evolution of curves driven by curvature and external force, *Comput. Visual. Sci.*, Vol. 6, No. 4, pp. 211–225, 2004.
- [41] Mikula, K. and Sgallari, F., Semi-implicit finite volume scheme for image processing in 3D cylindrical geometry, *J. Comput. Appl. Math.*, Vol. 161, No. 1, pp. 119–132, 2003.
- [42] Osher, S. and Fedkiw, R., *Level set methods and dynamic implicit surfaces*, Springer-Verlag, Berlin, 2003.
- [43] Osher, S. and Sethian, J. A., Front propagating with curvature dependent speed: Algorithms based on the Hamilton–Jacobi formulation, *J. Comput. Phys.*, Vol. 79, pp. 12–49, 1988.
- [44] Patankar, S., *Numerical heat transfer and fluid flow*, Hemisphere Publications, Corporation, New York, 1980.
- [45] Perona, P. and Malik, J., Scale space and edge detection using anisotropic diffusion, In: *Proc. IEEE Computer Society Workshop on Computer Vision*, 1987.
- [46] Saad, Y., *Iterative methods for sparse linear systems*, PWS Publications Company, 1996.

- [47] Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, Cambridge, 2001.
- [48] Sarti, A., Malladi, R., and Sethian, J. A., Subjective Surfaces: A Method for Completing Missing Boundaries, *Proc. Nat. Acad. Sci. USA*, Vol. 12, No. 97, pp. 6258–6263, 2000.
- [49] Sarti, A. and Citti, G., Subjective Surfaces and Riemannian Mean Curvature Flow Graphs, *Acta Math. Univ. Comenianae*, Vol. 70, No. 1, pp. 85–104, 2001.
- [50] Sarti, A., Malladi, R., and Sethian, J. A., Subjective Surfaces: A Geometric Model for Boundary Completion, *Int. J. Comput. Vis.*, Vol. 46, No. 3, pp. 201–221, 2002.
- [51] Sarti, A., Mikula, K., and Sgallari, F., Nonlinear multiscale analysis of three-dimensional echocardiographic sequences, *IEEE Trans. Med. Imaging*, Vol. 18, pp. 453–466, 1999.
- [52] Sarti, A., Mikula, K., Sgallari, F., and Lamberti, C., Nonlinear multiscale analysis models for filtering of 3D + time biomedical images, In: *Geometric Methods in Bio-medical Image Processing*, Malladi, R., ed., Springer, Berlin, pp. 107–128, 2002.
- [53] Sarti, A., Mikula, K., Sgallari, F., and Lamberti, C., Evolutionary partial differential equations for biomedical image processing, *J. Biomed. Inform.*, Vol. 35, pp. 77–91, 2002.
- [54] Sethian, J. A., Numerical algorithm for propagating interfaces: Hamilton–Jacobi equations and conservation laws, *J. Diff. Geom.*, Vol. 31, pp. 131–161, 1990.
- [55] Sethian, J. A., *Level set methods and fast marching methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*, Cambridge University Press, Cambridge, 1999.
- [56] Walkington, N. J., Algorithms for computing motion by mean curvature, *SIAM J. Numer. Anal.*, Vol. 33, No. 6, pp. 2215–2238, 1996.

- [57] Weickert, J., *Anisotropic Diffusion in Computer Vision*, Teubner-Stuttgart, 1998.
- [58] Weickert, J., Romeny, B. M. t. H., and Viergever, M. A., Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE Trans. Image Process.*, Vol. 7, No. 3, pp. 398–410, 1998.

The Editors



Dr. Jasjit S. Suri received his BS in computer engineering with distinction from Maulana Azad College of Technology, Bhopal, India, his MS in computer sciences from University of Illinois, Chicago, and Ph.D. in electrical engineering from University of Washington, Seattle. He has been working in the field of computer engineering/imaging sciences for 20 years. He has published more than 125 technical papers in body imaging. He is a lifetime member of research engineering societies: Tau-Beta Pi, Eta-Kappa-Nu, Sigma-Xi, and a member of NY Academy of Sciences, Engineering in Medicine and Biology Society (EMBS), SPIE, ACM, and is also a senior member at IEEE. He is in the editorial board/reviewer of several international journals such as Real Time Imaging, Pattern Analysis and Applications, Engineering in Medicine and Biology, Radiology, Journal of Computer Assisted Tomography, IEEE Transactions of Information Technology in Biomedicine and IASTED Board.

He has chaired image processing tracks at several international conferences and has given more than 40 international presentations/seminars. Dr. Suri has written four books in the area of body imaging (such as cardiology, neurology, pathology, mammography, angiography, atherosclerosis imaging) covering medical image segmentation, image and volume registration, and physics of medical imaging modalities like: MRI, CT, X-ray, PET, and ultrasound. He also holds several United States patents. Dr. Suri has been listed in Who's Who seven times, is a recipient of president's gold medal in 1980, and has received more than 50 scholarly and extracurricular awards during his career. He is also a Fellow of American Institute of Medical and Biological Engineering (AIMBE) and ABI. Dr. Suri's major interests are: computer vision, graphics and image processing (CVGIP), object oriented programming, image guided surgery and teleimaging. Dr. Suri had worked with Philips Medical Systems and Siemens Medical Research Divisions. He is also a visiting professor with the department of computer science, University of Exeter, Exeter, UK. Currently, Dr. Suri is with JWT Inc.



Dr. David Wilson is a professor of biomedical engineering and radiology, Case Western Reserve University. He has research interests in image analysis, quantitative image quality, and molecular imaging, and he has a significant track record of federal research funding in these areas. He has over 60 refereed journal publications and has served as a reviewer for several leading journals. Professor Wilson has six patents and two pending patents in medical imaging. Professor Wilson has been active in the development of international conferences; he was Track Chair at the 2002 EMBS/BMES conference, and he was Technical Program Co-Chair for the 2004 IEEE International Symposium on Biomedical Imaging. Professor Wilson teaches courses in biomedical imaging, and biomedical image processing and analysis. He has advised many graduate and undergraduate

students, all of whom are quite exceptional, and has been primary research advisor for over 16 graduate students since starting his academic career. Prior to joining CWRU, he worked in x-ray imaging at Siemens Medical Systems at sites in New Jersey and Germany. He obtained his PhD from Rice University. Professor Wilson has actively developed biomedical imaging at CWRU. He has led a faculty recruitment effort, and he has served as PI or has been an active leader on multiple research and equipment developmental awards to CWRU, including an NIH planning grant award for an In Vivo Cellular and Molecular Imaging Center and an Ohio Wright Center of Innovation award. He can be reached at dlw@po.cwru.edu.



Dr. Swamy Laxminarayan currently serves as the chief of biomedical information engineering at the Idaho State University. Previous to this, he held several senior positions both in industry and academia. These have included serving as the chief information officer at the National Louis University, director of the pharmaceutical and health care information services at NextGen Internet (the premier Internet organization that spun off from the NSF sponsored John von Neuman National Supercomputer Center in Princeton), program director of biomedical engineering and research computing and program director of computational biology at the University of Medicine and Dentistry in New Jersey, vice-chair of Advanced Medical Imaging Center, director of clinical computing at the Montefiore Hospital and Medical Center and the Albert Einstein College of Medicine in New York, director of the VocalTec High Tech Corporate University in New Jersey, and the director of the Bay Networks Authorized Center in Princeton. He has also served as an adjunct professor of biomedical engineering

at the New Jersey Institute of Technology, a clinical associate professor of health informatics, visiting professor at the University of Bruno in Czech Republic, and an honorary professor of health sciences at Tsinghua University in China.

As an educator, researcher, and technologist, Prof. Laxminarayan has been involved in biomedical engineering and information technology applications in medicine and health care for over 25 years and has published over 250 scientific and technical articles in international journals, books, and conferences. His expertise lies in the areas of biomedical information technology, high performance computing, digital signals and image processing, bioinformatics, and physiological systems analysis. He is the co-author of the book *State-of-the-Art PDE and Level Sets Algorithmic Approaches to Static and Motion Imagery Segmentation* published by Kluwer Publications and the book *Angiography Imaging: State-of-the-Art-Acquisition, Image Processing and Applications Using Magnetic Resonance, Computer Tomography, Ultrasound and X-ray, Emerging Mobile E-Health Systems* published by the CRC Pres and two volumes of *Handbook of Biomedical Imaging* to be published by Kluwer Publications. He has also worked as the editor/co-editor of 20 international conferences and has served as a keynote speaker in international conferences in 13 countries.

He is the founding editor-in-chief and editor emeritus of IEEE Transactions on Information Technology in Biomedicine. He served as an elected member of the administrative and executive committees in the IEEE Engineering in Medicine and Biology Society and as the society's vice president for 2 years. His other IEEE roles include his appointments as program chair and general conference chair of about 20 EMBS and other IEEE conferences, an elected member of the IEEE Publications and Products Board, member of the IEEE Strategic Planning and Transnational Committees, member of the IEEE Distinguished Lecture Series, delegate to the IEEE USA Committee on Communications and Information Policy (CCIP), U.S. delegate to the European Society for Engineering in Medicine, U.S. delegate to the General Assembly of the IFMBE, IEEE delegate to the Public Policy Commission and the Council of Societies of the AIMBE, fellow of the AIMBE, senior member of IEEE, life member, Romanian Society of Clinical Engineering and Computing, life member, Biomedical Engineering Society of India, U.S. delegate to IFAC and IMEKO Councils in TC13. He was recently elected to the Administrative Board of the International Federation for Medical and Biological Engineering, a worldwide organization comprising 48

national members, overseeing global biomedical engineering activities. He was also elected to serve as the republications co-chairman of the Federation.

His contributions to the discipline have earned him numerous national and international awards. He is a fellow of the American Institute of Medical and Biological Engineering, a recipient of the IEEE 3rd Millennium Medal and a recipient of the Purkynje award from the Czech Academy of Medical Societies, a recipient of the Career Achievement Award, numerous outstanding accomplishment awards, and twice recipient of the IEEE EMBS distinguished service award. He can be reached at s.n.laxminarayan@ieee.org.

Index

- Absolute error, 492, 493f
- Accepted points, 208
- Acoustic shade, 5
- Active contour models, 512, 572, 589
- Active meta-cells, 385, 386, 388, 397, 403
- Active sets, 461
- Adaptive mesh refinement data structures, 382
- Additive operative splitting (AOS) schemes, 573
- Advanced segmentation techniques, 479–526.
 - See also* Fuzzy segmentation; Stochastic image models
- Advection (intrinsic) diffusion model, 591, 592, 609
- Advective part of the speed function, 206
- Adventitia, arterial, 22, 23, 31, 44, 47, 49f, 50f
- Affine (firm) thresholding, 320
- Affine-invariant denoising method, 392
- Algorithme à trous*, 312
- Aliasing, 156, 432
 - in phase images, 158–159
 - in speed images, 157
- Alzheimer's disease, 96
- Aneurysms, 174, 183, 188, 480, 520
- Anger camera, 98–99
- Angular resolution, IVUS, 15–17
- Anisotropic diffusion
 - deformable models and, 389, 391, 392, 393–394, 396, 401
 - level set segmentation and, 440, 441–442, 443–444, 447, 449
 - linear, 444, 447
 - planar, 444, 447
- Anisotropic speed function, 223–224
- Antisymmetry, 321
- AOS schemes. *See* Additive operative splitting schemes
- Arsenic-75, 60
- Arterialized venous (a-v) sampling method, 91–92
- Arterial spin tagging projection coronary magnetic resonance angiography (CMRA) technique, 178
- Artery reconstruction, 394–396
- Artifacts
 - inflow related, in MRA, 130–131
 - of IVUS catheter, 5
 - of IVUS transducers, 24
 - motion-induced, 180–181
 - Venetian blind, 166
- Artifacts scatterers, 24
- Asymmetric echo acquisitions, 137
- Attenuated coronary blood-myocardium in-flow contrast 3D coronary magnetic resonance angiography (CMRA), 192
- Attenuation coefficient, 36–37
- Attenuation correction, 78–83
- Autocorrected magnetic resonance angiography (MRA), 180–181
- Automatic seed initialization, 521–522, 523, 524–525
- Axial resolution, IVUS, 13–14, 15f
- Backward difference formula, 268, 271
- BACSPIN. *See* Breathing AutoCorrection with SPiral INTERleaves coronary magnetic resonance angiography
- Band-limited wavelets, 289, 297
- Barium fluoride, 85
- Batman image, 593, 595f, 596
- Bayes classifiers, 343, 489, 495
- Bayes' rule, 485
- BCFCM. *See* Bias corrected fuzzy c-means objective function
- Beam intensity, IVUS, 17, 18f

- Beam number, IVUS, 37–38
- Beam sweeping criterion, IVUS, 17–19
- Beam width, IVUS, 15–16
- Bell function, 317
- BFS algorithm. *See* Breadth-first-search algorithm
- BGO. *See* Bismuth germanate oxide
- Bias corrected fuzzy c-means (BCFCM)
 - objective function, 505–512, 513f, 514f
 - algorithm, 506–507
 - cluster prototype updating, 506
 - membership evaluation, 505
 - parameter estimation, 505
 - results, 507–512
- Bias fields, 504, 506
- Bicubic interpolation function, 212–214, 217–218, 227
- Binary trees, 210–212, 215, 315
- Biological volume datasets, 415–467. *See also*
 - Level set segmentation
 - multiple volume results, 438–439
 - segmentation from multiple nonuniform, 429–439
- Bismuth germanate oxide (BGO), 66–67, 78, 85
- Black blood magnetic resonance angiography (MRA), 168
- Blank scans, 81
- Bloch equations, 124
- Block detectors, 66, 85
- Blood
 - IVUS imaging and, 31, 50f
 - PET and, 91–92
- Blood flow. *See* Flow
- Blood pool contrast enhancement, 173
- Boltzmann constant, 482
- Boltzmann distribution, 482
- BONO. *See* Branch-on-need octree
- Born approximation, 10
- Brain
 - DT-MRI of, 440–449
 - MRI of, 507–512, 513f, 514f, 565f
- Brain stroke, 480, 521
- Brain tumors, 60, 514f
- Branch-on-need octree (BONO), 383–385
- Breadth-first-search (BFS) algorithm, 387
- Breast cancer, 96
- Breath-hold contrast enhanced magnetic resonance angiography (MRA), 177–178
- Breathing AutoCorrection with SPiRAL
 - INterleaves (BACSPIN) coronary magnetic resonance angiography (CMRA), 191
- Bright blood imaging, 167
- Brightness, 391–392
- Brightness error term, 281
- Brushlet(s), 314
 - attributes of, 316–319
 - spatial-temporal analysis using, 326–332
- Brushlet basis functions, 318
- B-spline functions, 325, 376, 503
- Bump function, 317
- Burnt nodes, 369–370, 371
- Calculated attenuation correction, 82–83
- Cancer
 - brain, 60, 514f
 - breast, 96
 - colon, 96
 - gallbladder, 186–187
 - head and neck, 96
 - liver, 96
 - lung (*see* Lung cancer)
 - pancreatic, 96
 - PET and, 58, 75, 96
 - thyroid, 96
- Canny edge detectors, 341, 374, 423, 432, 436
- Capacity function, 603
- Carbon-11 (^{11}C), 62t, 96
- Cardiac-gated 2D phase contrast magnetic resonance angiography (PC MRA), 151–153
- Cardiac-triggered free-breathing 3D balanced fast-field echo projection magnetic resonance angiography (MRA), 182
- Cardiac ultrasound, 341–342
- Carotid bifurcations, 138–140
- Carotid siphons. *See* Vessel turns
- Cartesian coordinates, 259
- Catheters, IVUS, 3–4, 5, 6
- CE-MRA. *See* Contrast-enhanced magnetic resonance angiography
- Centered difference formula, 268
- Central and forward difference method, 271
- Central difference method, 271
- Central slice theorem, 72
- Cerebral arteriovenous malformations (AVMs), 187
- Cerebral ischemia, 176
- Cervical magnetic resonance angiography (MRA), 182–185
- CFL. *See* Courant-Friedrichs-Levy restriction
- Chain rule, 418
- Characteristic function, 368
- Children of nodes, 210–212, 315, 316, 385
- Chromaticity, 391–392
- Chromosomes, 489–490
- Cine gradient-echo technique, 176
- Cine ventriculography, 123
- Circle of Willis, 181
- Clique(s), 483, 484

- Clique coefficients, 490
- Cluster prototype updating, 502, 506
- CMRA. *See* Coronary magnetic resonance angiography
- Coarse to fine procedures, 344, 345
- Coincidence detection, 63–65, 78, 80. *See also* Detectors, PET
- Coincidence resolving time, 64–65, 67, 78
- Colon cancer, 96
- Color images, RAGS and, 555–557, 558, 560f, 561, 566–571, 572, 573
- Combinatorial manifolds, 368
- Complex flow, 162–163
- Compton scattering, 77
- Computed tomography (CT), 57–58, 79, 96, 100, 363, 479
 - geometric snakes and, 541
 - level set segmentation and, 416, 429, 451, 467
 - lung segmentation and, 480, 482, 491–498
 - PET combined with, 101–102
- Conjugate mirror filters (CMF), 310
- Consoles, IVUS, 3, 4f
- Constructive Solid Geometry (CSG) model, 422, 432
- Continuation algorithms, 379–380
- Continuous wavelet transform, 307–309
- Contrast-enhanced magnetic resonance angiography (CE-MRA), 173–174, 184
 - breath-hold, 177–178
 - collateral flow assessment with, 179
 - FLASH, 181
 - multiphase, 181
 - ultrashort, 187
- Contrast to noise ratio signal (CNRS), 30, 44, 46t, 48f, 192
- Control points, 344
- Convolution-backprojection reconstruction method, 73
- Convolution methods, 77
- Copper-64, 60
- Coronary magnetic resonance angiography (CMRA), 188, 193
 - attenuated coronary blood-myocardium in-flow, 192
 - BACSPIN, 191
 - 3D real-time navigator, 190
 - free-breathing 3D, 191
 - motion-adapted gating window in, 191–192
 - ROC analysis, 189–190
- Coulombic attractive forces, 61
- Courant-Friedrichs-Levy (CFL) restriction, 517–518
- Co-volume level set method, 583–620
 - numerical results, 609–616
 - related mathematical models, 587–597
 - semi-implicit, 586, 598–609
- Crisp segmentation, 480
- Cross-scale regularization
 - for images with low SNR, 335–338
 - for tomographic images, 332–338
- CSG model. *See* Constructive Solid Geometry model
- CT. *See* Computed tomography
- Curvelets, 325
- Data cache, 386–387
- Daubechies scaling function, 294
- Daubechies wavelets, 289, 293, 297
- DBC. *See* Differential backscattering cross-section
- Dead-time correction, 76–77
- Decomposition
 - brushlet, 327
 - tetrahedral, 367, 368
 - wavelets and, 315, 317, 324
- Decomposition filters, 310, 311–312
- Deconvolution methods, 77
- Deformable models, 359–406. *See also* Level set deformations; Snake models
 - background to, 364–371
 - diffusion model convergence with, 388–392
 - discrete, 367
 - experimental results, 393–399
 - free-form, 364–365, 455–456
 - initialization of, 361–363, 364, 367, 371–377, 401
 - reconstruction method for, 381–383
- Deformable surface models, 363, 367
- Degree of freedom (DF) nodes, 601–602, 606, 607
- Delayed coincidence detection method, 78
- 3D-ELLIP. *See* Three-dimensional elliptical centric view ordering
- Delta functions, 230, 235, 518–519, 520
- Denoising
 - affine-invariant method, 392
 - deformable models and, 392, 401
 - state-of-the-art and applications, 326–338
 - thresholding operators for, 319–320, 325–326, 336
 - threshold value selection and, 323–324
 - time inclusion in, 328–332
 - wavelet transforms and, 306, 316, 319–338, 345
- Density gradient estimation, 557
- Density parameter estimation, 456–457
- Depth, of SPS models, 266
- Depth maps, 274, 275, 278, 285, 286
- Derin-Elliot model, 484

- Detectors, PET
 - description of, 66–67
 - events detected with, 67–69
 - normalization in, 76
- Differences of normals method, 460–461
- Differential backscattering cross-section (DBC), 10, 12, 31–32, 33f, 34f
- Diffused region forces, 546, 550, 571–572
 - numerical solutions for, 553
 - properties of, 547–549
 - weak-edge leakage and, 561, 562f
- Diffusion models, 388–392
- Diffusion-reaction equation, 363
- Diffusion tensor magnetic resonance imaging (DT-MRI), 440–449, 450f
 - geometric modeling and, 444–445
 - segmentation and, 445–449
 - tensor invariants in, 442, 443–444
- Diffusion-weighted imaging, 174
- Diffusive part of the speed function, 206
- Digital subtraction magnetic resonance angiography (DSA MRA), 172, 173–174, 187
- Dilation, 309, 373, 374
- Direct method, for SPS models, 267
- Dirichlet boundary conditions, 583, 596, 602, 605, 607, 610, 612
- Discrete deformable models, 367
- Discrete sampling, 308
- Discrete search space, 362
- Discrete wavelet transforms, 309–313
 - dyadic, 311–313
 - orthogonal, 309–310
- Distant points, 208
- Divergence theorem, 603
- Dominant Gaussian components extracting algorithm, 486–487, 492
- Down-sweep process, 211
- DSA MRA. *See* Digital subtraction magnetic resonance angiography
- DT-MRI. *See* Diffusion tensor magnetic resonance imaging
- 3D-TRICKS. *See* Three-dimensional time-resolved imaging of contrast kinetics
- Dual active contour (ACM), 361–362
- Dual T-snakes model, 362
- Dynamic programming (DP), 362
- Dynamic-range compression, 151
- Echo amplitude, 7, 25
- Echocardiography, 615–616
- Echogram generation
 - 1D, 24–26
 - 2D, 26–28
- Echo time (TE), 118, 120, 133
 - flow-related enhancement and, 128
 - high-resolution MRA and, 182
 - INFLOW method and, 170
 - shorter, 162, 170
 - slice-transition phenomenon and, 126
 - spin phase phenomenon and, 132
 - TOF MRA and, 135, 139, 142, 144, 145
 - variable, 182
- Edge crispening, 321
- Edge definition, 390–391
- Edge detection and detectors
 - Canny, 341, 374, 423, 432, 436
 - deformable models and, 366
 - level set segmentation and, 423, 431–432, 434, 436
 - RAGS and, 555
 - wavelet, 333–334, 336, 339–342
- Edge points, 341, 389
- Elastic net algorithm, 376
- Electron capture, 61
- Elliptic solvers, coupling to, 230–240
- EM algorithm. *See* Expectation maximization algorithm
- Embedding function, 201, 202, 207
- Emission computed tomography, 59. *See also*
 - Positron emission tomography;
 - Single-photon emission computed tomography
- Empirical approaches to scatter correction, 77
- Energy functions
 - Gibbs random field and, 483
 - SPS models and, 260, 264, 281–282
 - wavelet-based methods and, 293, 294, 295
- Enhancement operators, 320–323, 336
- ENO method, 206
- Enrichment functions, 232–233
- Entropy condition, 202, 369, 371, 372, 554, 555
- Entropy-satisfying schemes, 574–575
- Entry slice phenomenon, 127–128, 130
- Epanechnikov kernel, 557
- Epilepsy, 96
- Erosion, 373, 374
- Error
 - absolute, 492, 493f
 - IVUS and, 5–6
 - level set segmentation and, 453, 455
 - lung CT and, 492, 493f
 - maximum radial, 563, 565f
 - sum square, in IVUS, 35–36, 44, 46t
 - total squared brightness, 265
- Error sinograms, 456
- Euclidean curves, 537
- Euclidean distance functions, 434, 502, 554
- Euclidean metric, 538–539
- Euclidean space, 373, 557
- Euler equations, 264–265, 267, 282

- Eulerian methods, 241, 370
- Euler-Lagrange equations, 365, 392, 539
- Expectation maximization (EM) algorithm, 74, 492, 495
 - fuzzy segmentation and, 503–504, 507–509
 - maximum likelihood, 74–75
 - ordered subsets, 75
 - stochastic image models and, 486, 487–488
- Extended finite element method (X-FEM), 231–235, 239, 240
- External forces
 - deformable models and, 359, 369
 - RAGS and, 550, 552
- Face recognition, neural nets for, 375
- Fast marching method, 207–214, 215, 216, 217, 243, 298
 - applications of, 242
 - geometric snakes and, 573
 - initialization of, 209, 212–214
 - level set segmentation and, 418
 - novel extension of, 223–225
 - velocity extensions and, 226
- Fast spoiled gradient echo (SPGR), 186
- Fat-suppressed three-dimensional magnetic resonance angiography (MRA), 186
- FBP. *See* Filtered backprojection
- FCM algorithm. *See* Fuzzy c-means algorithm
- FDG. *See* [^{18}F]fluorodeoxyglucose
- FDM. *See* Finite difference method
- Feature points, 344
- [^{18}F]fluorodeoxyglucose (FDG), 87–88, 93
- [^{18}F]fluorodeoxyglucose-6-phosphate (FDG-6-P), 88
- Filter bank, 309–313, 319
- Filtered backprojection (FBP), 72–73, 75, 85, 86
 - introduction of, 60
 - level set segmentation and, 451–452, 462, 463–465, 466
 - wavelet transforms and, 333, 334f, 335
- Filters and filtering
 - conjugate mirror, 310
 - decomposition, 310, 311–312
 - Gabor, 327
 - Gaussian, 28, 30, 447
 - Hann, 333
 - high-pass, 310
 - homomorphic, 503
 - Lemarié-Battle, 338, 339
 - linear, 421
 - low-pass, 310, 333, 421, 422
 - median, 30
 - morphological, 421–422
 - ramp, 85, 333
 - reconstruction, 311–312
 - steerable, 327
 - Wiener, 332
- Finite difference method (FDM), 260, 366, 371, 606
 - convergence of, 271
 - formula and examples, 267–271
- Finite elements methods, 366
- First-order gradient moment nulling, 160–161
- Fitness function, 490
- FLAG. *See* Flow adjusted gradient
- Flame propagation, 201, 202
- FLASH contrast-enhanced magnetic resonance angiography (CE-MRA), 181
- Flip angle, 137, 143, 144, 164–165, 166
- Flow
 - collateral, assessment of, 179
 - complex, 162–163
 - laminar, 121–123, 124, 125
 - physical principles of, 124–134
 - plug, 122, 124, 125
 - principles of, 117–124
 - pulsatile, 121–123
 - turbulent, 121–123
- Flow adjusted gradient (FLAG), 169, 171–172
- Flow compensation (FC), 160
 - PC, 162
 - phase/frequency, high-resolution MRA with, 182
 - TOF MRA and, 134, 142, 144–145
- Flow eddies, 167
- Flow encoding gradients, 146, 155
- Flow phase, 146
- Flow-related enhancement, 128–129
- Flow velocity, 121–123, 136, 145–146
- Flow void, 147
- Fluorine-18 (^{18}F), 62, 96
- Forward difference formula, 268, 269, 271, 274, 277
- Fourier-based image reconstruction, 71–73
- Fourier transforms, 450, 451. *See also* Inverse Fourier transforms
 - brushlets and, 317–318
 - SPS models and, 267, 273, 274, 275, 276, 277, 286–287
 - wavelets and, 290, 306–307, 308, 314
 - windowed, 306–307, 308
- Fractional volume segments, 124
- Free-breathing 3D coronary magnetic resonance angiography (CMRA), 191
- Free-form deformations, 364–365, 455–456
- Fresnelets, 325
- Frog embryo, MRI scan of, 427–428
- Fully-discrete semi-implicit co-volume scheme, 604
- Function, defined, 58

- Fuzzy c-means (FCM) algorithm, 480, 499–500, 507–512, 513f, 514f. *See also* Bias corrected fuzzy c-means objective function; Modified fuzzy c-means objective function
 - Fuzzy logic, 480
 - Fuzzy segmentation, 480, 498–512
 - Gabor coefficient spectral signatures, 338
 - Gabor functions, 316
 - Gabor oriented filters, 327
 - Gabor transforms, 314–315, 338
 - Gabor wavelets, 314–315, 318, 344
 - Gadolinium (Gd) chelate, 165
 - Gadolinium (Gd)-enhanced magnetic resonance angiography, with MRC, 186–187
 - Gadolinium (Gd)-enhanced three-dimensional magnetic resonance angiography (MRA), 179–180
 - Gadolinium (Gd) oxyorthosilicate, 85
 - Gain fields, 504–505
 - Galerkin procedures, 231
 - Gallbladder carcinoma, 186–187
 - Gallium-68 (^{68}Ga), 62t
 - Gamma camera, 98–99
 - Gamma rays, 60, 63, 65, 66–67, 80, 81, 85
 - Gaussian blurring, 389
 - Gaussian components, 485–487, 488, 495
 - Gaussian components classification algorithm, 488–489
 - Gaussian components extracting algorithm, dominant, 486–487, 492
 - Gaussian curvature, 458
 - Gaussian diffusion, 390, 393–394
 - Gaussian distributions, 521
 - Gaussian filters, 28, 30, 447
 - Gaussian functions, 314, 325, 334, 584
 - Gaussian kernels, 421, 423, 424
 - Gaussian noise, 323, 453, 455, 457, 507, 510f, 511f
 - Gaussian surfaces, 524
 - Gauss-Newton type algorithm, 90
 - Gauss-Seidel algorithm, 607
 - Generalized linear-least squares (LLS) methods, 94
 - Genetic algorithm (GA), 489–490, 495
 - Geometric GGVF snake, 536, 572, 573
 - experiments and results, 559–571
 - properties of, 544–547
 - Geometric modeling, 444–445
 - Geometric (geodesic) snakes, 359–360, 535–576. *See also* Region-aided geometric snake
 - examples of, 541
 - experiments and results, 559–571
 - properties of, 537–541
 - shortcomings of, 536, 542–544
 - Ghost point method, 238–240
 - Gibbs Markov model, 485, 489, 495
 - Gibbs random fields (GRF), 481, 482–483
 - Global threshold, 323
 - Glucose metabolism, 58, 93
 - Godunov's method, 205, 206–207, 208
 - Gradient flow forces, 546, 547
 - Gradient inversion method, 119
 - Gradient moment rephasing, 167
 - Gradient-recalled acquisition, 164
 - Gradient recalled echo (GRE), 118, 120, 164
 - blood pool contrast enhancement and, 173
 - 2D Fourier transform, 130
 - flow-related enhancement and, 129
 - inflow related artifacts and, 130
 - multislice techniques, 165–166
 - slice-transition phenomenon and, 126, 127
 - spin phase phenomenon and, 132
 - TOF MRA and, 134
 - Gradient vector flow (GVF), 363, 389, 390, 400, 401, 403, 544–546. *See also* Geometric GGVF snake
 - Graphical techniques, in PET, 93
 - Gray-level images
 - deformable models and, 366, 374, 389, 391, 396–397
 - fuzzy segmentation and, 498
 - gradient flow force and, 547
 - intensity models and, 479
 - IVUS and, 5, 6, 7, 25f, 26, 29, 30, 31, 39–40, 41, 42–48, 49f, 50f, 51
 - level set segmentation and, 417, 421, 424, 447, 462
 - in lung CT, 491
 - RAGS and, 555, 561, 563–566
 - SPS models and, 257
- GRE. *See* Gradient recalled echo
- GRF. *See* Gibbs random fields
- Grid points
 - fast marching method and, 208–209, 212, 213–214
 - as ghost points, 238
 - immersed interface method and, 235–236, 237
 - level set method and, 215–216, 217–218
 - ordered upwind method and, 225
- Griffin dataset, 438
- GVF. *See* Gradient vector flow
- Hamilton-Jacobi equations, 205, 207, 418
- Handwritten digit recognition tasks, 375, 376–377
- Hann filters, 333
- Hann windows, 73, 85

- Hard thresholding, 320
- Head and neck cancer, 96
- Heap-sort algorithm, 210–212, 215, 217, 218
- Heart, MRI of, 566f, 567f
- Heaviside step, 518–519
- Heisenberg rectangles, 307, 308
- Hessian matrix, 433, 436
- Hierarchical matrix decomposition technique, 272, 284
- High frequency intravascular ultrasound (IVUS), 34, 35f
- High-level processes, 484
 - lung CT and, 491
 - parameter estimation for, 489–490
- High-pass filters, 310
- High-resolution magnetic resonance angiography (MRA) with phase/frequency flow compensation, 182
- High-velocity loss, 126–127
- Homomorphic filtering, 503
- Horn image irradiance equation, 258–259
- Hough transforms, 363
- Huntington's disease, 96
- Hyperbolic conservation laws, 202, 205
- Hyperbolic space, curvature flow in, 220
- Image acquisition
 - in 2D PC MRA, 150–151
 - in 3D PC MRA, 153–154
- Image-based feature attraction term, 431
- Image contrast
 - in PC MRA, 154
 - in TOF MRA, 135–136
- Image enhancement, 319–338
- Image irradiance equation. *See* Irradiance equation
- Image models, 479, 484–485
- Image parameters, 143–145
- Image reformation, 163–165
- Image registration, 343–344
- Image resolution, 12–13
- Image segmentation. *See* Segmentation
- Immersed interface method, 235–238, 239–240
- Implicit models, 359, 360
- Incremental projection updates (IPU), 461–462
- INFLOW method, 168, 169–171
- Initialization. *See also* Reinitialization
 - automatic seed, 521–522, 523, 524–525
 - of deformable models, 361–363, 364, 367, 371–377, 401
 - of the fast marching method, 209, 212–214
 - of geometric GGVF snake, 561
 - of level sets, 218, 227–228, 416, 420, 421–422, 424–425, 432, 436–437, 447, 448, 462, 521–522, 523, 524–525
 - mathematical morphology for, 373–375
 - region-based approaches, 371–373
 - of T-surfaces, 371–381
- Insert neighbors, 387
- Integrability, 265–266
- Integrability term, 281
- Integral transforms, 267
- Integrated projection, 95
- Intensity gradient term, 281
- Intensity inhomogeneities, 480, 502, 503, 504, 508, 511f, 513f
- Intensity models, 479
- Interface, 201–202, 217–218
 - difficulty with level set method, 203–204, 242
 - elliptic solvers and, 230
 - fast marching method and, 207
 - immersed method, 235–238, 239–240
 - particles and, 240–241
 - velocity extensions and, 225–230
 - X-FEM and, 233
- Intermediate frequency intravascular ultrasound (IVUS), 34, 35f
- Intermediate value theorem, 268
- Internal forces
 - deformable models and, 359, 369
 - RAGS and, 550
- Interval trees, 384, 386, 387, 397, 398, 399, 403
- Intima, arterial, 22, 23, 31, 45–47, 49f, 50f
- Intima/media transition, 30, 36, 38, 42, 44
- Intravascular ultrasound (IVUS), 1–52
 - angular resolution in, 15–17
 - axial resolution in, 13–14, 15f
 - basic image model, 7–8
 - beam intensity, 17, 18f
 - beam number influence, 37–38
 - beam sweeping criterion, 17–19
 - !D echogram generation in, 24–26
 - 2D echogram generation in, 26–28
 - final image processing in, 28
 - formal definition of image model, 9–12
 - image resolution in, 12–13
 - image simulation in, 23–28
 - limitations of technique, 5–6
 - need for data generation model, 6–7
 - optimal attenuation coefficient in, 36–37
 - optimal frequency in, 35–36
 - polar real *vs.* simulated images, 41–48
 - real *vs.* simulated images, 29, 35–36, 38–40
 - scatterer number of arterial structures and, 19–23
 - spatial resolution in, 32–34
 - technical aspects of, 3–4
 - validation of image simulation model, 29–50
- Intravoxel incoherence. *See* Phase dispersion

- Inverse Fourier transforms
 - brushlets and, 318
 - SPS models and, 273, 274, 275, 276, 277, 286, 287
- IPU. *See* Incremental projection updates
- Irradiance equation, 258–259, 261–264, 267, 279, 282
 - FDM and, 268, 270–271
 - Tsai-Shah's method and, 277
- Isointensity surface reconstruction method, 521
- Isosurface(s), 424, 425
- Isosurface generation methods, 379–380
- Isotropic diffusion, 444
- Iterative-based image reconstruction, 71, 73–75
- Iterative formula, 269, 271, 282–283, 296, 372
- IVUS. *See* Intravascular ultrasound
- Jacobi method, 270, 271
- Kanizsa, triangle of, 613
- Kawasaki disease, 188
- K-d-tree subdivision schemes, 379
- Kety-Schmidt one-compartment model, 93
- K-layers, 31, 32
- K-mean classifier, 339
- K-space three-dimensional navigator-gated magnetic resonance angiography (MRA), 178
- Lagrange equations, 366
- Lagrange multipliers
 - fuzzy segmentation and, 501, 505
 - SPS models and, 264–265, 266
 - wavelet-based methods and, 297
- Lagrangian methods, 201, 241, 589
- Lambertian reflectance, 262, 273
- Lambertian surfaces, 259, 260, 263–264
- Laminar flow, 121–123, 124, 125
- Laplacian operators, 548
- Larmor frequencies, 118, 124, 153
- Least recently used (LRU) meta-cell, 387
- Lemarié-Battle filters, 338, 339
- Level-dependent threshold, 323
- Level set(s), 382
 - curve evolution with, 516–517
 - geometric snakes and, 573
 - RAGS and, 551–552, 554–555, 574–575
- Level set deformations, 370–371, 401, 416, 448, 455–456, 512–515. *See also* Level set surface deformations
- Level set equation, 583, 587
- Level set function representation, 515
- Level set method, 201–243. *See also* Co-volume level set method
 - applications of, 242
 - basic, 203–222
 - basic algorithm assembly, 219
 - elliptic solvers and, 230–240
 - example calculations, 220–222
 - numerical implementation of, 205–207
 - particle, 240–241
 - recent developments in, 222–241
- Level set representation, 203–205, 551–552
- Level set segmentation, 415–467, 480, 512–525.
 - See also* Biological volume datasets
 - algorithm overview, 436–437
 - density parameter estimation, 456–457
 - framework for, 420–428
 - implementation of, 459–462
 - from multiple nonuniform datasets, 429–439
 - numerical schemes for, 460–461
 - quality measurement in, 523
 - sparse-field method in, 420, 461
 - stability and CFL restriction, 517–518
 - surface estimation in tomographic data, 450–467
 - tracking the front, 518–519
 - volume segmentation algorithm in, 523
- Level set surface deformations, 417, 420, 422–425, 429, 452. *See also* Level set deformations
- Level set surface models, 417–420
- Levy distance, 489, 492, 495
- Limited-angle tomography, 452
- Linear anisotropic diffusion, 444, 447
- Linear approaches
 - to SPS models, 273–279
 - velocity extensions and, 228–229
- Linear filtering, 421
- Linearization approaches, in PET, 93–94
- Linear least-squares (LLS) methods, 93, 94
- Line integral, 69, 72
- Lipschitz continuous boundary, 583
- Liver cancer, 96
- Local approach, to SPS models, 272
- Local mass balance property, 603
- Local maxima, 341, 342f, 343, 389, 536, 563
- Local minima, 542
- Local scale property, 378
- Local solutions, 421
- Logan plot, 93
- Longitudinal relaxation times (T1), 124, 135, 137, 163, 164
- Low frequency intravascular ultrasound (IVUS), 34, 35f
- Low-level processes, 484
 - lung CT and, 491
 - parameter estimation for, 485–489
- Low-pass filters, 310, 333, 421, 422

- LSO. *See* Lutetium oxyorthosilicate
- Lumen, arterial, 23, 30, 44, 49f
- Lumen/intima transition, 30, 36, 38, 42, 44
- Lunar surface, 258
- Lung cancer, 490–498
 CT of, 480, 482, 491–498
 PET of, 96
- Lung extraction, 491–498
- Lutetium oxyorthosilicate (LSO), 85
- Lymphoma, 96
- Magnetic resonance angiography (MRA), 117–193
 acquisition methods, 169–172
 arterial spin tagging technique in, 178
 autocorrected, 180–181
 black blood, 168
 blood pool contrast enhancement and, 173
 cardiac-triggered free-breathing 3D, 182
 cervical, 182–185
 contrast-enhanced (*see* Contrast-enhanced magnetic resonance angiography)
 coronary (*see* Coronary magnetic resonance angiography)
 with diffusion-weighted imaging, 174
 digital subtraction, 172, 173–174, 187
 3D-navigator echo, 175–176
 Doppler flow quantification and, 176–177
 fat-suppressed 3D, 186
 Gd-enhanced 3D, 179–180
 Gd-enhanced with MRC, 186–187
 high-resolution, 182
 k-space 3D navigator-gated, 178
 level set segmentation of, 480, 520–525
 limitations and future of, 192–193
 magnetization transfer (*see* Magnetization transfer magnetic resonance angiography)
 phase contrast (*see* Phase contrast magnetic resonance angiography)
 with phase/frequency FC, 182
 quadruple contrast enhancement with, 187–188
 recent advances in, 172–192
 sensitivity encoding and, 172–173
 techniques and principles of, 134–168
 time-of-flight (*see* Time-of-flight magnetic resonance angiography)
- Magnetic resonance cholangiography (MRC), 186–187
- Magnetic resonance imaging (MRI), 96, 467, 479
 diffusion tensor (*see* Diffusion tensor magnetic resonance imaging)
 fuzzy segmentation of, 480, 499, 502–512
 level set segmentation of, 416, 424–425, 427–428, 429, 430, 438–439
 RAGS and, 565f, 566f, 567f
 wavelets in, 305, 339, 340f, 345
- Magnetic resonance spectroscopic imaging (MRSI), 187–188
- Magnetization, principles of, 117–124
- Magnetization transfer magnetic resonance angiography (MT MRA)
 with RF labeling technique, 184–185
 TOF, 167
- Magnitude image, 146
- Mallat algorithms, 291, 297
- Mammograms, 322, 343, 616
- MAP estimation. *See* Maximum *a posteriori* estimation
- Marching cubes, 380, 422, 436, 445, 446f, 447, 450f
- Markov models, 342
- Markov random fields (MRF), 339, 481, 483–484
- Maximum *a posteriori* (MAP) estimation, 452, 457, 485, 489
- Maximum intensity projection (MIP), 157, 163
 black blood MRA and, 168
 level set segmentation and, 466
 for stenoses, 190
 TOF MRA and, 140–141, 142
- Maximum likelihood-expectation maximization (ML-EM) algorithm, 74–75
- Maximum-likelihood (ML) methods, 452
- Maximum radial error (MRE), 563, 565f
- Mean curvature flow, 202
 co-volume level set method and, 583, 585, 594–596, 619
 in flame propagation, 201
 in hyperbolic space, 220–222
 level set segmentation and, 458–459, 460
 level set surface deformation and, 423
 minimal surfaces and, 220
 Riemannian, 585, 619
- Mean shift algorithm, 536, 546, 557–559, 574
- Media, arterial, 22, 23, 31, 47, 49f, 50f
- Media/adventitia transition, 30, 36, 38, 42, 44
- Median filters, 30
- Melanoma, 96
- Memory utilization, 363, 383–388, 397, 399, 402, 403
- Meshes
 co-volume level set method and, 601–602, 605
 deformable models and, 367, 368–369, 371, 401, 402, 417
 elliptic solvers and, 230
 fast marching method and, 207–208, 213–214
 level set segmentation and, 519
 RAGS and, 553
 SPS models and, 268
 X-FEM and, 233, 240

- Meta-cell partition, 384–385, 386
- Meta-cell technique, 364, 383, 384–385, 397–398, 399, 402, 404
 - active meta-cells in, 385, 386, 388, 397, 403
 - elements of, 386–387
- Meta-intervals, 384–385, 386, 387
- Metropolis algorithm, 489, 490, 495, 497
- Meyer wavelets, 289, 297
- MFCM. *See* Modified fuzzy c-means objective function
- Minimal curvature, of SPS models, 266
- Minimal surfaces with voids, 220
- Minimax threshold, 323–324
- Minimization approach, to SPS models, 272
- Minimum-maximum principle, 605
- MIP. *See* Maximum intensity projection
- ML-EM algorithm. *See* Maximum likelihood-expectation maximization algorithm
- MLS methods. *See* Moving least-squares methods
- M-matrix, 605
- Model elements, 369
- Modified fuzzy c-means (MFCM) objective function, 500–505
 - cluster prototype updating, 502
 - membership evaluation, 501
 - parameter estimation, 500–502
 - signal modeling, 504–505
- Moments, 161–162
- Monotonicity, 321
- Morphological chains, 374
- Morphological filtering, 421–422
- Morphological principle, 587
- Morphological propagators, 421
- Mother wavelets, 289, 290
- Motion-adapted gating window, 191–192
- Motion-induced artifacts, 180–181
- MOTSA. *See* Multiple overlapping thin-slab acquisition
- Mouse embryo, MRI scan of, 424–425, 430, 438–439
- Moving least-squares (MLS) methods, 429–430, 433–436, 467
- MRA. *See* Magnetic resonance angiography
- MRC. *See* Magnetic resonance cholangiography
- MRF. *See* Markov random fields
- MRI. *See* Magnetic resonance imaging
- MRSI. *See* Magnetic resonance spectroscopic imaging
- MT MRA. *See* Magnetization transfer magnetic resonance angiography
- Multidimensional discrete dyadic wavelet transform, 312, 313
- Multigrid method, 272
- Multilayer perception (MLP), 375–376
- Multiphase contrast-enhanced magnetic resonance angiography (CE-MRA), 181
- Multiple coincidences, 68f, 69
- Multiple energy window techniques, 77
- Multiple overlapping thin-slab acquisition (MOTSA), 165, 166
- Multiple sclerosis, 479
- Multiresolution approximation/analysis, 289, 291–292
- Multiscale envelope features, 339
- Multiscale methods
 - SPS models and, 271–272
 - in texture classification, 338–339
 - wavelet transform and, 306–319, 326–338
- Multislice gradient recalled echo (GRE) techniques, 165–166
- Multivariate kernel density estimate, 558
- Myocardial perfusion, 99–100
- Nal(Tl) crystals, 60, 66, 85
- Narrow-band techniques, 420, 436, 437, 519–520
 - applications of, 218–219
 - deformable models and, 401–402, 403
 - geometric snakes and, 573
- Near-diagonalization of signal and noise, 315–316
- Neighborhood effect, 480, 484, 499, 507, 512
- Neighboring weak/strong edges, 562
- Neumann boundary conditions, 588
- Neural networks, 298, 363, 371, 375–377
- Neural networks classifiers, 339
- Neutrinos, 61
- Neutrons, 61
- Newton's method, 214, 227, 273, 277, 278, 287, 586
- Nitrogen-13 (^{13}N), 62t, 96
- NLLS. *See* Nonlinear least-squares method
- Node elements, 369
- Noise. *See also* Contrast to noise ratio signal; Denoising; Signal-to-noise ratio
 - co-volume level set method and, 585, 587, 589, 597, 612
 - deformable models and, 376, 390, 393, 400, 401
 - fuzzy segmentation and, 480, 499, 500, 504, 508, 512
 - Gaussian, 323, 453, 455, 457, 507, 510f, 511f
 - geometric snakes and, 536, 542
 - level set segmentation and, 415–416, 421, 430, 432, 445, 447, 451, 452, 453, 455, 457, 458
 - near-diagonalization of signal and, 315–316

- RAGS and, 536, 546, 561, 562–563, 564f, 571, 572
- salt and pepper, 480, 499, 500, 514f, 587
- speckle, 329–331
- stochastic image models and, 481
- wavelets and, 315–316, 341, 342, 343
- Non-degree of freedom (NDF) nodes, 601, 602
- Nonlinear least-squares (NLLS) method, 90, 92
- Non-negative garrote thresholding, 325
- Non-negative least squares (NNLS) algorithms, 94
- Nonocclusion, 262
- Nonseparable wavelets, 297
- Nontissue scatterers, 24
- Nucleons, 61
- Numerical flux functions, 202, 205–207, 208
- Object characteristic function, 362, 378, 381, 388
- Octree subdivision schemes, 379
- ODESSA. *See* Oscillating dual-equilibrium steady-state angiography
- One-dimensional orthogonal wavelet theory, 288–291
- Optimization approaches, 279–288
- Ordered subsets expectation maximization algorithm (OS-EM), 75
- Ordered upwind methods, 223–225
- Orthogonality boundary condition, 220
- Orthographic projection, 262
- Oscillating dual-equilibrium steady-state angiography (ODESSA), 185–186
- OS-EM. *See* Ordered subsets expectation maximization algorithm
- Out-of-core isosurface extraction techniques, 363–364
 - for memory utilization, 383–385
 - for segmentation, 386–388, 396–399, 402, 404
- Oversegmentation color region maps, 566–568, 569f, 570f
- Oxygen-15 (^{15}O), 62t, 96
- Pancreatic cancer, 96
- Panel-clustering method, 272
- Parametric imaging, 92–93
- Parametric snakes, 359, 360, 361, 535, 555
- Parent nodes, 210–212
- Parkinson's disease, 96
- Partial derivatives, computing, 433–436
- Partial differential equations (PDE)
 - co-volume level set method and, 610
 - level set segmentation and, 416, 421, 459, 461, 517, 519, 521
 - linear, 259, 260, 279
 - nonlinear, 259, 261, 263, 264
 - numerical methods for, 267–272
- RAGS and, 536, 546
- SPS models and, 259, 260, 261, 263, 264, 267–272, 279
- Particle level set method, 240–241
- Particle model of the deformable surface, 360–361, 367–368
- Patlak plot, 93
- PDE. *See* Partial differential equations
- Pentland's algorithm, 273–276, 277, 278, 279, 285, 286–287
- Perona-Malik function, 584, 588, 589
- PET. *See* Positron emission tomography
- PETT II. *See* Positron emission transaxial tomography
- PETT III. *See* Whole-body positron computed tomograph
- Phantoms
 - denoising on, 328–332
 - fuzzy segmentation and, 503, 507, 508, 509, 510f, 511f
 - level set segmentation and, 522, 523
- Phase-contrast Doppler flow quantification, 176–177
- Phase contrast magnetic resonance
 - angiography (PC MRA), 145–156, 524. *See also* Flow adjusted gradient; Rapid sequential excitation
 - 2D, 134, 146, 147–153
 - 3D, 134, 146, 153–156
 - 4D, 189
 - ODESSA compared with, 186
- Phase dispersion, 159–160, 161–162
- Phase images, 146, 158–159
- Photomultiplier tubes (PMTs), 66–67
- Photons, 60
- PIT. *See* Postinjection transmission scanning
- Pivoting, 380
- Pixels
 - co-volume level set method and, 600, 601, 605, 609–610
 - fuzzy segmentation and, 498–499, 500
 - IVUS, 30
 - Markov random fields and, 483, 484
- Planar anisotropic diffusion, 444, 447
- PL generation algorithms, 380, 381
- PL manifold extraction, 381
- Plug flow, 122, 124, 125
- Point spread function (PSE), 182
- Polar images, 41–48
- Positron(s), 60, 63
- Positron annihilation, 60, 62–63, 84. *See also* Coincidence detection

- Positron emission, 61–62, 63
- Positron emission tomography (PET), 57–102
 - applications of, 95–97
 - brief history of, 59–60
 - calibration in, 83
 - compartmental model fitting in, 89–90
 - data acquisition in, 69–71
 - data corrections in, 76–83
 - image reconstruction in, 71–75
 - input function in, 90–92
 - modes of decay in, 61–62
 - physiological parameter estimation in, 86–95
 - resolution limitations of, 83–86
 - wavelet transforms and, 326, 332–338
- Positron emission transaxial tomography (PETT II), 60
- Positronium, 62
- Positron range, 84
- Postinjection transmission (PIT) scanning, 83
- Preprocessing steps, 363, 371, 397, 399, 402, 416
- Presaturation pulses, 130–131, 137–138, 140
- Principal component analysis (PCA), 298
- Processing list, 387, 403
- Projection method, for SPS models, 267
- Prompt circuits, 78
- Prompt events, 65, 67
- Propagation approach, to SPS models, 272
- Proton(s), 61
- Proton-rich isotopes, 61
- Pseudo-Gibbs phenomena, 324
- Pull-back unit, IVUS, 3–4
- Pulsatile flow, 121–123
- Pulse duration, 14

- Quasi-isosurfaces, 424
- Query algorithm, 387

- Radial heat equation, 232–233, 234f
- Radial scatterer distributions, 31, 33f
- Radiofrequency (RF) coils, 480, 502
- Radiofrequency (RF) labeling technique, 184–185
- Radioisotopes, 59
- Radionuclides, 60
- Radon transforms, 71–72, 81, 332, 450
- RAGS. *See* Region-aided geometric snake
- Ramp filters, 85, 333
- Ramp function, 317
- Random coincidences, 68–69, 78
- Random field models, 480
- Randoms correction, 78
- Rapid sequential excitation (RSE), 169, 171–172
- Rate constants, 88
- Real-time three dimensional (RT3D) cardiac ultrasound, 326–332
- Receiver operator characteristics (ROC)
 - analysis, 189–190
- Reconstruction filters, 311–312
- Reconstruction method, 381–383
- Red blood cell number, IVUS and, 19–23, 30
- Reflectance maps, 258, 259, 261, 262, 280, 281
 - linear approaches to, 273–279
 - Zheng-Chellappa method and, 282, 283, 284
- Region-aided geometric snake (RAGS), 536–537, 544, 546–575
 - experiments and results, 559–571
 - formulation of, 549–550
 - mean shift algorithm and, 536, 546, 557–559, 574
 - numerical solutions for, 552–555
 - shortcomings of, 571
 - summary of algorithm, 559
 - on vector-valued images, 536, 555–557
- Regularization. *See* Cross-scale regularization
- Regularization term, 281, 297
- Reinitialization
 - of the level set method, 202, 203, 215–216, 217, 218, 219, 520
 - of the particle level set method, 241
- Relativity theory, 60
- Renal arteries, MRA of, 182
- Repetition time (TR), 126, 128, 129, 165
 - FLAG and, 172
 - INFLOW method and, 169, 170
 - inflow related artifacts and, 130
 - short, 164
 - TOF MRA and, 135, 136, 143–144
- Ridgelets, 325
- Riemannian mean curvature flow, 585, 619
- Riemannian metric, 584
- Riemannian notation, 556
- Riemannian space, 538, 596
- ROC analysis. *See* Receiver operator characteristics analysis
- Roughness of image, 339
- RSE. *See* Rapid sequential excitation
- RT3D cardiac ultrasound. *See* Real-time three dimensional cardiac ultrasound
- Rubidium-82 (^{82}Rb), 62t

- Salt and pepper noise, 480, 499, 500, 514f, 587
- Sample mean shift, 558
- Saturation effects, 163–166
- SCAD thresholding, 326
- Scaling function, 289, 294, 308–309
- Scatter correction, 77
- Scattered coincidences, 68
- Scintillation camera, 98–99
- Seed a node element, 387

- Segmentation. *See also* Advanced segmentation techniques; Level set segmentation; Mean shift algorithm; Subjective surface based medical image segmentation
- crisp, 480
 - diffused region force and, 547, 549
 - fuzzy, 480, 498–512
 - out-of-core approaches to, 386–388, 396–399, 402, 404
 - RAGS and, 536, 546
 - wavelets in, 338–343
- Segmentation problem, 416
- Semi-implicit co-volume scheme, 586, 598–609
- Sensitivity encoding (SENSE), 172–173
- SE sequence. *See* Spin echo sequence
- Shaded surface display, 163
- Shannon wavelets, 289
- Shape from shading (SPS) models, 257–298
- calculus of variations, 264–267
 - constant functions used in, 265–267
 - mathematical background of, 261–272
 - multiscale methods for, 271–272
 - numerical algorithm implementation in, 272–288
 - numerical methods for linear and nonlinear, 267–272
 - optimization approaches to, 279–288
 - problem of, 257–260
 - strip method in, 260, 262, 263–264
 - uniqueness and existence, 262
 - wavelet-based methods and (*see* Wavelet-based methods)
- Shape models, 364–365, 479
- Shear force, 123
- Shepp-Logan windows, 73, 85
- Shocks, 537, 554, 611–612
- Short echo proton magnetic resonance spectroscopic imaging (MRSI), 188
- Sigmoid functions, 322
- Signal loss, turbulence-induced, 132–133
- Signal modeling, 504–505
- Signal-to-noise ratio (SNR), 86, 159, 165, 166, 192
- BACSPIN and, 191
 - blood pool contrast enhancement and, 173
 - cross-scale regularization for images low in, 335–338
 - 3D-navigator echo MRA and, 175–176
 - 3D-TRICKS and, 180
 - fuzzy segmentation and, 500, 507, 509, 512
 - ODESSA and, 186
 - PC MRA and, 150
 - PET and, 86
 - wavelet transforms and, 329, 331, 333, 335–338
- Signal void, 168, 170, 182
- Signed distance function, 203, 215, 216–217
- deformable models and, 370–371
 - level set segmentation and, 520, 523
 - particle level set method and, 241
 - X-FEM and, 233
- Simulation-based methods, for scatter correction, 77
- Sine-cosine transforms, 316
- Single event, 69
- Single-photon emission computed tomography (SPECT), 59, 87, 89, 90, 92, 93
- data correction in, 77, 78–83
 - description and applications, 98–100
 - wavelet transforms and, 326, 332–338
- Single scale methods, for SPS models, 272
- Singular value decomposition (SVD) solver, 436
- Sinogram(s)
- error, 456
 - extrapolation from, 465–466
 - level set segmentation and, 450, 451, 453, 459, 464, 465–466
 - in PET, 69, 71, 85
- Sinogram-based deformations, 459
- Sinusoidal waves, 306
- Slantlets, 325
- Slice thickness, 136–137, 143, 145
- Slice-transition phenomenon, 126–129, 132
- Smoking, 490–491
- Snake models, 359–362, 512–515
- geometric (*see* Geometric (geodesic) snakes)
 - original, 365–367
 - parametric, 359, 360, 361, 535, 555
 - T-, 361, 362, 364, 377–378, 400
- SNR. *See* Signal-to-noise ratio
- Soft thresholding, 320
- SOR method. *See* Successive over relaxation iterative method
- Sound speed, 5, 11
- Sparse-field method, 420, 461
- Spatial adaptive threshold, 323, 324
- Spatial-frequency tiling, 307, 308
- Spatial misregistration effects, 155–156
- Spatial resolution, 32–34
- Spatial-temporal analysis, 326–332
- Speckle noise, 329–331
- SPECT. *See* Single-photon emission computed tomography
- Spectral analysis, 94–95
- Speed function, 202, 215, 217, 219, 220, 419, 431–433, 437, 517
- advective part of, 206
 - anisotropic, 223–224

- Speed function (*cont.*)
 - deformable models and, 370
 - diffusive part of, 206–207
 - ordered upwind method and, 223–224
 - particle level set method and, 240
 - RAGS and, 574
- Speed images, 157
- Spin echo (SE) sequence, 118, 119, 120
 - flow-related enhancement and, 128
 - slice-transition phenomenon and, 126, 127
 - spin phase phenomenon and, 131, 132
- Spin isochromats, 124, 126–129, 133
 - flow information in, 120–121
 - in motion, 119–120
 - PC MRA and, 155
- Spin-lattice relaxation time (T₁), 165
- Spin-phase dispersion. *See* Phase dispersion
- Spin-phase phenomenon, 131–132, 133–134
- Spline functions, 325, 334, 335, 336
- SPS models. *See* Shape from shading models
- SSFP sequence. *See* Steady-state free precession sequence
- Standard data, IVUS, 31
- Static force fields, 552
- Static vector fields, 552
- Static velocity fields, 555
- Statistical learning, 298
- Steady-state free precession (SSFP) sequence, 185–186, 190, 193
- Steerable filters, 327
- Stein unbiased estimated risk, 324
- Stenoses, 167, 480, 520
 - black blood MRA and, 168
 - coronary MRA ROC and, 189–190
 - 3D-navigator echo MRA and, 176
 - MIP for, 190
 - TOF MRA of, 139
- Stereographic coordinate systems, 259
- Stochastic image models, 480, 481–490
 - high-level processes in, 484, 489–490
 - low-level processes in, 484, 485–489
 - statistical framework for, 482
- Stopping functions, 550, 552, 555, 556, 561
- Strip method, 260, 262, 263–264
- Strong smoothness, 266
- Structure, defined, 58
- Structuring element, 373
- Subjective surface based medical image
 - segmentation, 583–620. *See also* Co-volume level set method
- Successive over relaxation (SOR) iterative method, 607–608, 610–611
- Sum square error, 35–36, 44, 46t
- SVD solver. *See* Singular value decomposition solver
- TACs. *See* Time-activity curves
- Taylor series expansions
 - immersed interface method and, 235–236, 237
 - level set segmentation and, 517
 - spin isochromats and, 119
 - SPS models and, 260, 267, 268, 270, 273, 277, 282, 284
 - wavelet-based methods and, 295
- T-BON octree. *See* Temporal branch-on-need octree
- ^{99m}Tc-teboroxime, 100
- TE. *See* Echo time
- TEM. *See* Transmission electron microscopy
- Temporal branch-on-need (T-BON) octree, 384–385
- Tensor invariants, 442, 443–444
- Tensor product of wavelets, 291–292, 295, 297
- Tentative points, 208–209, 225
- Termination condition, 366
- Tetra-cubes, 379, 380, 381
- Tetrahedral decomposition, 367, 368
- Texture classification, 326–327, 338–339
- Three-dimensional (volumetric) acquisition, in PET, 69, 71
- Three-dimensional elliptical centric view ordering (3D-ELLIP), 180
- Three-dimensional magnetization-prepared true fast imaging, 190
- Three-dimensional navigator echo (NE) magnetic resonance angiography (MRA), 175–176
- Three-dimensional real-time navigator magnetic resonance coronary angiography, 190
- Three-dimensional time-resolved imaging of contrast kinetics (3D-TRICKS), 180
- Thresholding
 - affine (firm), 320
 - on 3D wavelet modulus, 333–335
 - hard, 320
 - non-negative garrote, 325
 - SCAD, 326
 - soft, 320
 - spatial denoising via, 327–328
- Thresholding operators, 319–320, 325–326, 336
- Threshold value selection, 323–324
- Thyroid cancer, 96
- Tikhonov regularization approach, 297
- Tilt Optimized Nonsaturated Excitation (TONE), 165, 166, 167
- Time-activity curves (TACs), 87, 89, 92
- Time discretization, semi-implicit scheme in, 598–600
- Time-limited wavelets, 289, 297

- Time-of-flight magnetic resonance angiography (TOF MRA), 124, 126, 134–145, 163, 164, 167, 168, 193, 521, 524. *See also* INFLOW method
 - 2D, 134, 135–141, 166
 - 3D, 134, 141–145, 166
 - ultrashort CE MRA *vs.*, 187
- Tissue scatterers, 23–24
- TOF MRA. *See* Time-of-flight magnetic resonance angiography
- Tomographs (scanners), 67
- TONE. *See* Tilt Optimized Nonsaturated Excitation
- Topological/logical operations, 421
- Total squared brightness error, 265
- TR. *See* Repetition time
- Tracer kinetic modeling, 86–89
- Trains pulses, 13–14
- Transducers, IVUS, 3, 7, 8f, 10–12, 16–19, 24–26, 28, 31, 43
 - angular velocity of, 19, 20f
 - artifacts caused by, 24
 - sheathing dimensions of, 19–21
- Transmission electron microscopy (TEM), 416, 425–427, 452, 463–465, 467
- Transmission scans, 81–83
- Transverse relaxation times (T₂), 124, 163
- Transverse tetrahedrons, 368
- Travel-time arrivals, computing, 223, 224
- Triangular elements, 368
- True coincidences, 67, 68f, 78
- Tsai-Shah's algorithm, 273, 276–279, 280f, 287
- T-snakes model, 361, 362, 364, 377–378, 400
- T-surfaces model, 361, 364, 386–388, 397, 398–399, 400–401, 402, 403
 - artery reconstruction and, 394–396
 - entropy condition of, 369, 371
 - initialization of, 371–381
 - properties of, 367–370
 - reconstruction method and, 381, 382–383
- Turbulent flow, 121–123
- Two-dimensional (planar) data acquisition, in PET, 69, 71–75
- Two-dimensional Fourier transform gradient recalled echo (2D-FT-GRE), 130
- Two-dimensional separable wavelets, 291–292, 297
- Ultrashort contrast-enhanced magnetic resonance angiography (CE-MRA), 187
- Ultrasound pulse, 9, 10–12, 13–14, 24–25, 26
- Ultrasound speed, 14, 15f
- Uncertainty principle, 307
- Underconstrained tomographic problems, 450–451
- Undersegmentation color region maps, 566–568, 569f, 570f
- Unit normal term, 281
- Universal threshold, 323
- Up-sweep process, 211
- Upwinding finite difference scheme, 575
- Upwind methods, 206, 226f
 - co-volume level set method and, 586
 - level set segmentation and, 419–420, 437, 460, 461
 - ordered, 223–225
 - RAGS and, 554, 555
- Validated data, IVUS, 31
- Variable echo time (VTE), 182
- Vascular malformations, 480, 521
- Vector-valued images, 536, 555–557
- Velocity dephasing, 146–147, 149
- Velocity encoding (VENC), 150, 156–163, 189
- Velocity extensions, 203, 205, 215, 219
 - attributes of, 216–218
 - improved, 225–230
- Velocity image, 155
- Velocity mapping, 179
- Venetian blind artifact, 166
- Venous anatomy, 140
- Vessel geometry, 136
- Vessel turns, 156, 167, 168
- Visible Human project, 394, 396, 399, 541
- Viterbi algorithm, 362
- Volume segmentation algorithm, 523
- Volumetric function, 418
- Voxels, 159–160
 - fuzzy segmentation and, 504
 - IVUS and, 21–23, 30
 - level set segmentation and, 416, 420, 421, 429, 521
 - TOF MRA and, 143
- Wavelet(s), 267, 305–347
 - band-limited, 289, 297
 - 1D, 288–291
 - 2D, 291–292, 297
 - Daubechies, 289, 293, 297
 - Gabor, 314–315, 318, 344
 - Haar, 289
 - image registration using, 343–344
 - Meyer, 289, 297
 - mother, 289, 290
 - nonseparable, 297
 - segmentation using, 338–343
 - Shannon, 289
 - time-limited, 289, 297
- Wavelet-based matrix compression, 272

- Wavelet-based methods, 261, 288–297
 - background of, 288–292
 - SFS, 293–296
- Wavelet modulus, three-dimensional, 333–335
- Wavelet packets, 314, 315–316, 318, 338–339
- Wavelet packets dictionary, 315, 316
- Wavelet transforms, 306–338
 - continuous, 307–309
 - denoising and, 306, 316, 319–338, 345
 - discrete, 309–313
 - image enhancement and, 319–338
- Weak-edge leakage problem, 536, 542, 543, 545, 546, 561, 566, 567, 568f, 571, 572
- Weak smoothness, 265
- Weighted curvature, 459
- Weighted integration, 95
- Weighted linear-least squares (LLS) methods, 94
- White blood angiograms, 168
- Whole-body positron computed tomograph (PETT III), 60
- Wiener filters, 332
- Wilcoxon's signed rank test, 176
- Windowed Fourier transforms (WFT), 306–307, 308
- Window functions, 316–318
- X-FEM. *See* Extended finite element method
- Yokoi plot, 93
- Zheng-Chellappa minimization method, 281–286, 293
- Zucchini dataset, 439, 440f