SEGMENTATION OF MEDICAL IMAGE VOLUMES USING INTRINSIC SHAPE INFORMATION

A DISSERTATION

SUBMITTED TO THE PROGRAM IN

MEDICAL INFORMATION SCIENCES

AND THE COMMITTEE OF GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Smadar Shiffman

January 1999

In memory of

Sara ne Gobetz and Jaacob Koker,

whom the Nazis uprooted from Amsterdam, The Netherlands,

and murdered in Sobibor, Poland,

on May 28, 1943.

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Sandy Napel
Department of Radiology
Stanford University
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Russ Altman
Department of Medicine
Stanford University

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Rajeev Motwani
Department of Computer Science
Stanford University

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

Recent advances in computing capabilities indicate that, in the near future, radiologists may be able to evaluate patient cases by observing 3D views of patient data, rather than observing sequences of 2D views. An important step in preparing image volumes for 3D viewing is segmentation of the data, followed by editing for removal of irrelevant structures. The application area of my work is editing of computed tomography angiography (CTA) volumes to allow improved visualization of vascular pathology. Current segmentation and editing methods are inadequate for editing of CTA volumes because (1) the methods are time consuming and are of limited reproducibility, or (2) the methods require elaborate prior global shape models, which are hard to construct for CTA data because of normal and pathological variation in the structure of blood vessels.

I propose a different approach to segmentation of image volumes that requires only a small amount of user intervention and that does not rely on prior global shape models. The approach, intrinsic shape for volume segmentation (IVSeg), comprises two methods. The first method analyzes isolabel-contour maps to identify salient regions that correspond to major objects. The method detects transitions from within objects into the background by matching isolabel contours that form along the boundaries of objects as a result of multilevel thresholding with a fine partition of the intensity range. The second method searches in the entire sequence for regions that belong to an object that the user selects from one or a few sections. The method uses local overlap criteria to determine whether regions that overlap in a given direction (coronal, sagittal, or axial) belong to the same object. For

extraction of blood vessels, the method derives the criteria dynamically by fitting cylinders to regions in consecutive sections and computing the expected overlap of slices of these cylinders.

In a formal evaluation study, I showed that IVSeg reduced user editing time for CTA cases by a factor of 5 without affecting the results in any significant way. IVSeg's contribution is a novel combining of intensity information and intrinsic shape information to solve the editing problem. Because IVSeg does not make assumptions about the contents of the input images, I expect it to be useful for editing of images independent of imaging modality. Topics for future research include extension of the methods in IVSeg to nonscalar images—for example, color images—and segmentation of volumes via surface matching. Another research direction is application of IVSeg’s methods to automated indexing and retrieval of images by content.

Acknowledgments

I am extremely fortunate to have had the opportunity to pursue a doctoral degree at the Medical Information Sciences (MIS) program at Stanford, and to perform my research under the supervision of Sandy Napel at the Radiological Sciences Laboratory (RSL). I am grateful to Sandy for his support and help, and for his responsiveness to new ideas. Sandy's thorough critique ensured that I stay on the path to exercising my best possible performance. I thank the members of my reading committee, Russ Altman and Rajeev Motwani, for taking the time to review my dissertation. Russ helped me to clarify ideas that became the foundation for my research, and Rajeev provided essential suggestions. I also thank Dwight Nishimura and Iain Johnstone for serving on my oral-defense committee and for reviewing intermediate stages of my work.

Four institutions provided support for my research: The National Library of Medicine (LM-07033), the CAMIS resource grant (LM05305), the National Institutes of Health (R01HL50305, P41-RR09784), Siemens Medical Systems, and the Lucas Foundation. In both MIS and RSL, I enjoyed the tremendous flexibility and the willingness of the faculty to facilitate challenging research. I am grateful to Edward Shortliffe and Lawrence Fagan, and to Gary Glover and Norbert Pelc, for providing stimulating and enriching research environments at MIS and RSL, respectively.

My work benefited greatly from three excellent computing environments. I am indebted to Thomas Rindfleisch, Christopher Lane, and Farhad Shafa for ensuring computing support at Stanford Medical Informatics (SMI), and to Thomas Brosnan and Sandy Napel for

enabling a smooth computing environment at RSL and at the 3D laboratory, both within Stanford's Department of Radiology. I am thankful to Carl Crawford for allowing me to use the CT simulator, to Trevor Hastie for helping me to refine the representation of splines that I incorporated into my work, and to Mike Walker for helping me with the design and the analysis of my evaluation studies. I acknowledge that discussions that I had with Chye Hwang Yan, Shin Yi Yen, and Thilaka Sumaneweera promoted the elaboration of parts of my work. I appreciate the hours of tedious work that Geoffrey Rubin, Roger Shifrin, Christopher Beaulieu, Steven Heiss, Bruce Daniel, and Laura Logan put into editing CTA cases for my evaluation experiments. I thank Maria Tovar, Ramani Pichumani, Rhea Tombropolous, David Paik and Scott Schmidler for sharing helpful imaging-related information, and I thank Lyn Dupré for editing this work.

Many staff members and friends helped me sail smoothly through the tasks of being a graduate student and a mother at the same time. Darlene Vian was always generous with comfort and advice. The smiles of Barbara Morgan and Pat Swift helped me to endure my workload when I was exhausted. Rosalind Ravasio and Donna Cronister always made sure that logistics were smooth. Jennifer Niederman and Marianne Ault-Riché taught me how to listen. Dafna Bahir and Rivka Har-El shared my joy and frustration. My friends and fellow students at SMI and at RSL provided continual support and encouragement.

I am profoundly grateful to my mother Michal Elata, for always supporting the choices I made, and for being there for me while I struggled. I dedicate this dissertation to the memory of her parents. I thank my parents Chaim Elata and Gerda Elata-Alster for teaching me the importance of persistence, and I thank my siblings Vardit Kindler, David Elata, and Itai Elata for staying in touch. I am grateful to my husband, Dov Shiffman, and to my children, Naomi and Nadav, for coping with my being heavily immersed in my work for numerous days and nights. I allot my deepest gratitude to Dov, who took a great share in our mutual responsibilities to allow me to pursue my love of learning, and who stood by me all along this challenging path.

Table of Contents

Abstract v

Acknowledgments vii

List of Illustrations xv

List of Tables xviii

Chapter 1 Segmentation of Medical Image Volumes 19

1.1 Computed-Tomography Angiography 21

1.2 Editing of Image Volumes 23

1.2.1 Manual Tracing 23

1.2.2 Propagation of Operations Throughout a Sequence 26

1.2.3 Intensity Thresholding 26

1.2.4 Connectivity 31

1.2.5 Use of Shape Information 31

1.2.6 Summary 32

1.3 Specific Aims 32

1.4 Proposed Framework for Segmentation 33

1.5 Research Hypothesis 35

1.6 Significance 36

1.7 Dissertation Roadmap 36

Chapter 2 Image Segmentation: Current Methods and Evaluation Techniques 38

2.1 Framework for Characterization of Segmentation Methods 39

2.2 Boundary-Detection Methods 41

2.2.1 Convolution of the Image with Gradient Operators 41

2.2.2 Deformable Models 43

2.3 Region-Detection Methods 44

2.3.1 Threshold-Based Relaxation 44

2.3.2 Feature-Based Classification 45

2.3.3 Random-Field Models and Bayes' Formulation 48

2.4 Summary of 2D Segmentation Methods 49

2.5 Extensions of 2D Segmentation to 3D 50

2.6 Evaluation of Segmentation Algorithms 52

2.6.1 Performance Criteria 53

2.6.2 Evaluation Strategies 53

2.6.3 Standard Data for Evaluation 54

2.6.4 Evaluation Measures 56

2.6.4.1 Least-Square Error 56

2.6.4.2 Correlation 57

2.6.4.3 Rate of Misclassification 58

2.6.4.4 Pixel-Distance Error 58

2.6.4.5 Comparison of Rate of Misclassification and Pixel-Distance Error 59

2.6.4.6 Importance Weighting of Errors 59

2.6.4.7 Operator Variability 60

2.7 Summary 60

Chapter 3 Principles That Affect Images of Boundaries 61

3.1 Factors That Affect the Value of a Voxel 62

3.1.1 Physical Property Measured 62

3.1.2 The Response of Imaging Systems 63

3.1.2.1 Spatial-Domain Definition 63

3.1.2.2 Frequency-Domain Definition 64

3.1.2.3 Spatial Resolution 64
3.1.2.4 Partial-Volume Effect 66
3.1.3 Random Noise and Streak Artifacts 67
3.2 Examples from CT Simulations 68
3.2.1 Line-Spread Function 69
3.2.2 PVE in Cylinders 69
3.2.3 Images of Adjacent Objects 71
3.3 Summary 72

Chapter 4 Existing Methods for Contour Comparison 73
4.1 Contour Representation 74
4.1.1 List of Coordinates 74
4.1.2 Parametric Cubic Curves 75
4.1.3 Turning Angle 76
4.1.4 Spatial Moments 77
4.1.5 Fourier Descriptors 78
4.1.6 Wavelets 79
4.2 Distance Measures 80
4.3 Summary 82

Chapter 5 Segmentation of 2D Sections Using Isolabel-Contour Maps 83
5.1 Theory 85
5.1.1 Formation of Isolabel-Contour Maps 86
5.1.2 Characteristics of Isolabel Contours 89
5.1.3 Detection of Object Boundaries via Matching of Isolabel Contours 95
5.2 Application 96
5.2.1 Selection of Isolabel Contours 96
5.2.2 Extraction of Isolabel Contours 98
5.2.3 Representation of Contours 99
5.2.4 Distance Measure for Assignment of a Similarity Score 104
5.2.5 Contour Matching and Formation of Salient Regions 105
5.2.6 Interpretation of Images With Salient Regions 107

5.3 Performance Analysis 110
5.3.1 Sensitivity Analysis 110
5.3.1.1 Study Design for Sensitivity Analysis 110
5.3.1.2 Sensitivity Analysis Measures 110
5.3.1.3 Sensitivity Analysis Results 112
5.3.1.4 Sensitivity Analysis Conclusions 115
5.3.1.5 Comparison Study Design 115
5.3.1.6 Comparison Results 116
5.3.1.7 Comparison Conclusions 118
5.3.1.8 Repeat of Statistical Analysis for Observations without Leaks . 118
5.4 Summary 120

Chapter 6 Dynamic Region-Overlap Criteria for Extraction of Vessel Structures from a Sequence of 2D Sections 122
6.1 Theory 124
6.1.1 Testing Whether a Region Is Elliptic Via an Elliptic Approximation of a Contour 125
6.1.2 A Proof that Right Cylinders with Circular Bases Have Elliptic Sections 130
6.1.3 The Overlap of Consecutive Sections Through a Right Cylinder 135
6.1.3.1 Derivation of the Relative-Overlap Expression 136
6.1.3.2 Demonstration of Elliptic-Region Overlap 139
6.2 Application 142
6.2.1 Search for Elliptic Regions That Might be Amenable to Application of the Overlap Criteria 143
6.2.2 Dynamic Estimation of Expected Overlap for Elliptic Regions 146
6.3 Sensitivity Analysis 148
6.3.1 Study Design for Sensitivity Analysis 149
6.3.2 Sensitivity-Analysis Measures 151
6.3.3 Sensitivity-Analysis Results 151
6.3.4 Sensitivity-Analysis Conclusions 153
6.4 Summary 153

Chapter 7 Comparison of Manual Editing to Editing with IVSeg 155

7.1 IVSeg: A System for Editing Medical Image Volumes 156

7.1.1 Step 1: Identification of Salient Regions 156

7.1.2 Step 2: Region Grouping 159

7.2 Evaluation Methods 162

7.2.1 Experiment 1: Matching of IVSeg-Generated Regions and Manually Edited Regions 163

7.2.1.1 Data Collection for Experiment 1 163

7.2.1.2 Data Analysis for Experiment 1 165

7.2.2 Experiment 2: Comparison of User Time Required for Editing 166

7.2.3 Experiment 3: Comparison of Difference Between IVSeg and Manual Results to Inter-Expert Variability for Manual Editing 166

7.2.3.3 Data Collection for Experiment 3 167

7.2.3.4 Data Analysis for Experiment 3 168

7.2.3.5 Evaluation Results 169

7.2.4 Results for Experiment 1 169

7.2.4.6 Assessment at the Volume Level 170

7.2.4.7 Assessment at the Region Level 172

7.2.5 Results for Experiment 2 175

7.2.6 Results for Experiment 3 177

7.3 Evaluation Conclusions 178

7.4 Summary 179

Chapter 8 Conclusion 181

8.1 Summary of Contributions 181

8.1.1 2D Segmentations Using Analysis of Isolabel-Contour Maps 182

8.1.2 Composition of 3D Vessel Trees Through Classification of Regions . . . 183

8.1.3 IVSeg: A Robust Volume Editor That Requires Little User Input 184

8.2 Limitations 184

8.2.1 Assumptions That Limit the Applicability of the Method for Identification of Salient Regions 184

8.2.1.1 Assumption 1: There Exist Intensity Thresholds That Separate Between Target Objects and Their Background 184
8.2.1.2 Assumption 2: Sections of Target Objects Are Larger Than 10 Pixels 185
8.2.2 Assumptions That Limit the Applicability of the Region-Grouping Method 185
8.2.2.3 Assumption 1: We Can Represent Structures of Interest as Compositions of Right Cylinders 185
8.2.2.4 Assumption 2: We Can Use Overlap Criteria to Distinguish Between Regions That Belong to Different Structures 186
8.2.3 Computation Time 186
8.3 Directions for Future Research 187
8.3.1 Extensions to Current Methods 187
8.3.1.1 Determination of SDT as a Function of Target Object and Imaging Parameters 187
8.3.1.2 Adjustment of the Boundary of Salient Regions According to Local Image Characteristics 187
8.3.1.3 Tracking of Object Features Beyond Two Consecutive Sections 188
8.3.1.4 Extension of the Overlap Criteria to Include Organs of Arbitrary Form 188
8.3.1.5 Matching of Isolabel Surfaces 189
8.3.2 Improvement of Implementation 189
8.3.3 Additional Evaluation Studies 189
8.4 Summary 190

Appendix 1 The Error Associated with a Spatial Transformation of an Optimal Ellipse 191

Appendix 2 Derivations and Limits Analysis for Relative-Overlap Expression 194

References 197

List of Illustrations

Figure 1.1 A CTA section of the abdomen. 21

Figure 1.2 Anterior–posterior MIP of the complete dataset 22

Figure 1.3 An edited section that a radiologist generated by manual tracing around the aorta...24

Figure 1.4 A CT angiogram of an aortic aneurysm generated with manual editing followed by MIP. 24

Figure 1.5 The histogram for a CTA image..29

Figure 1.6 Results of thresholding an image with a single threshold at a variety of intensity levels. 30

Figure 1.7 The four steps in the framework that I propose..34

Figure 3.1 A reconstruction matrix..66

Figure 3.2 Partial-volume effect 67

Figure 3.3 An illustration of the line-spread function...69

Figure 3.4 An illustration of the PVE 70

Figure 3.5 Images of 6-mm slanted cylinders and their intensity profiles.........................71

Figure 3.6 Images of two adjacent cylinders 72

Figure 5.1 Steps that lead to the identification of salient regions in a 2D section............87

Figure 5.2 A label pattern formed by thresholding of the intensity range with a fine partition 87

Figure 5.3 A magnified CTA section and its isolabel-contour map 88

Figure 5.4 Contour patterns for the major structures in an image 89

Figure 5.5 Isolabel contours in a noiseless CT simulation 91

Figure 5.6 Isolabel contours from a CTA section 92

Figure 5.7 A simulation of a section through two cylinders that are 1 mm apart.............93

Figure 5.8 A hierarchical description of regions and their corresponding contours..........94

Figure 5.9 Consecutive isolabel contours 95

Figure 5.10 Contours that IVSeg extracts for an object with a nonmonotonic intensity profile..98

Figure 5.11 Results of applying connectivity to identify the next contour....................... 99

Figure 5.12 An extracted isolabel contour and its coordinates.......................................100

Figure 5.13 Splines fitted to a contour..102

Figure 5.14 Plots of four splines and plot of average spline...102

Figure 5.15 The isolabel contour that follows a contour 103

Figure 5.16 Turning-angle sequences and their optimal mapping..................................104

Figure 5.17 Pseudocode for matching of isolabel contours..105

Figure 5.18 Demonstration of salient regions..109

Figure 5.19 Histogram of the NVM for 1242 objects..114

Figure 5.20 Error-bar plot of mean difference of image MNVM...................................117

Figure 5.21 Leaks from fine branches of the iliac arteries into the pelvic bone.............118

Figure 5.22 Error-bar plot of mean difference of image MNVM for the repeat analysis...120

Figure 6.1 Intermediate results in the course of IVSeg’s operation 123

Figure 6.2 Hypothetical connected regions in consecutive CTA sections.......................123

Figure 6.3 Modeling of a vessel structure as a composition of small cylinders 125

Figure 6.4 Nonelliptic sections of hypothetical tortuous blood vessels..........................131

Figure 6.5 Sections of an in-plane branching vessel in orthogonal reformations of the data 132

Figure 6.6 A cylinder in an arbitrary coordinate system 133

Figure 6.7 Overlap between consecutive cylinder sections..136

Figure 6.8 Intersecting ellipses, top view 137

Figure 6.9 A filled scan-converted ellipse that corresponds to a cylinder section 140

Figure 6.10 Relative-overlap (RO) curves..141

Figure 6.11 Steps in IVSeg’s procedure for region grouping...143

Figure 6.12 Application of overlap criteria to test a new region 144

Figure 6.13 Procedure for identifying candidate regions in an orthogonal reformation...145

Figure 6.14 Cylinder fit to elliptic regions in consecutive sections................................147

Figure 6.15 A slanted right cylinder and its elliptic cross section..................................148

Figure 7.1 User interface for the user to specify threshold-partitioning parameters 157

Figure 7.2 Images within IVSeg’s sequence viewer..158

Figure 7.3 User interface for the user to invoke the region-grouping procedure 160

Figure 7.4 User interface for the region-grouping procedure following procedure activation...161

Figure 7.5 NVM for entire volumes that experts produced via manual editing and via editing with IVSeg. 170

Figure 7.6 A MIP view of case 20 that an expert edited manually and with IVSeg........171

Figure 7.7 Observations for which $NVM \geq 1$ displayed by case..173

Figure 7.8 Mean NVM for corresponding regions for the 10 cases 174

Figure 7.9 Bar chart of user time per section for the 10 cases..176

List of Tables

Table 2.1 Mapping of segmentation algorithms by characteristics 50

Table 5.1 *F* values for parameters and parameter interactions.......................................113

Table 5.2 Results of paired *t*-test for comparison of conventional thresholding and IVSeg..117

Table 5.3 Results of paired *t*-test for repeated comparison of conventional thresholding and IVSeg 119

Table 6.1 Structures used in sensitivity analysis 150

Table 6.2 *F* values for parameters and parameter interactions 152

Table 7.1 Cases that the experts edited for the evaluation study 165

Table 7.2 Cases that I used in each experiment 169

Table 7.3 Duration of manual editing and of editing with IVSeg 177

Chapter 1

Segmentation of Medical Image Volumes

An important component in evaluation of cross-sectional radiological image data is mental construction of 3D images of anatomical structure from sequences of 2D sections. Recent improvements in computing capabilities and cross-sectional imaging modalities—for example, computed tomography (CT), magnetic resonance (MR), and ultrasound—suggest that, in the near future, radiologists could evaluate cases by observing **3D views** of image data, rather than viewing the cross sections. I define 3D views as views that allow visual perception of 3D objects (note that 3D views are displayed via 2D images). Evaluation of cases from 3D views has the potential to reduce the time that the radiologist spends on each case, and may even reveal structures not discernible from the sequence of 2D sections.

There are several techniques available for 3D reconstruction of radiological image data—for example, maximum intensity projection (MIP) [1], shaded surface display [1], and volume rendering [2]. Independent of the reconstruction technique used, however, irrelevant structures might obscure details of organs of interest, and thus might interfere with case evaluation. Radiologists can remove obscuring structures by **editing** image volumes prior to reconstruction of 3D views of the data. Editing includes **segmentation**, or partitioning of the image data into distinct meaningful components, identification of structures of interest, and elimination of all other structures.

Segmentation is an important step in most applications that use medical image data. For example, segmentation is a prerequisite for quantification of certain disease entities and for radiation treatment planning [3, 4], for construction of anatomical models [5], for definitions of flight paths in virtual endoscopy [6], for content-based retrieval by structure [7], and for volume visualization of individual objects [4]. The goal of my research was to develop segmentation methods that would help radiologists to edit image volumes. The application that provided the incentive for my work is visualization of vasculature from computed-tomography angiography (CTA) data; therefore, the discussion in this dissertation is mostly about CTA data. However, I expect that the methods that I have developed will generalize, because many of the problems that my work addresses are common to image data from other medical imaging modalities.

In this chapter, I explain the challenges associated with segmentation of CTA volumes, I describe current methods for editing CTA volumes and identify their shortcomings, and I propose a new approach that addresses these shortcomings. Then, I state my research hypothesis and summarize the contribution of my work. I provide a dissertation roadmap at the end of the chapter.

1.1 Computed-Tomography Angiography

Data acquisition for CTA is performed through administration of iodine to a patient, followed by helical CT scanning [8-11]. Presence of iodine in the blood results in elevated density and a corresponding increase in attenuation, measured in Hounsfield units (HU) [12]. Consequently, CTA sections portray highlighted regions of blood flow relative to the soft-tissue environment in which blood vessels are embedded. Radiologists can identify vascular pathology—for example, stenoses and aneurysms—from observation of these highlighted regions. Figure 1.1 shows a CTA section of the abdomen that portrays an aortic aneurysm.

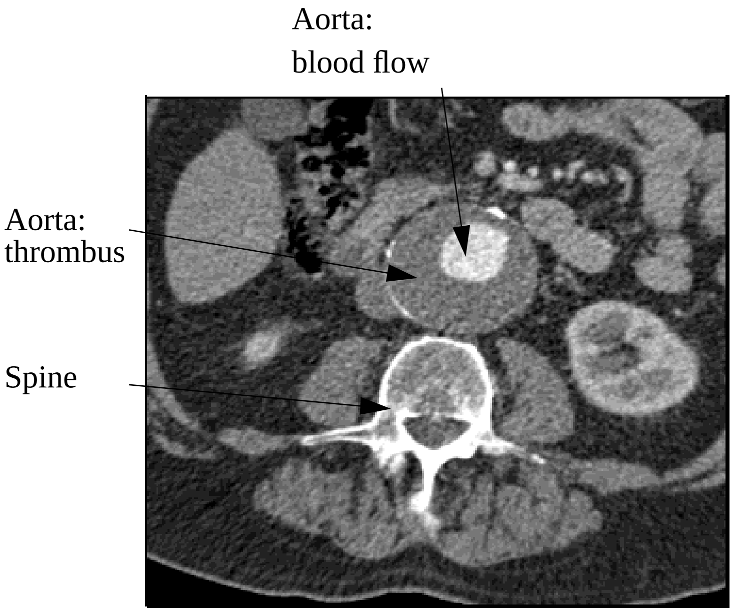


Figure 1.1 A CTA section of the abdomen that includes an aortic aneurysm. Note that the image of the aorta includes a central region of highlighted intensity, which denotes blood; a surrounding region of lower intensity, which designates thrombus; and several highlighted spots around the walls of the aneurysm, which designate calcification.

The ultimate goal for CTA is to produce 3D views of the data using 3D reconstruction techniques. The reconstruction technique that is most commonly used for CTA is MIP, whereby a set of projection rays is cast from one side of the volume to another, and the maximum intensity encountered along each ray constitutes a pixel value for a 2D image of the volume. Figure 1.2 shows an anterior–posterior MIP of the volume from which Figure 1.1 was taken. The MIP illustrates how structures of interest may be obscured by other structures when a dataset is reconstructed in 3D.

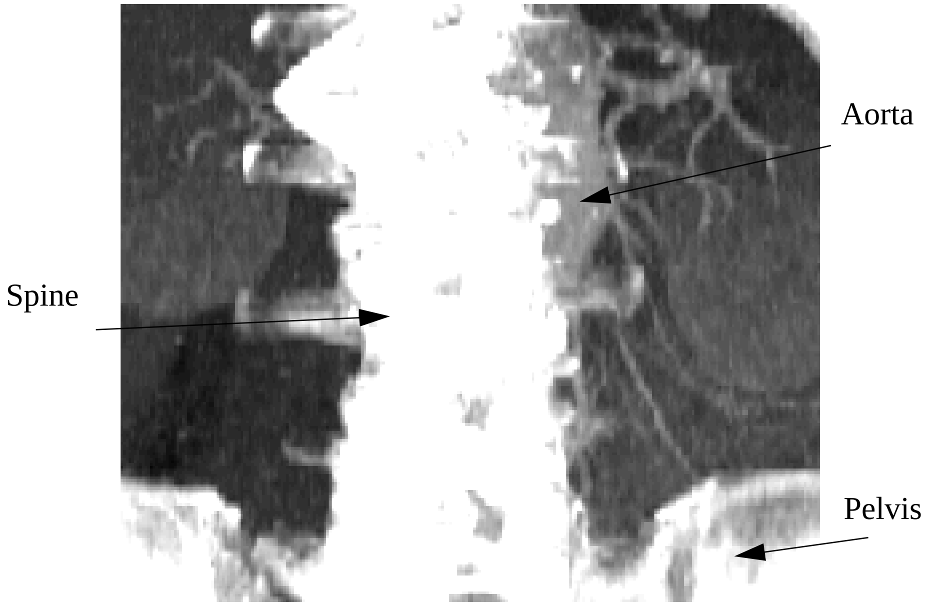


Figure 1.2 Anterior–posterior MIP of the complete dataset from which the section in Figure 1.1 was taken. Note that most of the aorta is obscured by the spine.

Ideally, a radiologist could edit 3D volumes to include only structures of interest by using simple automated segmentation—for example, segmentation based on intensity and spatial contiguity— and by eliminating all **voxels**, or volume elements, that are not within the chosen segmented regions. However, despite the enhancement of blood with iodine, several factors make it difficult to segment blood vessels from other structures automatically on the basis of intensity and spatial contiguity alone:

1. Measured attenuation may be corrupted by partial-volume effects and by other CT artifacts [13]. This corrupted measurement can cause portions of the intensity range of high-density structures (such as bone) and mid-density structures (such as soft tissue) to overlap with the intensity of blood vessels.
2. Attenuation within blood vessels may vary due to iodine transit delays within the vasculature.

3. Iodine enhances only the vascular lumen; yet, other structures, such as low-density thrombi and high-density calcium deposits, are also of diagnostic interest.

Examples of volumes that are difficult to segment automatically include those in which the aorta and the spine are close to each other to the extent that they might appear to be a single structure, and those in which the vascular lumen is surrounded by a thrombus, which has an appearance that is similar to soft tissue around it.

1.2 Editing of Image Volumes

The literature describes a variety of methods for editing of image volumes; most of them complement automated segmentation with user intervention. In Sections 1.2.1 through 1.2.5, I describe editing methods that are applicable to CTA data.

1.2.1 Manual Tracing

Manual tracing is a method that is widely used for delineation of regions of interest. The user identifies structures of interest on cross sections, and traces around them; the computer program tags voxels outside the traced regions and sets them to zero. Figure 1.3 shows an edited version of the section in Figure 1.1; a radiologist generated this image by tracing manually around the aorta. Figure 1.4 shows a MIP from 76 edited sections that a radiologist generated from the raw sections by tracing manually around the aorta and renal arteries. The resulting view provides a clear depiction of the aortic aneurysm with calcifications, and allows assessment of the size of the aneurysm and the aneurysm’s relationship to important branch structures, such as the renal and iliac arteries.

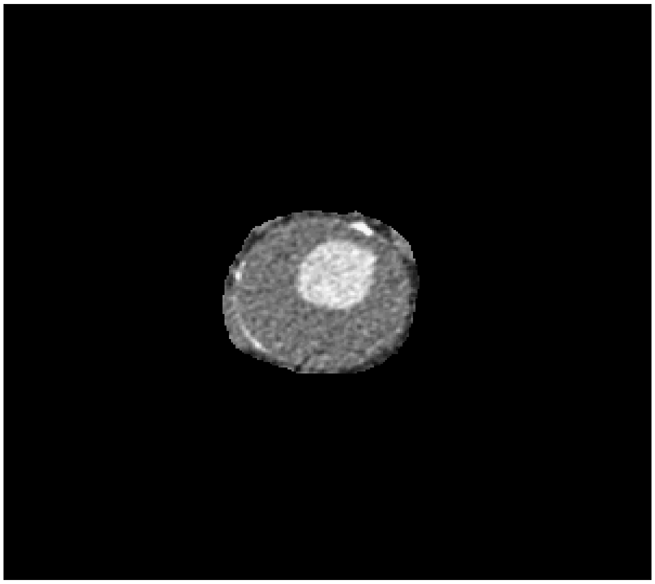


Figure 1.3 An edited section (section 40 in the sequence) that a radiologist generated by manual tracing around the aorta.

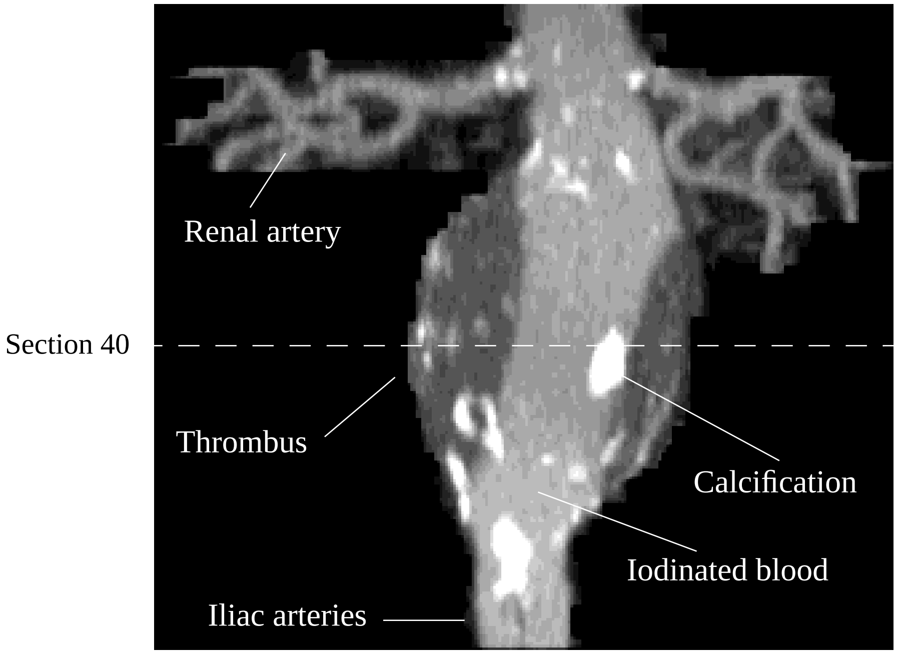


Figure 1.4 A CT angiogram of an aortic aneurysm generated with manual editing followed by MIP.

Editing via manual tracing can be performed in two modes [1]. In **exclusive editing mode**, experts trace regions selected for removal, and the software tags the voxels *inside* the surrounded regions so that the voxels will not show in subsequent 3D reconstruction. In **inclusive editing mode**, experts trace regions selected for inclusion, and the software tags the voxels *outside* the regions of interest. Inclusive editing must be performed with great care, because tracing is done close to the vessels, and thus editing errors might result in improper interpretation (for example, a false stenosis). The tolerance of inclusive editing to tracing errors depends on the 3D reconstruction method. For example, MIP projections are insensitive to inclusion of low-intensity margins; therefore, experts may save time by tracing inexact regions around included objects. Exclusive editing may be less sensitive to tracing errors than inclusive editing, because these errors affect target blood vessels only when the vessels are tangent to the removed objects. The distinction between exclusive and inclusive editing modes is applicable to any editing method, independent of the technique used to select structures of interest.

There are two major disadvantages to manual tracing. First, manual tracing is time consuming—for example, an experienced radiologist may require an average of about 3 minutes to trace around structures in a single section. Second, manual tracing does not ensure reproducibility of edits at different times or by different experts. The effects of these disadvantages are reduced slightly when manual tracing is combined with simple intensity-based processing to form semiautomated editing protocols. The following example illustrates a semiautomated editing protocol that combines manual tracing with intensity thresholding. First, a radiologist produces a rough trace around a set of structures of interest and removes everything outside the trace. Then, the radiologist applies intensity thresholding to delineate structures of interest accurately, and includes only voxels whose intensity is within the specified intensity bounds. Finally, the radiologist traces around spurious voxels and removes them. This editing protocol reduces the average editing time to about 1.5 minutes per section; it also improves reproducibility because it limits the number of possible outlines for each object. In Sections 1.2.2 through 1.2.5, I describe additional methods, which partly address limitations of manual tracing.

1.2.2 Propagation of Operations Throughout a Sequence

There exist several visualization and image-processing systems that include tools to expedite manual editing by propagating operations throughout a sequence. The MASK system [4] includes tools for propagation of contours in a sequence of sections, so that user traces for one section can be used as initial settings for tracing within successive sections. Chaney and Pizer [3] reported a system in which traces of cursor paths on arbitrary projections of image data (for example, sagittal, coronal, or oblique reformations or reprojections) triggered section-by-section activation of editing filters. Hentschel and colleagues [14] developed a suite of methods for editing CTA sections of specific organs. They reduced the workload associated with manual tracing by presenting to the user a slab of several superimposed images, and by having the user define a series of modifications to be carried out automatically on all the images included in the slab.

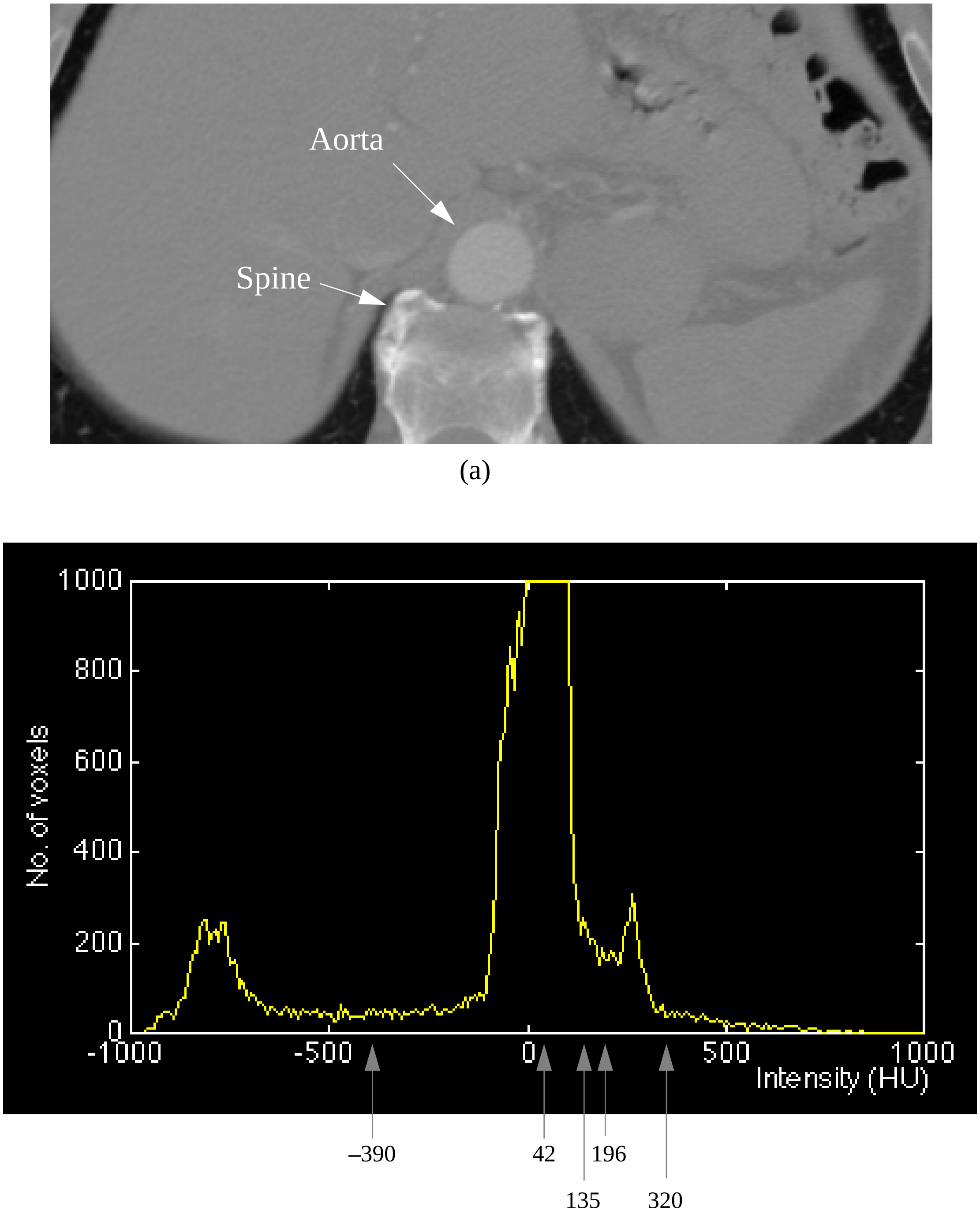
The preceding tools, however, still require at least some manual tracing and considerable user effort. Moreover, methods that require manual tracing on reprojections of a volume are not likely to be useful for removal of obscuring soft-tissue objects from sections, because such objects are typically embedded within complex anatomical structures and thus do not appear as separate entities in reprojections. Manual manipulation of slabs trades editing accuracy for reduced user intervention: An increase in the thickness of slabs results in a smaller number of slabs and leads to reduced user intervention. However, an increase in the thickness of slabs also results in averaging of information from a greater number of sections and consequently a single operation is less likely to be optimal for all sections included in the slab. Reproducibility may be improved by these techniques due to the reduced user input required, but complete reproducibility is unlikely to be achieved. Currently, most radiologists edit CTA sequences by applying to subsequences one of the methods described in the following sections.

1.2.3 Intensity Thresholding

Intensity thresholding assigns labels to pixels based on the pixels' intensities. Given a set of thresholds T of size $|T| = m$, corresponding subsets of the intensity range $[t_i, t_{i+1}]$

where $t_i, t_{i+1} \in T$, and a corresponding set of labels L of size $|L| = m - 1$, intensity thresholding produces a labeled image $g : N \times N \to L$ such that $g(x, y) = l_i$ where $1 \le i \le m - 1$ and $f(x, y) \in [t_i, t_{i+1}]$. When $|T| > 1$, I refer to the method as **multilevel thresholding**. The use of intensity thresholding for segmentation of medical images is appealing because intensity thresholds are easy to implement. However, objects that appear in medical images often have overlapping intensity distributions; therefore, thresholding is likely to result in misclassification of voxels, or in assignment of voxels to the wrong structure. When the prior distributions for tissue types are known, we can find the threshold that minimizes the probability of voxel misclassification [15]. Often, however, elaborate prior distributions that take into account imaging artifacts are not known, and users must select thresholds through trial and error to produce results that portray the structures of interest adequately. Typically users guess an initial threshold based on the expected intensity of the object of interest, then modify the threshold as they observe the results, until they find a good threshold. Noise and partial-volume effects also make it difficult for radiologists to find good thresholds. Users distinguish between good and inadequate thresholds mostly by the shape of the region of interest and by the extent to which the region is separated from its surroundings.

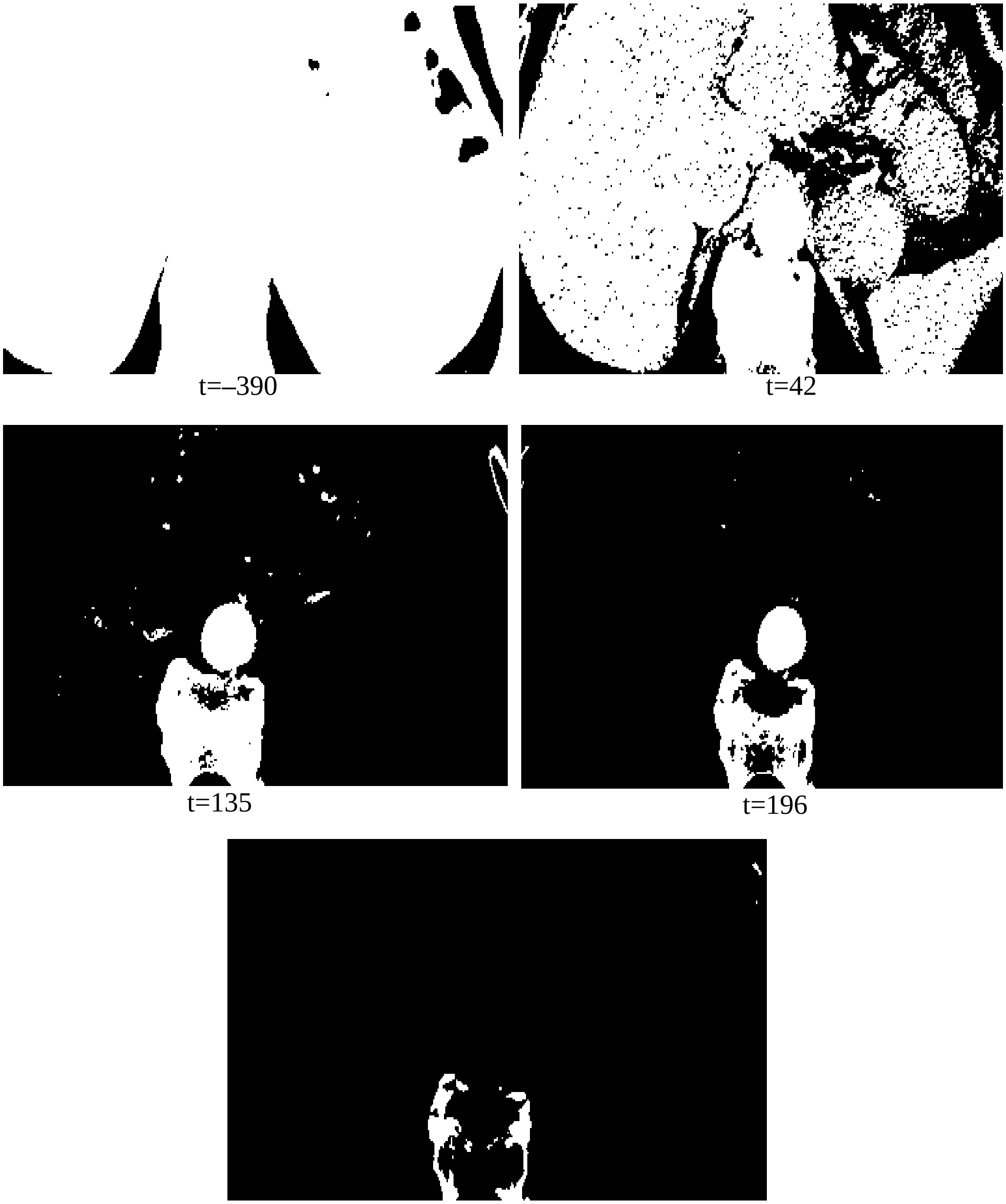
Sensitivity of the results of fully automated intensity thresholding to the selected threshold is an important concern when the imaged scenes include close objects with similar intensity distributions. In this situation, only a small range of thresholds produces adequate results, and selection of a threshold outside this range results either in elimination of portions of structures of interest from the segmented volume, or in **leaks** into undesired structures. Figures 1.5 and 1.6 illustrate the sensitivity of thresholding to selected absolute thresholds. Note that, in this example, the aorta and the spine are close to each other. The intensity range for the aorta is about 50 to 500 HU; that for the spine is about 0 to 1000 HU. Thus, the ranges overlap. Given a trial-and-error procedure for selection of a threshold that will separate the aorta from the spine, thresholds in the range 135 to 196 could have been reasonable a priori guesses. However, as Figure 1.6 demonstrates, a threshold of 196 separates the aorta and the spine, whereas a threshold of 135 results in a leak from the aorta into the spine. Thresholds that are much smaller than 135 result in embedding of the aorta in surrounding structures (for example, in Figure 1.6, see the image for a threshold of –390). Thresholds that are much greater than 196 result in a loss of part or all of the aorta (as happens, for example, in the image for a threshold of 320). Thus, in this example, only the threshold of 196 separates the aorta from the spine adequately.



(a)

(b)

Figure 1.5 The histogram for a CTA image. (a) The raw image (b) The image's histogram, clipped at 1000 voxels. The arrows on the bottom designate thresholds used to generate the images in Figure 1.6.



t=–390

t=42

t=135

t=196

t=320

Figure 1.6 Results of thresholding the image in Figure 1.5 with a single threshold at a variety of intensity levels. In each image, pixels whose intensities are above the threshold (t) are white. Of these five thresholds, only *t*=196 results in a clear separation of the aorta and spine.

1.2.4 Connectivity

Researchers have proposed semiautomated editing methods for removal of obscuring structures that are based on spatial connectivity of voxels [2, 14, 16]. These methods typically tag voxels that are connected to a prespecified initial **seed point**, and that meet a **similarity criterion**. The similarity criterion, which is often defined in terms of a threshold, determines the degree to which a voxel must be close in intensity to previously tagged voxels to qualify for tagging. Voxels that satisfy the criterion are considered as seed points for subsequent processing in the same manner. The procedure continues until there are no more candidate seed points. Examples of commonly used similarity criteria are absolute intensity thresholds and thresholds that limit the allowed intensity difference between the seed and a connected neighbor. The literature also refers to connectivity methods as *region-growing* [17] and *flood-filling* methods [18].

Hentschel and colleagues [14] used thresholding combined with repeated connectivity to determine automatically the contours of the ribs and spine for removal of these structures. They obtained good results for bone-image removal by having the user define a threshold for connectivity within each slab. However, my experience with CTA data is that connectivity often is inadequate for inclusive selection of pathological vessels from surrounding tissue, because of the considerable user intervention that is required for finding a good threshold.

1.2.5 Use of Shape Information

One research group demonstrated the merits of complementing intensity thresholds with shape information [19]. They enhanced an existing bone-removal method [14] to conduct an automated shape-based search for adequate thresholds. The search was guided by vertebral-shape criteria—for example, width, perimeter, and ratio of height to width—that were based on shape models that were derived empirically. The enhanced method allowed radiologists to remove bone from sequences of CTA images in less than 4 minutes on average. Many algorithms for segmentation of medical images use a priori knowledge in the form of global organ-shape models to guide the segmentation process—for example, the algorithms described by Duncan and colleagues [20] and by Brinkley [5]. Typically, these models are derived through an automated **supervised-learning** process [21], which produces a general model from training samples that an expert defines. However, similar shape models are not likely to be of help for inclusive editing of pathological vessels from CTA data because of the profound variation in the data.

1.2.6 Summary

A major difficulty with current methods for editing image volumes is that they often rely on intensity-thresholds that are hard to determine a priori. A search for a threshold that produces adequate results—a threshold that avoids leaks without losing significant portions of relevant objects—can require tedious manipulation of the volume. Although complementing intensity information with shape information is useful for some editing tasks, for inclusive editing of vessels from CTA volumes, predefined models are not likely to be successful due to limited generalizability. Thus, a central assumption in my work was that a priori global shape information regarding a particular dataset was unobtainable; consequently, it was my goal to pursue shape information from within images themselves.

1.3 Specific Aims

The goal of my work was to address the limitations of the approaches described in Section 1.2 by developing a method for segmentation of image volumes that would be robust and that would allow radiologists to perform inclusive editing of volumes with ease. The criteria for robustness included accurate and consistent segmentation of vessels under a range of noise levels and under a variety of image-acquisition conditions—for example, varying section thickness and spatial sampling. The criterion for ease of editing was short user-editing time. For example, for inclusive editing of blood vessels, I hoped to achieve user-editing time of less than 10 seconds per section on average. An additional requirement was that the method detect tortuous vessels without leaking into close structures, and without missing protrusions.

1.4 Proposed Framework for Segmentation

I propose a new framework for segmentation of volumes that allows extraction of objects from volumes with little user effort. To simplify the presentation of the framework, I introduce several definitions. I refer to coherent meaningful regions in an image as **salient regions**, to objects of interest (objects that the user selects) as **relevant objects**, and to image regions that belong to relevant objects as **relevant regions**. The framework constitutes the following four-step procedure (Figure 1.7):

1. **Region identification—** is automated identification of salient regions within 2D sections, and generation of images that display the salient regions with gray-scale coded labels. I refer to the resulting images as *images with salient regions*, and to the entire resulting sequence as a *volume of salient regions.*
2. **Region selection—**user selection of relevant regions from only a few of the images produced by the preceding step.
3. **Region grouping—**automated search within the volume of salient regions to group regions that relate to regions that the user selected. This step ends with the production of a sequence of mask images that include only relevant regions. The mask images indicate which voxels should be included in the edited images.
4. **Masking—**automated extraction from raw sections of voxels that are indicated by the mask images.

If user inspection reveals that relevant regions are missing, steps 2 and 3 may be repeated for correction of the results. From my experience with an implementation of the proposed framework, user modification is necessary in only a few images, and the modifications require little work. Following the production of edited images, the user can input the images to a 3D reconstruction procedure to generate 3D views of the data. Although 3D reconstruction is not part of the proposed method, it is helpful to illustrate the results of editing. Throughout this document, I illustrate the results of editing with views generated via MIP reconstruction.

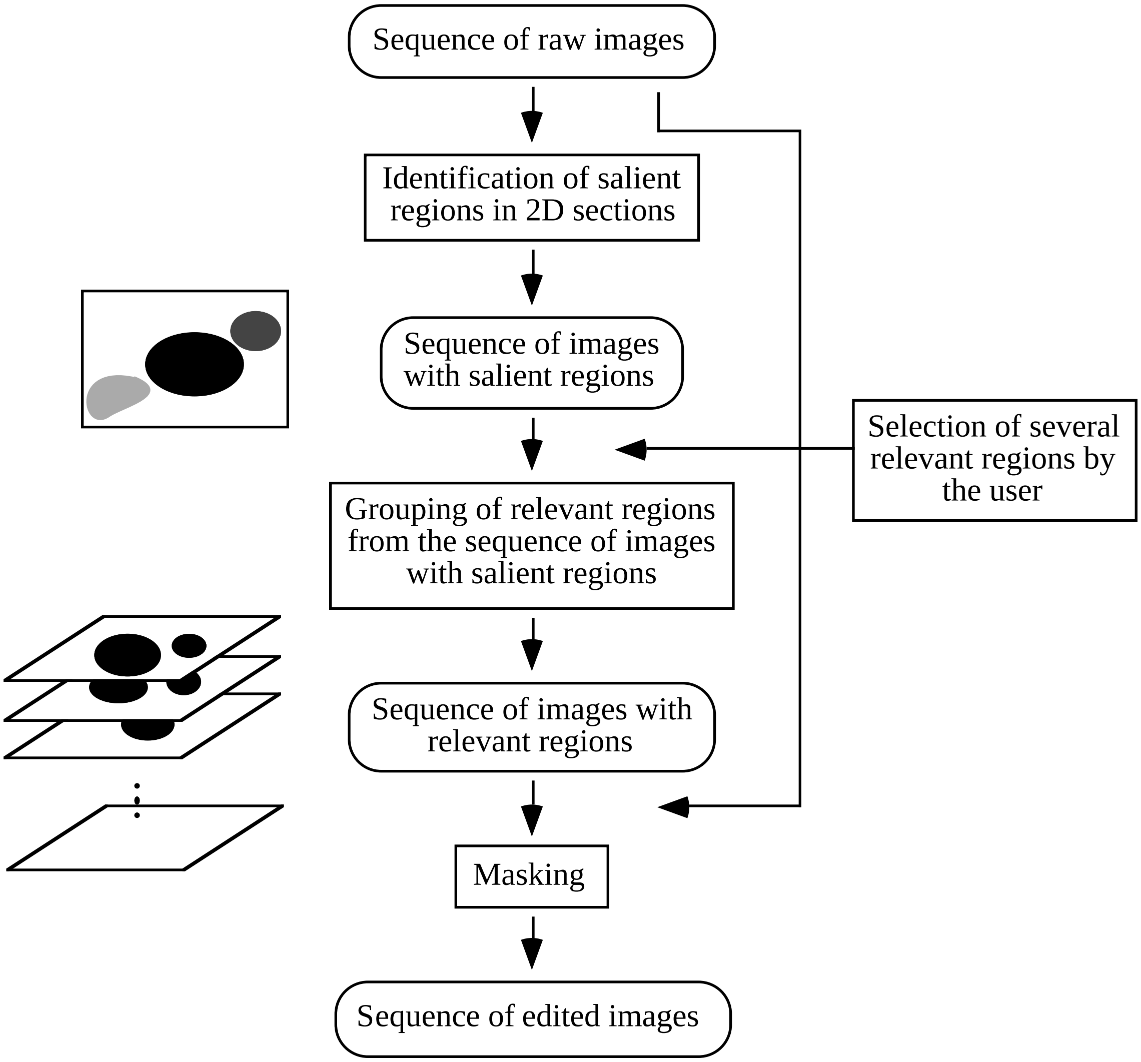

Sequence of raw images
Identification of salient regions in 2D sections
Sequence of images with salient regions
Selection of several relevant regions by the user
Grouping of relevant regions from the sequence of images with salient regions
Sequence of images with relevant regions
Masking
Sequence of edited images

Figure 1.7 The four steps in the framework that I propose (rectangular boxes). Note that users select relevant regions by clicking on salient regions with a mouse. The pictures on the left illustrate that the framework performs region identification separately for each section, and performs region grouping for the entire volume.

The two major procedures in the framework that I propose, region identification and region grouping, build on existing algorithms, multilevel thresholding and connectivity, respectively. Both of them use intensity thresholding. However, the two procedures complement the existing algorithms with shape information, which they derive from the image itself, to address limitations associated with intensity thresholding. Because the contribution of the new framework stems from the addition of shape information that is intrinsic to the image, I call the framework—and the program in which I implemented the procedures that the framework includes—**IVSeg**, for Intrinsic Shape for Volume Segmentation.

IVSeg exploits intrinsic shape information in two ways. First, as part of the procedure for identification of salient regions, within each individual cross section, IVSeg enhances conventional thresholding to produce patterns of similar quasi-concentric contours along the boundaries of object images; then, it exploits these patterns to detect transition from within objects into the background. IVSeg identifies such transitions by examining contours in sequence and detecting a contour whose shape is significantly different from the shape of the preceding contours. As each contour roughly represents the results of intensity thresholding with a single threshold for a local environment, the contour-matching procedure effectively considers results of multiple thresholding to decide on the locus of the object boundary. This use of shape considerations for automated segmentation mimics the way that humans use shape considerations to find a good threshold.

Second, as part of the procedure for region grouping, IVSeg determines whether salient regions in consecutive sections that appear to be connected in the *z* direction (or that overlap in the *xy* plane) belong to the same object by checking for coherence in the *z* direction. IVSeg assumes that an object is composed of primitive components, and determines the size and orientation of each component dynamically from each pair of overlapping salient regions. From these component parameters, IVSeg determines the expected overlap between consecutive component regions, and compares the expected overlap to the actual overlap observed for the salient regions. If the actual overlap is at least as large as the expected overlap, IVseg determines that the salient regions belong to the same object.

1.5 Research Hypothesis

The hypothesis of my research is that *IVSeg allows radiology experts to edit volumes of medical images robustly in less than 10 seconds per section on average, with results comparable to those produced by experts using conventional editing tools, but with less inter-*

and intra-operator variability. To verify the hypothesis, I conducted a study that assessed the compatibility of edits that experts produced using conventional editing tools with edits that they produced with IVSeg, and that measured the editing time for each method. I describe the evaluation study in detail in Chapter 7.

1.6 Significance

My research makes contributions to the fields of medicine, medical informatics, computer science, and electrical engineering. The contribution to medicine is a tool that will allow radiologists to visualize easily organs of interests without obscuring structures. The contribution to medical informatics is a novel way to separate relevant from irrelevant visual information. I expect that the methods in IVSeg could benefit a variety of applications that require segmentation of medical images—for example, construction of organ models and content-based retrieval from image databases. The contribution to computer science and electrical engineering is an approach that exploits intrinsic shape information to facilitate segmentation of image volumes.

1.7 Dissertation Roadmap

The dissertation is structured as follows. Chapters 2 through 4 describe the background of my work. Chapter 2 presents a review of existing segmentation methods. Whereas Section 1.2 includes a description of several segmentation methods in its presentation of current approaches to editing CTA volumes and the limitations of these approaches, Chapter 2 provides a more comprehensive review of existing methods for segmentation of medical images that I considered before I realized that a different approach might be necessary. Chapter 2 also includes a discussion of techniques for evaluating segmentation algorithms, and explains the background for choices I made with respect to the evaluation study in this work. Chapter 3 explains basic concepts that underlie medical imaging systems, which also constitute the underpinnings of the proposed framework. Chapter 4 discusses techniques for contour comparison, and explains the choices that I considered for comparing the shape of isolabel contours in the proposed segmentation framework. Chapter 4 relates to the discussion of measures for comparison of segmentation results in Chapter 2 in that segmentation results can be analyzed in terms of the associated boundaries.

The next two chapters describe the methods used by IVSeg. Chapter 5 presents the theory and application of identification of salient regions. Chapter 6 presents the theory and application of region grouping. Chapter 7 reports an evaluation study that I conducted to compare the ease and quality of editing CTA volumes with IVSeg to those with editing CTA volumes with conventional editing tools. Chapter 8 includes a summary of contributions, a description of the limitations of this work, and an outline of directions for future research.

Chapter 2

Image Segmentation: Current Methods and Evaluation Techniques

In Section 1.2, I discussed segmentation methods that radiologists have used to edit CTA volumes. My objective in the following review is to describe general approaches to segmentation that I considered, and to analyze the advantages and limitation of each approach as it pertains to editing of CTA volumes. Following the discussion of image segmentation here, I describe strategies for evaluation of segmentation algorithms that I considered for my evaluation study.

The goal of image segmentation is to partition images into coherent regions that correspond to objects in the imaged scene. Achievement of this goal is difficult in medical images due to the effects of noise, partial volume, and distortions, all of which are inherent in image acquisition, and may result in similarity in appearance of voxels that belong to different tissues. The literature describes many segmentation methods that employ a variety of approaches; I review only those approaches that are applicable to segmentation of CTA images, and I look at only the basic algorithms within each approach. A thorough discussion of the numerous variant and application-specific improvements to the basic segmentation algorithms is beyond the scope of this review. Additional information is provided in survey papers [21-26]. For simplicity, I describe methods for segmentation of 2D images, and then discuss implications for the extensions of these methods to 3D.

To guide my review, I use a framework for characterization of segmentation methods that includes five dimensions. This framework relates the methods I describe to one another. I use the framework to summarize the review after I discuss individual methods.

2.1 Framework for Characterization of Segmentation Methods

Most existing segmentation methods can be categorized along the following dimensions:

1. *Target entities pursued.* Typically, segmentation methods search for either boundaries of objects or regions covered by objects. Boundary-detection methods identify boundaries by observing discontinuity of image features. Region-detection methods identify regions by grouping pixels according to similarity of image features [17]. A few hybrid methods combine advantages by seeking both boundaries and regions.
2. *Features extracted.* Many segmentation algorithms operate on features derived from pixel intensity values at various scales, or local neighborhoods. Example features are the statistics mean, median, and mode, and differential operators that detect morphological properties such as edges and ridges [27]. It is desirable that features selected for segmentation be invariant to isotropic scale, rotation, translation, and affine transformations of the coordinate system in which they are represented, and to general intensity transformations, such as contrast enhancement [28].
3. *Sources of prior information.* All segmentation algorithms make assumptions about the imaged scene. Common assumptions relate to the intensity and to the general shape of the objects in the scene. However, segmentation algorithms differ in the way that they derive these assumptions. Supervised segmentation algorithms construct representative prior models of the imaged scene from training images for

which users have defined regions of interest. Unsupervised segmentation algorithms use mostly information that is intrinsic to the images, and do not rely on user-defined regions of interest [21]. Because my work assumes considerable variability in the form of expected target structures, the work precludes the use of supervised methods; therefore, this review will concentrate mostly on unsupervised methods.

4. *Representation of uncertainty.* Due to the effects of random processes—for example, measurement noise—on image acquisition, uncertainty is inherent in the interpretation of pixel intensities. Many segmentation algorithms account for uncertainty explicitly through stochastic or fuzzy-logic representation of pixel information. The stochastic schemes assign values to pixels according to a probability distribution. The fuzzy schemes assign plausible values that may hold simultaneously.
5. *Problem formulation.* There is no unified theory of segmentation of medical images; consequently, many ad hoc algorithms have been developed [25]. Ad hoc algorithms are difficult to compare and generalize. Researchers have proposed alternatives to ad hoc methods by modeling segmentation as optimization problems in various domains. Examples are movement of particles until they reach an optimal state [29], and activation of neural networks until they converge to a steady state [30].

In Sections 2.2 and 2.3, I describe common approaches to boundary detection and to region detection, respectively. I use the following notation in the discussion. An image is a mapping $f : N \times N \to I$, where N is the number of rows and columns in the image, and I is the range of possible intensities. I use $f(x, y)$ to denote the intensity of pixel (x, y), where $1 \le x, y \le N$.

2.2 Boundary-Detection Methods

I considered two boundary-detection methods that rely on gradient operators: convolution with operators and deformable models.

2.2.1 Convolution of the Image with Gradient Operators

The simplest boundary-detection algorithms are based on convolution of the image with gradient operators. A gradient operator is defined as

$$\nabla f(x, y) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j , \qquad (2.1)$$

where f is the image, i and j are positive unit vectors in the x and y directions, respectively,

$$|\nabla f(x, y)| = \left(\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right)^{1/2} \qquad (2.2)$$

is the magnitude, and

$$\operatorname{atan} \left\langle \frac{\left(\frac{\partial f}{\partial y} \right)}{\left(\frac{\partial f}{\partial x} \right)} \right\rangle \qquad (2.3)$$

is the orientation of the gradient. Many gradient operators are discrete variations of Equation 2.1; they typically detect boundaries by responding with large magnitude (see Equation 2.2) to abrupt intensity changes, and with small magnitude to gradual intensity changes. Example operators are the Roberts, Sobel, and Perwitt operators [17], all of which respond with large magnitudes to gradients at specific orientations (see Equation 2.3). Gradient operators are sensitive to noise; therefore, researchers often combine them with other methods that compensate for noise.

Canny, [31] proposed a technique for finding optimal operators for detection of edges of particular forms (as indicated by edge models) from noisy images. Canny identified three desirable properties for an edge detector: good *detection*, as indicated by a low rate of false responses; good *localization*, as indicated by a small difference between the detected edge and the true edge; and occurrence of a *single response*. Canny provided a mathematical formulation of these properties, and used the formulation to guide a search for an operator that complied optimally with the desirable properties.

Canny defined good detection as high signal-to-noise ratio (SNR), where the SNR is the quotient of the detection filter’s response to an edge and the filter’s root-mean-squared response to noise only, and good localization (LOC) as a small root-mean-squared distance of the detected edge from the center of the model edge with an assumption of Gaussian noise. He showed that the design of an optimal gradient operator reduces to simultaneous maximization of SNR and minimization of LOC subject to the third desirable property, which constrains the number of responses to a boundary and thus reduces the probability of error. The complexity of the preceding optimization prohibits an analytical solution in most practical cases; therefore, Canny proposed the use of numerical methods to obtain optimal operators.

Often, convolution with gradient operators results in a depiction of many small edges, or edge elements, that do not necessarily correspond to boundaries of salient objects. For practical applications, edge elements must be linked together such that they fit boundaries of objects of interest. Gonzalez and Woods [17] describe several techniques for linking edge elements, including the Hough transform, which searches for intersections of lines that correspond to candidate edge points in parameter space, and graph-search techniques, which represent the layout of edge elements in images with graphs, then perform heuristic search or dynamic programming to find optimal links on the graphs.

Convolution with gradient operators does not account for variable effects of partial volume, which are prevalent in images of tortuous vessels. In Section 2.2.2, I describe a popular boundary-detection method that circumvents the effects of partial volume by eliciting shape information from the user.

2.2.2 Deformable Models

Energy-minimizing deformable models identify object boundaries that reside in proximity to an initial shape model. Kass and colleagues [32] proposed using **snakes**, or energy-minimizing splines, to model dynamic contours that can conform to object shapes in response to a set of forces and constraints. Snakes represent spline modifications as a response to the activation of three forces within an associate energy function. Given a parametric formulation of the position of a snake $v(t) = (x(t), y(t))$, which specifies x and y coordinates as a function of the parameter t, the energy function is

$$E = \int_0^1 E(v(t))\,dt = \int_0^1 [E_{\text{internal}}(v(t)) + E_{\text{image}}(v(t)) + E_{\text{constraints}}(v(t))]\,dt,$$

where E_{internal} represents the internal energy of the spline that results from bending; E_{image} represents forces that attract the snake to salient image features, such as boundaries; and $E_{\text{constraints}}$ represents external energy forces.

The internal energy term, $E_{\text{internal}} = \frac{1}{2}(\alpha(t)|v'(t)|^2 + \beta(t)|v''(t)|^2)$, is composed of weighted first-order and a second-order derivatives of $v(t)$; these derivatives impose smoothness constraints on the spline. The two weights $\alpha(t)$ and $\beta(t)$ control the relative importance of the two derivatives. For example, a weight $\beta(t) = 0$ allows the snake to become second-order discontinuous, and to take the shape of a corner.

The image energy term is a weighted combination of three energy functionals:

$$E_{\text{image}} = W_{\text{line}}E_{\text{line}} + W_{\text{edge}}E_{\text{edge}} + W_{\text{termination}}E_{\text{termination}}.$$

For example, $E_{\text{line}} = f(x, y)$, where $f(x, y)$ is the intensity of the pixel at coordinates (x, y), attracts the snake to the brightest or darkest nearby contour, depending on the sign of the corresponding weight and on the other constraints. $E_{\text{edge}} = -|\nabla f(x, y)|^2$ attracts the snake to large image gradients. $E_{\text{termination}} = \frac{\partial\theta}{\partial n_{\perp}}$, where θ is the gradient angle in a smoothed version of the image and $n_{\perp}$ is a unit vector perpendicular to the gradient direction, represents the curvature of the contour, and attracts the contour to line terminations.

The constraints energy term incorporates the effects of two forces: springs and volcanos. *Springs* simulate attraction forces between two points, such as $-k(x_1 - x_2)^2$, where x_1 and x_2 are the two points, one positioned on the spline and the other in the vicinity of the spline, and k is a constant. *Volcanos* simulate repulsion forces, such as $\left(\frac{1}{r}\right)^2$, where r is the distance between the volcano and points on the spline, and $\frac{1}{r}$ is clipped near $r = 0$ to prevent numerical instability (hence the name *volcano*). Researchers use several numerical optimization methods to solve for the spline positioning that minimizes E. These methods are based on formulations such as finite differences, finite elements, dynamic programming, and gradient decent [33].

Other deformable models are reported in the literature. Szekely and colleagues [34] demonstrate the use of flexible Fourier surface models to segment MR brain-volume data. McInerney and Terzopoulos [35] present results of segmentation of arterial trees of the retina from conventional angiograms with topologically adaptable contour models. Chuang and Kuo [36] use wavelet deformable models to segment the contours of the heart from CT images. Common to all deformable models is the integration of a variety of image features into a single formula. Limitations of deformable models are their sensitivity to initial conditions and their dependence on many adjustable parameters.

2.3 Region-Detection Methods

Sections 2.3.1 to 2.3.3 contain descriptions of three region-detection methods that I considered for this work: threshold-based relaxation, feature-based classification, and random-field models with a Bayes’ formulation.

2.3.1 Threshold-Based Relaxation

I described the simplest region-detection method—intensity thresholding—in the discussion about current techniques for editing CTA volumes in Section 1.2.3. One disadvantage of intensity thresholding is that the results are sensitive to noise. To compensate for noise, we usually complement thresholding with a smoothing operation. Typically, smoothing operators trade off degree of smoothness for preservation of small structures. Smoothing of threshold images can be achieved via repetitive application of smoothing operators, such as a median filters [17].

Sahoo and colleagues [15] described a threshold-based method that compensates for noise by adjusting labels of pixels to labels of pixels' neighbors via probabilistic relaxation. The method uses a probabilistic representation of the possible labels for a pixel. It updates probabilities based on the following relation r_{ij} between labels $l, l' \in L$ of neighboring pixels i, j . The relation is based on pixel *dependence*, as determined by the connectivity of pixels, and on label *compatibility,* as determined from the degree of similarity between labels of pixels.

$$r_{ij}(l, l') = \begin{cases} -1 & l, l' \text{ - incompatible,} \\ \ \ 0 & i, j \text{ - independent,} \\ \ \ 1 & l, l' \text{ - compatible.} \end{cases}$$

For example, labels of unconnected pixels (and thus independent pixels) are unrelated; labels of connected pixels are positively related if they are similar (or compatible), and are negatively related if they are different (or incompatible). Following an initialization through conventional thresholding, a relaxation process examines, for all pixels, the dependencies between the pixels and their neighbors, and updates the probability distribution of the pixels' labels based on probability distributions for labels of adjacent pixels. An update rule increases the probability of labels that are positively related to high-probability labels of neighboring pixels, and reduces the probability of labels that are negatively related to high-probability labels. The probability of labels does not change if the labels are positively related to low-probability labels. Lin and colleagues [30] described a neural-network formulation of the threshold-based relaxation method.

2.3.2 Feature-Based Classification

Image segmentation can be viewed as a classification task, where pixels are assigned to classes in feature space that correspond to salient objects in the imaged scene. Segmentation via classification includes several steps. First, the algorithm defines classes by partitioning the feature space into mutually exclusive clusters, and designating a label for each cluster. Second, the algorithm derives a feature vector for each pixel, and searches for the closest cluster according to distance criteria. Third, the algorithm assigns to the pixel a label that is identical to the label of the closest cluster. Features that are commonly used for classification include point measurements (for example, intensity—thus, intensity thresholding could be viewed as a special case of feature-based classification); aggregate measures based on a pixel's neighborhood (for example, mean and standard deviation of intensity); measurements resulting from application of differential operators to local neighborhoods [23] (for example, intensity gradients); and multispectral measurements (for example, MR data [21]).

We can use either supervised or unsupervised methods to determine the classes for classification from a set of points in feature space and an indication of the number of target classes. Supervised and unsupervised methods differ mainly in that the former involves user judgment, whereas the latter does not. Supervised methods have the user define a set of classes from a training set, and then estimate characteristics of classes. If the data in the training set are highly variable, the estimated characteristics may not generalize well beyond the training set and may yield poor classification. Unsupervised methods operate solely on the data themselves, and search for clusters within the data.

An important consideration for clustering is selection of features that contribute significantly to discrimination of clusters; such features typically are independent and nonredundant. Coleman and Andrews [37] describe a procedure for obtaining optimal features. They suggest decorrelating features with the Karhunen–Loeve transform [38], which performs a multidimensional axis rotation of a given set of features into a set of orthogonal features. They further suggest evaluating the classification in terms of the interclass and intraclass variance, and selecting a target number of clusters that optimizes a measure of these variances.

Given a set of features and a target number of clusters, the following procedure finds the clusters. First, the procedure selects initial cluster means arbitrarily, and assigns feature vectors to the mean that is nearest to them. Second, it computes new cluster means based on the preceding assignments. Third, it continues feature-vector assignment and recomputation of means until convergence—that is, until the assignment of vectors to clusters does not change in successive iterations.

Often, regions that result from pixel classification do not correspond well to salient objects in the imaged scene. Regions may exhibit a level of granularity that is either overly fine or not fine enough, relative to the granularity level of regions of interest. Spliting and merging of regions can adjust the granularity to suit that of regions of interest. Here, I describe a split-and-merge algorithm that uses region homogeneity as a criterion for granularity adjustments; I adapted the description from Strasters and Gerbrands [39].

The split-and-merge algorithm assumes a region-adjacency graph, where nodes represent regions in an image and arcs represent spatial relationships among the regions. The algorithm first examines the graph to identify potential regions for merger. The algorithm joins regions whose merger complies with the homogeneity criteria, and replaces graph nodes that correspond to merged regions with a single node. Then, the algorithm splits inhomogeneous regions to form new regions, and replaces the corresponding nodes in the graph with nodes that correspond to the new regions. This procedure continues until all nodes in the graph represent homogeneous regions, and no more region can be merged.

Strasters and Gerbrands [39] described several homogeneity criteria that have been used with split-and-merge algorithms. Among them are a *min–max criterion*, which considers a region to be homogeneous if the differences between extreme intensities are within a specified bound; a *thresholding criterion*, which examines whether the intensities in a region fall within a specified range; and a *variance criterion*, which limits the allowable variance within a homogeneous region. A disadvantage of spliting and merging of regions based on these criteria is limited robustness, because the final results depend on the order with which the regions are considered.

2.3.3 Random-Field Models and Bayes' Formulation

Dubes and Jain [40] surveyed segmentation algorithms that use random field models to represent prior information about an image probabilistically, so that Bayes' decision theory could be applied to the segmentation problem. Here I describe the general approach taken by these algorithms. Given M pixels in an image, a random-field model is a distribution for a random vector X of size M, where each random variable X_t represents the label of pixel t, $1 \le t \le M$. Segmentation assigns labels from the set L to all X_t. The objective is to find the optimal pixel labeling $x^* = \{x^*_1, x^*_2, \ldots, x^*_M\}$, given a set of pixel observations $Y = \{Y_1, Y_2, \ldots, Y_M\}$ and a prior Markov random field that models the statistical dependence among neighboring pixels $P(X = x)$. The Markov random field must be realized from a parametric model through a random sampling process, and must satisfy the following proprieties for $S|t$, the set of all M pixels excluding pixel t, and for ∂t, the set of pixels in the neighborhood of t excluding pixel t:

1. *Positivity:* $P(X = x) > 0$ for all x, and for all sample spaces Ω.
2. *Markov property:* $P(X_t = x_t | X_{S|t} = x_{S|t}) = P(X_t = x_t | X_{\partial t} = x_{\partial t})$.
3. *Homogeneity:* $P(X_t = x_t | X_{\partial t} = x_{\partial t})$ is the same for all pixels t.

The segmentation algorithm assumes that observed random variables are conditionally independent given the true labels. Thus, it can use the maximum a posteriori (MAP) method to choose the estimate $\hat{x}$ that maximizes the posterior probability of $X = \hat{x}$ given $Y = y$, where

$$P(X = x | Y = y) = \frac{P(Y = y | X = x)\, P(X = x)}{P(Y = y)}.$$

Although the marginal conditional distribution of Y_i given the true label $X_i = x_i$ can have any form, typically applications of the segmentation algorithm assume the distribution to be normal, with mean μ_{x_i} and variance $\sigma^2_{x_i}$. Researchers used several optimization methods to solve MAP for segmentation; such methods include stochastic relaxation [41] and related approximations [40].

The Bayesian approach to segmentation is appealing because it models the uncertainty inherent in image acquisition through a rigorous formulation. However, the approach assumes that a user could sample a representative distribution to derive prior Markov random fields, and thus applicability of the method is restricted to where generalizable prior information is available.

2.4 Summary of 2D Segmentation Methods

Many papers report sequences of segmentation methods for particular applications. For example—researchers have combined boundary-detection methods and region-detection methods to avoid disadvantages and to benefit from advantages of each. Gauch and colleagues [33] developed a sequential hybrid deformable model that first minimized a region-detection function to obtain initial boundary estimates, then minimized a boundary-detection function to obtain smooth, closed, object boundaries. Chakraborty and Duncan [42] developed an iterative hybrid method that modeled segmentation as a cooperation task between two modules, and used game-theory principles to coordinate operations of the modules to achieve common segmentation goals. The method decomposed segmentation into subtasks: One of the modules performed region-detection segmentation through maximization of a MAP function based on a Markov random-field model, whereas the other module detected boundaries by activating a deformable contour model. The output of each module was fed to the other module after every iteration. The procedure terminated when both modules had converged to their optimal states.

Table 2.1 relates the preceding methods to one another, based on the five-dimensional characterization that I proposed in Section 2.1. Note that many advanced versions of these algorithms may have characteristics slightly different from those of the generic versions.

Table 2.1 Mapping of segmentation algorithms by characteristics.

Segmentation algorithm	Target entity	Features extracted	Source of prior information	Representation of uncertainty	Problem formulation
Convolution of image with gradient operators	boundary	derivatives	supervised boundary model	deterministic	filtering
Deformable models	boundary	pixel values, derivatives	supervised boundary model	deterministic	filtering with optimal operators
Threshold-based relaxation	region	pixel values	supervised compatibility relations	stochastic	relaxation
Feature-based classification	region	depends on implementation	supervised training set or unsupervised clustering	deterministic	optimization
Random-field models and Bayes' formulation	region	pixel value	supervised probability distribution	stochastic	optimization

2.5 Extensions of 2D Segmentation to 3D

The segmentation methods that I described in the preceding sections are readily extensible to 3D in cases where the assumptions that we make for 2D segmentation apply also to 3D datasets. In the context of 3D processing, we are interested in depicting coherent surfaces or coherent regions whose extension is 3D. Examples of 3D extensions of 2D algorithms include 3D connectivity, where the search for connected voxels to include in the segmented structure is performed in 3D [2, 16], and 3D deformable models, where the initial conditions are set and the searches for optimal solutions are performed in 3D [43].

Medical image volumes are amenable to segmentation with 3D methods, which process voxel arrangements in the same manner independent of orientation, or with 2D methods, which process 2D sections independently and then combine the results to form the segmented volume. Two considerations are relevant to the decision of whether to perform volume segmentation in 2D or in 3D: the volume resolution, and the complexity of the algorithm. 2D algorithms are likely to produce more accurate results when the volume resolution is anisotropic and the resolution for one axis is significantly lower than the resolution for the other two axes. 2D algorithms are likely to be more practical if the complexity of the algorithm in greater than linear in the number of voxels.

2D methods may determine the segmentation of a single section using information from several adjacent sections. The rationale for augmenting information from a given section with information derived from adjacent sections is based on the assumption that coherence of anatomical organs manifests as coherence of their images in consecutive cross sections. Information from adjacent sections is often relevant to the segmentation task within a particular section. Although there may be many ways to represent structure coherence in consecutive sections [44], the generalized cylinder model [45], which represents coherence in anatomical objects implicitly, pertains most closely to this work.

Soroka [46] and Shani [47] demonstrated the utility of generalized cylinders for semiautomated recognition of the heart and kidney, respectively. Kitamura and colleagues [48] and Fessler and Mackovski [49] demonstrated the usefulness of generalized cylinders for semiautomated reconstruction of arterial trees from X-ray images and from MR angiograms, respectively. In these studies, the utility of generalized cylinder models was limited to areas where the medial axis of the organ was not parallel to the image plane. The first three papers pointed out the difficulties in applying the generalized cylinder model where the organ cross sections were not ellipses. Fessler and Mackovsky, however, handled non-elliptic vessel-branch cross sections by modeling bifurcations explicitly as intersecting ellipses.

The evaluation of segmentation algorithms is as important as the development of the algorithms. In Section 2.6, I discuss current approaches to evaluation of segmentation algorithms.

2.6 Evaluation of Segmentation Algorithms

To evaluate a segmentation algorithm, we must determine performance criteria and performance measures, design an experiment, and conduct the experiment. Although evaluation of segmentation algorithms is a prerequisite to their use in medical imaging applications, the literature includes only a few reports of such evaluations [50, 51]. Several researchers have pointed out difficulties in evaluation of segmentation algorithms. Pratt [25] stated that comparative evaluation is difficult because of the large number of methods and the differences in the kinds of information that they provide, because of difficulties in determining the optimum parameters associated with each technique, and because of the lack of definitive performance criteria. Lee and colleagues [52] argued that a performance criterion suitable for one application may not be suitable for another application, and that the results could depend on the criteria chosen, on object size, on contrast in the images, on variance in object and background, and on noise. They also indicated that interpretation and generalization of the results may not be straightforward. Chalana and Kim [53] pointed out difficulties due to the lack of definitive gold standards, and due to the lack of standardized statistical protocols.

In Sections 2.6.1 through 2.6.4, I describe the elements critical to plans for evaluation of segmentation algorithms.

2.6.1 Performance Criteria

Several performance criteria are relevant to evaluation of segmentation algorithms:

1. *Accuracy*: The closeness of the results of segmentation to the true partitioning of the image scene
2. *Precision:* The degree to which repetitive applications of the segmentation algorithm under similar conditions produces the same result
3. *Computational complexity*: The dependence of the required processing time and memory requirements on the input size
4. *Usability:* The extent to which the algorithm is easy to use

There are standard methods for determining computational complexity [54]. Measurements of accuracy, precision, and usability, however, pose a challenge for the designer of an evaluation, because there are many different strategies to measure performance according to these criteria. I discuss two different evaluation strategies in Section 2.6.2.

2.6.2 Evaluation Strategies

We can evaluate the accuracy, precision, and usability of a segmentation algorithm qualitatively or quantitatively. Qualitative evaluation demonstrates the merits of a segmentation algorithm. Typically, subjects observe the results, and then comment on the results' quality, or they accomplish a task using the results, and then the researcher draws inferences about the quality of segmentation from the quality of the performance of the task [55]. Qualitative evaluation can be laborious if many images need to be evaluated. Also, because human judgements might not be consistent, qualitative evaluation requires multiple observations by the same observer at different times or by different observers. When evaluation involves accomplishment of a task, it may be difficult to isolate the effects of the segmentation algorithm on task performance from the effects of other factors. Despite its limitations, qualitative evaluation is the only way to evaluate the *usability* of an interactive segmentation algorithm. For evaluation of accuracy and precision of segmentation, quantitative evaluation is a viable alternative.

Quantitative evaluation involves objective measurement of the results produced by a segmentation algorithm on one or more representative datasets. Most quantitative evaluations measure the difference between the regions portrayed in segmentation results and the regions derived from predefined gold standards. This measurement poses two challenges. First, gold standards for live organs are unobtainable in practice; therefore, we must estimate organ measurements to substitute for gold standards. Second, there are many ways to measure region differences, and it is important to select measures that are reliable and valid with respect to the application [56]. **Reliability of a measure** is the extent to which the measure results in the same value when we use the measure repetitively to evaluate a single object. **Validity of a measure** is the extent to which the measurement is free from a consistent bias with respect to the target object.

In Sections 2.6.3 and 2.6.4, I elaborate standards and measures for evaluation.

2.6.3 Standard Data for Evaluation

I define three types of **evaluation sets**, or datasets with standard measurements of target organs, to assess the performance of a segmentation algorithm.

1. *Patient data.* The dataset includes a sequence of patient sections. Experts produce standard measurements by tracing regions for each section individually. To reduce the effects of human error, several experts derive standard regions for which they can reach a consensus.
2. *Phantom data.* The dataset includes a sequence of sections that researchers acquire through scanning of a physical phantom that models the target organ. Experts obtain gold-standard measurements from the phantom specifications and scans.
3. *Simulation data.* The dataset includes a sequence of simulation sections. Experts obtain gold-standard measurements via calculations of objects' sections from object models and simulation parameters. Users of a simulation program can con-

trol the extent to which simulated data are realistic by setting image-acquisition parameters—for example, noise level—and by constructing a scene model of arbitrary complexity.

The ability to make predictions about the performance of a segmentation algorithm on future patient cases from the algorithm's performance on an evaluation set depends on the extent to which the datasets are representative of the target-application data. Patient data are good representatives of target-application data, because they include real effects introduced during acquisition of images of patient organs. Data from phantom scans may also include real effects introduced during image acquisition; therefore they may represent accurately the image-acquisition process used to produce patient data. However, frequently, phantoms are simple constructions that might help demonstrate the general feasibility of a method, but that do not necessarily indicate applicability of the method to patient data. Simulation data are the most simplified representations of patient scans because they do not involve true scanning, and because the scene typically comprises geometrical objects.

The preceding analysis demonstrates that no evaluation set is superior to the others. Simulations and phantom scans are likely to be more useful than patient data for evaluation of segmentation algorithms at early research stages because of their simplicity, because of the ability of the user to control acquisition and model parameters, and because of the ability to obtain a gold standard. Patient data are likely to be more useful than simulations and phantom scans at later research stages, when utility of a segmentation algorithm for clinical applications is a major concern.

Since regions and their boundaries uniquely define each other, the results of comparing regions and those of comparing boundaries should be similar; researchers can infer the results of one of these comparison from the results of the other. My work includes comparison of regions and boundaries in two places. The procedure for identification of salient regions (see Chapter 1, Section 1.4) matches isolabel contours, which are region boundaries, to determine transitions from inside an object into the background. The evaluation (see Chapter 7) compares computer-generated regions to human-generated regions to see when they are compatible. In Section 2.6.4, I discuss measures that have been reported in evaluation studies for determining the difference between regions portrayed in segmentation results. In Section 4.2, I describe measures of contour similarity that appear in the literature in the context of shape matching. Although the similarity measures in the two section appear in different contexts, both could be used interchangeably.

2.6.4 Evaluation Measures

Yasnoff and colleagues [55] described a set of desirable properties for measures used in quantitative evaluation:

1. The magnitude of measures should reflect the difference between the segmented region and the standard to which that region is compared.
2. Measures should support categorization by pixel label. Such a categorization could provide insight into biases that apply to only particular objects.
3. Measures should allow weighting by importance of errors in different regions—for example, errors in object versus background regions, or errors at the center of an object versus errors at the boundaries.
4. Measures should be invariant to image size.

I discuss several measures that I considered for the evaluation of this work. For this discussion, I assume that segments are designated by having all their pixels tagged with a similar label, which is different from labels of adjacent regions. I define a **misclassification** as a label assignment for a pixel that is different from the label of the corresponding pixel in the standard image.

2.6.4.1 Least-Square Error

Pratt [25] summarized common measures for assessing the differences between two images. From these measures, the least-square error (LSE) is most suitable for comparison of a segmented image and a gold-standard image. The LSE measure is defined by

$$e_{LSE} = \frac{1}{NxN}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|f(i,j) - \hat{f}(i,j)\right|^2,$$

where $f(i,j) \quad 1 \le i, j \le N$, is the gold-standard image and $\hat{f}(i,j) \quad 1 \le i, j \le N$ is the segmented image. The normalized LSE is defined by

$$e_{NLSE} = \frac{\frac{1}{NxN}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|f(i,j) - \hat{f}(i,j)\right|^2}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left|f(i,j)\right|^2}.$$

LSE is used often because it is simple to compute. However, the measure does not always correspond well to human subjective testing. To improve correspondence of results of evaluation with results of human testing, researchers have proposed several methods.

2.6.4.2 Correlation

Bryant and Bouldin [57] suggested using normalized correlation to determine the similarity between an image and a gold standard. The normalized correlation is defined by

$$R(u,v) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} f(i,j) \cdot \hat{f}(u+i-1, v+j-1)}{\left[\sum_{i=1}^{N}\sum_{j=1}^{N} f^2(i,j)\right]^{\frac{1}{2}} \cdot \left[\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{f}^2(u+i-1, v+j-1)\right]^{\frac{1}{2}}},$$

and is limited to the range [0,1]. Assuming that the images are aligned, we compute the value $R(1,1)$. Otherwise, we compute the value for various displacements, and pick the displacement that yields the highest correlation value. Correlation is simple to implement, but it has several draw backs:

1. Correlation does not distinguish between erroneous segmentation at the boundaries of objects and at the center of objects.
2. Correlation considers erroneous segmentation for all classes as equally important.
3. Correlation provides no information about possible causes for the error.

2.6.4.3 Rate of Misclassification

The rate of misclassification (ROM) is the weighted sum of misclassified pixels [52]. Yasnoff and colleagues [55] demonstrated that confusion matrices can help us to differentiate between errors for different classes and thus could help us to identify the source of errors. **Confusion matrices** tabulate the true classification of voxels by the observed classification, and thus allow us to distinguish between misclassification of object pixels as background and misclassification of background pixels as object. We can use information obtained from the confusion matrices to weight errors according to the latter's expected importance.

2.6.4.4 Pixel-Distance Error

Pixel-distance error (PDE) takes into account the relative positioning of segmentation errors within an object. The use of PDE is based on the assumption that the amount of error for a segmentation is related to the distance between the misclassified pixels and other regions that have a similar correct classification. The PDE is defined as

$$PDE = 100\frac{\sqrt{\varepsilon_T}}{A},$$

where

$$\varepsilon_T = \sum_{m=1}^{k} d_m^2,$$

k is the number of misclassified pixels, d_m is the Euclidean distance of the mth pixel to the nearest segment with a similar correct classification, and A is the total number of pixels in the image. Like ROM, PDE can be presented in a confusion matrix to allow weighting of error by importance.

2.6.4.5 Comparison of Rate of Misclassification and Pixel-Distance Error

Yasnoff and colleagues [55] reported two experiments that assessed how well ROM and PDE correlated with human observers. The first experiment used the two measures and human observers to compare results of different segmentation algorithms. The results were approximated from results of an automated segmentation procedure by controlled distortions of the segmentation results. The second experiment used the measures and human observers to compare pairs of segmentation results and gold standards for different images. For the two experiments, the investigators computed the correlation between each of ROM and PDE and an average of observer scores, and between averages of two subgroups of 10 observers—to see how the first two measures compared to interobserver averages. The investigators concluded that both ROM and PDE correlated with human observers, and that distance to correct classification and shape were both important for assessing severity of segmentation error.

2.6.4.6 Importance Weighting of Errors

Straster and Gerbrands [39] proposed a **figure of merit (FOM)** for classification; it illustrates how weighting by importance can be combined with PDE:

$$FOM = \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{1 + \gamma d^2(i, j)},$$

where $N \times N$ is the number of pixels in the image, $d(i, j)$ is the Euclidean distance of pixel (i, j) to the nearest segment with a similar correct classification, and γ is a scaling constant. A correct segmentation will yield a FOM of 1. We can use γ to change the contribution of particular errors to the FOM. The FOM assumes that importance weights are associated with pixels based on their adjacency to critical structures, as determined by the gold standard.

2.6.4.7 Operator Variability

Mitchell and colleagues [58, 59] suggested that an important aspect of the evaluation of an interactive image-analysis system are the effects of the system on interoperator and intraoperator variability. They evaluated a system for quantification of changes in the appearance of multiple-sclerosis lesions from MR images, and measured the influence of different quantification methods on the variability of results. They performed an analysis of variance to determine the standard error of measurement and the variability of both manual traces and results that operators produced with the assistance of an automated algorithm. The interoperator and intraoperator variability measures generalize to evaluation of interactive segmentation algorithms like those of IVSeg.

2.7 Summary

In this chapter, I described the alternatives that I considered for editing of CTA volumes. I expected that none of the approaches that I encountered in reviewing the literature would allow Stanford radiology experts to extract objects without incorporation of prior models, and without having leaks into adjacent structures of similar appearance. From my literature review and from early experience with segmentation algorithms, I concluded that an approach based on fine multilevel thresholding could exploit characteristics inherent in image acquisition to elicit the generic shape of an organ from the image itself, and could use the elicited shape information to detect transitions from object to background robustly. In this review, I also discussed approaches to evaluation of segmentation algorithms that I considered when I designed an evaluation plan for IVSeg.

In Chapter 3, I explain the basic properties of imaging systems that constitute the foundation for IVSeg.

Chapter 3

Principles That Affect Images of Boundaries

Imaging systems generate images of objects by measuring objects' physical properties at sample points and by reconstructing the measurements into coherent views [60]. Images of objects can be viewed as a superposition of an infinite number of blurred images of infinitesimal object points, which result from the system's **point-spread function (PSF)**: the response of the system to the points. Due to the broadened intensity distributions of point images, these images may overlap and thus may interfere with images of neighboring points. Various noise sources and artifacts cause additional interference, resulting in images that are imperfect representation of the imaged scene. Although imaging systems apply correction filters as part of the reconstruction process, the imperfections cannot be eliminated completely. Consequently, segmentation algorithms must be designed to find correct region boundaries despite these imperfections.

One source of artifacts in medical images is the partial-volume effect (PVE). IVSeg exploits the PVE and the gradual intensity gradients that result from PVE to perform segmentation in the following way. IVSeg enhances conventional thresholding to produce patterns of similar quasi-concentric contours along the boundaries of object images. Then, it compares the shapes of contours to detect transitions from within objects into the background. A premise that underlies this approach to segmentation is that, independent of the form, size, position, and orientation of objects, contours that enclose voxels from a single object are likely to be similar to one another, and contours that enclose voxels from different objects are likely to be dissimilar. The premise is based on observations of the outcome of partitioning the gradual intensity gradient that is associated with object boundaries.

In this chapter, I explain the origin of the gradual intensity gradient associated with images of object boundaries, and demonstrate how it might impede automated segmentation of images. In Section 3.1, I describe factors that affect the value of voxels. Most of the discussion pertains to medical images independent of imaging modality; I provide examples from CT scanning. In Section 3.2, I present the results of CT simulations that I conducted to illustrate the aggregate effects of the concepts presented in Section 3.1.

3.1 Factors That Affect the Value of a Voxel

Several factors affect the value of a voxel in an image: the physical property being measured by the imaging system, the response of the imaging system, random noise, and artifacts.

3.1.1 Physical Property Measured

Different medical imaging modalities measure different physical properties of body tissues. For example, CT images portray the local attenuation of X-ray beams, MR images portray a facet of the local response of hydrogen nuclei in materials that have been placed in a magnetic field to excitation with a radio-frequency pulse, nuclear-medicine images portray the local radiation emitted from organs in which radio-isotope chemicals have accumulated, and ultrasound images portray facets of the reflected high-frequency sound waves transmitted into the body [61].

3.1.2 The Response of Imaging Systems

The response of a system can be defined in the spatial domain or in the frequency domain. The spatial-domain definition specifies how the imaging system transforms an infinitesimally small point into an image with noninfinitesimal dimensions. The frequency-domain definition of resolution specifies how efficiently the imaging system transfers spatial frequencies from a scene into an image.

3.1.2.1 Spatial-Domain Definition

The response of a system to an infinitesimal point is an intensity distribution called the PSF of the system. Image formation can be described as the convolution of the spatial representation of an object with the system's PSF. The PSF is typically a function of system geometry and, for digital systems, of the sampling period that the system uses; it is typically bell shaped. For example, in CT, the PSF is determined by characteristics of the source, detectors, and sampling period. We can estimate the PSF of a scanner by scanning a phantom that includes extremely small objects [12]. The measurement that results from a phantom scan is only an estimate of the PSF, because it also includes the effect of the convolution kernel used to correct for blur in the reconstruction process.

The **line-spread function (LSF)** of an imaging system is the distribution that results from imaging an infinite line of infinitesimal width that is oriented perpendicular to the imaging plane. We can compute the LSF by integrating the contribution from all points along the line. The LSF is related to the PSF by

$$L(x) = 2\int_0^{\infty} P\left(\sqrt{x^2 + z^2}\right) dz ,$$

where $P(x)$ is the PSF (assuming a circularly symmetric system response), and z is the direction perpendicular to the imaging plane. The **edge-response function** describes the image of a sharp contrast boundary in a direction perpendicular to the boundary. We can compute the edge response by convolving the LSF with a unit step function

$$E(x) \equiv L(x) * H(x) \, ,$$

where

$$H(x) = \begin{cases} 0 \, ; & x < 0, \\ \frac{1}{2} \, ; & x = 0, \\ 1 \, ; & x > 0. \end{cases} \, .$$

The LSF is the derivative of the edge response in a direction perpendicular to the edge.

3.1.2.2 Frequency-Domain Definition

The frequency transfer of an imaging system is called the **modulation transfer function (MTF)**. Image formation can be described as a multiplication of the frequency representation of an object with the system's MTF. The MTF effectively filters the amplitude of spatial frequencies in the object representation. The MTF of a system is derivable from the PSF by the Fourier transform. The widths of the MTF and the PSF are inversely related. In Section 3.1.2.3, I discuss this relationship further in the context of spatial resolution.

3.1.2.3 Spatial Resolution

The spatial resolution determines the imaging system's ability to resolve objects in the image from observations. The spatial resolution of medical images can be discussed from both qualitative and quantitative points of view. From a qualitative point of view, the spatial resolution relates to the ability of a human observer to distinguish between two small high-contrast objects that are located close to each other. From a quantitative point of view, spatial resolution specifies the minimum distance between small high-contrast objects that is required for the objects to be resolved when viewed from a displayed image. In the following paragraphs, I discuss factors that affect spatial resolution in CT scanners.

Medical imaging systems reconstruct the measurements that they take—for example, in CT, X-ray attenuation along 1D projections through the object—into 2D matrices of voxels, where voxels represent a finite volume in the corresponding section through the body. Thus, all points in the scanned object that lie within the volume represented by a voxel affect the value of that voxel. It is also possible that values of points within a given voxel will affect the values of other voxels, depending on the PSF of the system response. The spatial resolution of an image is limited not only by the PSF of the scanner, but also by the choice of voxel size, which is set arbitrarily. A voxel size that is larger than the width of the PSF at one-half of maximum value will result in a resolution that is less than the maximum resolution that the scanner allows. In this case, small objects that were detected in the projection data may not appear in the final image.

In the absence of noise, the ability to resolve objects from images depends on the voxel values that result from the objects, and on the spacing of the objects relative to the size of a voxel. Adjacent objects that produce voxels with the same value are not resolvable (Figure 3.1, A). For two objects to be resolvable, the voxels that they produce must be separated by a voxel with a different value (Figure 3.1, B). If the position of two objects with respect to the matrix is such that two voxels exactly cover the objects with an intervening voxel with a different value, then the objects are resolvable. This positioning allows for maximum resolution. However, if the position of the objects is such that two voxels cover each of the objects and the spacing between the objects is equal to the size of a voxel (Figure 3.1, C), as in the previous case, then the objects will not be resolvable because they will produce four consecutive voxels with the same average value. Objects that are smaller than the size of a voxel (Figure 3.1, D) are detected if the signal that they produce is sufficiently different from the background signal. Two such objects are not resolvable if the distance between them is equal to or smaller than the size of a voxel (Figure 3.1, E).

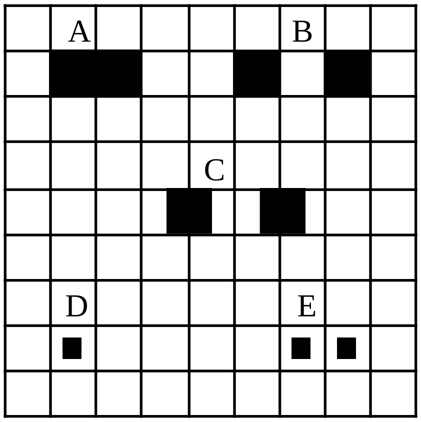


Figure 3.1 A reconstruction matrix. The black rectangles indicate the size and location of objects relative to size and location of voxels. *Source*: Adapted from L.M. Zatz, “Basic principles of computed tomography scanning,” In *Radiology of the Skull and Brain*, Thomas H. Newton and D. Gordon Potts, eds, C.V. Mosby, St. Louis, 1981, page 3865.

The spatial resolution of imaging systems illustrates the relationship between the PSF and MTF. Theoretically, an infinitely narrow PSF and the corresponding infinitely broad MTF yield a perfect replica of the imaged object, because the system exactly transfers infinitely many frequency components. In practice, a narrow PSF and the corresponding broad MTF yield high spatial resolution—the system transfers frequency components with little filtering—and results in images with fine detail. A broad PSF and the corresponding narrow MTF yield low spatial resolution—the system transfers frequency components with extensive filtering and the result is a blurred image.

In Section 3.1.2.4, I explain how the system response combines with the positioning of an object relative to the reconstruction grid to form the PVE.

3.1.2.4 Partial-Volume Effect

The **partial-volume effect** (PVE) occurs when objects are measured by apparatus of finite size. For example, in CT, PVE is caused by the finite size of the source and detectors, and also by the finite size of voxels in the reconstruction grid. Consequently, the computed attenuation for a voxel, at best, is the average attenuation of all the tissues contained in the voxel. The computed attenuation for a voxel may also include contributions from tissues contained in other voxels. In general, all edges in an imaged scene exhibit the PVE, which manifests in gradual intensity changes for voxels in a direction normal to an object's boundary (Figure 3.2).

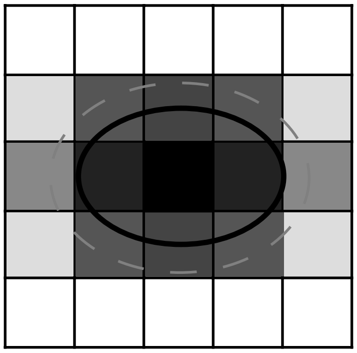


Figure 3.2 Partial-volume effect. The solid ellipse designates object boundaries. The dashed ellipse designates the extent of the intensity gradient due to the systems's point-spread function. The shade of voxels on the reconstruction grid correspond to the partial volumes that they include (shade darkness is inversely related to the amount of partial volume in a voxel).

3.1.3 Random Noise and Streak Artifacts

All imaging systems are subject to random noise at various stages in the imaging process. In CT, there are several sources of noise [62, 63]. A major source of noise is the Poisson process associated with detection of a finite number of X-ray photons. The noise variance is proportional to the reciprocal of radiation exposure to the patient [62]; therefore, we can reduce the noise by increasing the X-ray dose per section. One way to increase the dose within a section is to use larger section widths; however, increased section widths reduce spatial resolution.

In CT, streak artifacts cause degradation of image quality. Streaks result from inconsistencies in the reconstruction of projections due to patient or organ motion, presence of dense objects outside the scan circle, partial volume, and beam hardening [12]. Beam hardening is a result of the polyenergetic nature of x-ray beams, where low-energy photons are attenuated more rapidly than are high-energy photons and therefore the effective energy of surviving photons increases as the beam extends deeper into the body [13].

Section 3.2 provides simulation examples to illustrate how factors that affect the value of a voxels combine in CT. Similar effects can be observed for other medical imaging modalities.

3.2 Examples from CT Simulations

To illustrate the effects of the phenomena that I described in Section 3.1 on the boundaries of object images, I performed several simulations. I simulated the scan of a model of an infinitesimally wide cylinder to demonstrate the response of a scanner. I simulated the scan of a cylinder model of a blood vessel at various orientations relative to the image plane to demonstrate the PVE. I simulated the scan of two close cylinders to demonstrate how PVE might result in object images that are difficult to separate. I generated the simulations with a simulation program that was developed by Carl Crawford and colleagues at General Electric Medical Systems. The simulation program follows three major steps in CT scanning: forward projection, filtering, and backprojection. The program computes ray values by aggregating the contribution of individual *ray lets*, where the user can control the number and spatial distribution of ray-lets per ray. The scanning parameters for the simulations were the following: section thickness was 3 mm; image size was 256×256; voxel size was 0.52 mm; height of source was 0.7 mm; width of source was 1.6 mm; width of detector was 1.844 mm; number of projections was 256; number of detectors per projection was 128. The source comprised 25 ray-emitting points, and the detector comprised 25 ray-detecting points, so 625 ray lets were used per ray.

3.2.1 Line-Spread Function

Figure 3.3 shows a simulation of a CT scan of an approximate line model of infinitesimal width—a thin cylinder—oriented perpendicular to the imaging plane, and with a linear attenuation that, in the absence of PVE, would result in a measurement of 581 HU. Note the bell shape of the intensity profile, and the gradual intensity falloff from the center of the object to the boundary.

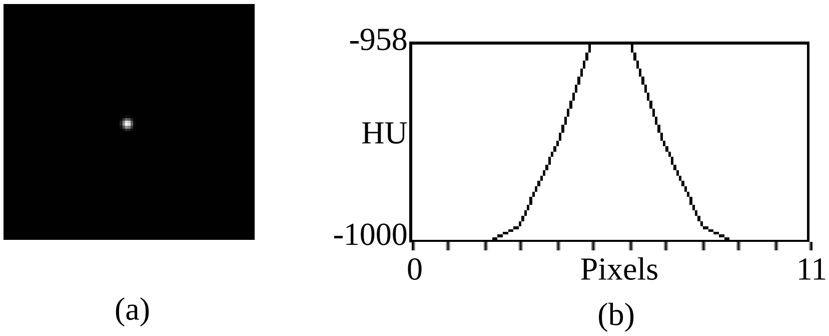
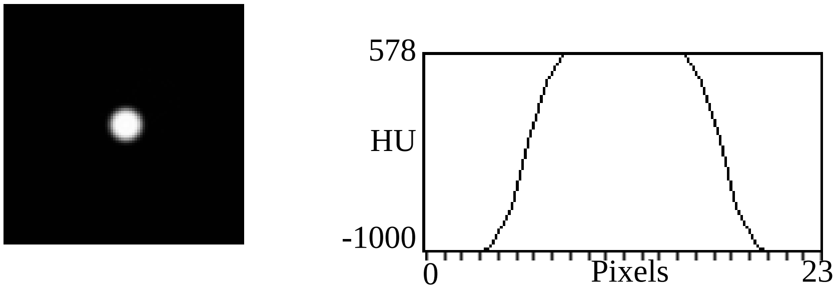


Figure 3.3 An illustration of the line-spread function. (a) The image of a cylinder with a diameter of 0.3 mm. (b) The intensity profile of a horizontal line along 11 pixels across the image of the cylinder at the center. Note that, although the diameter of the cylinder is smaller than the size of a pixel (0.52 mm), the image of the cylinder is nonzero for about 6 pixels, or for about $6 \cdot 0.52 = 3.12$ mm.

3.2.2 PVE in Cylinders

The previous example showed PVE for a thin object that modeled an infinite line. The following examples demonstrate the PVE for object models that are similar to blood vessels, at various orientations relative to the imaging plane, and with a linear attenuation that would result in a measurement of 581 HU in the absence of PVE. Figure 3.4 shows an image of a cylinder oriented perpendicular to the imaging plane. Note the gradual intensity falloff at the edges.



(a)

(b)

Figure 3.4 An illustration of the PVE. (a) The image of a 6-mm-wide cylinder. Note that the section of the cylinder preserves the cylinder's circular shape. (b) The intensity profile of a horizontal line through the center of the cylinder image. The image of the cylinder is nonzero for about 16 pixels along the horizontal line. Note that, although the diameter of the cylinder is 6 mm, the image of the cylinder is nonzero for about $16 \cdot 0.52 = 8.32$ mm (0.52 is the pixel size).

Figure 3.5 shows images of cylinders oriented at two angles relative to the image plane. Note that, as the angle with respect to the image plane decreases, the intensity falloff along the edges of the cylinder is more gradual and the dimensions of the cylinder image are greater in the direction of the slant. These phenomena occur because the number of voxels in the direction of the slant that exhibit partial volume increases with a decrease in slant angle.

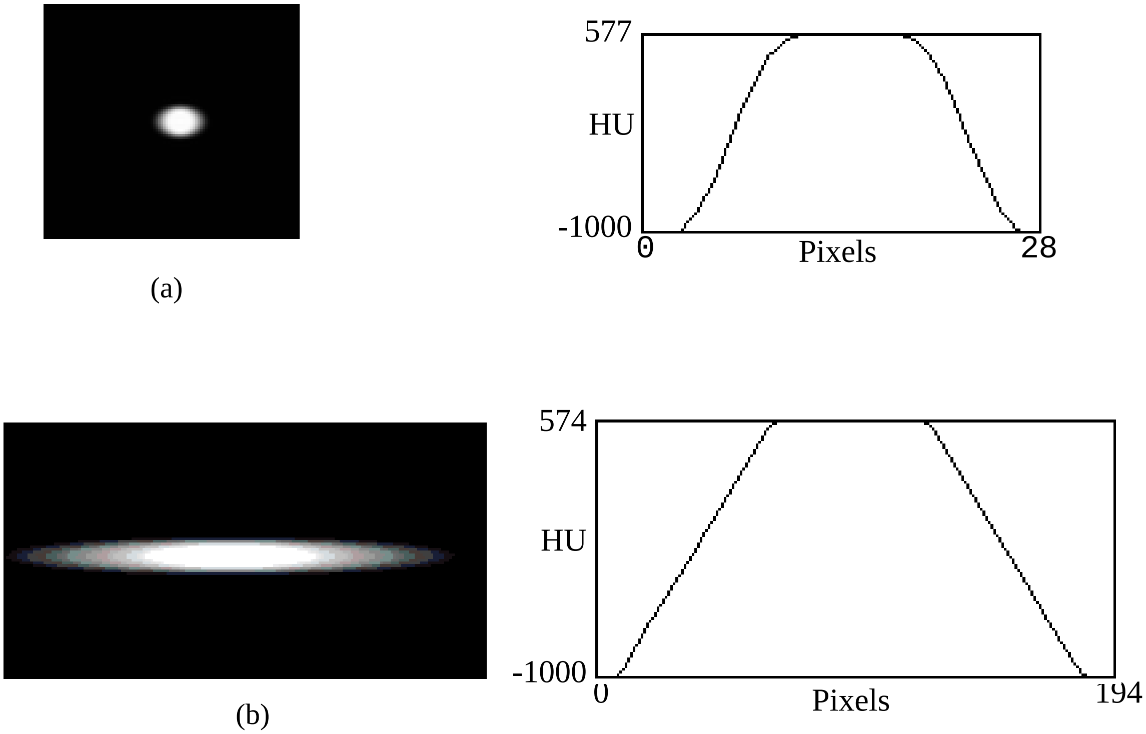


Figure 3.5 Images of 6-mm slanted cylinders and their intensity profiles. (a) Cylinder is oriented at 45° to the horizontal axis. (b) Cylinder is oriented at 5.7° to the horizontal axis. Note that the sections of slanted cylinders are ellipses, and that the lengths of the major axes are inversely related to the slant angle (relative to the imaging plane).

3.2.3 Images of Adjacent Objects

Figure 3.6 shows images of two adjacent cylinders aligned perpendicular to the imaging plane. Note that it is easier to distinguish the objects as the distance between them grows. When the objects are tangent, they appear as a single object in the image, and observers probably would use prior knowledge of each object's shape to predict the locus of boundary points for that object. Even when objects are not tangent, but are extremely close, observers might find it difficult to distinguish between them because of PVE.

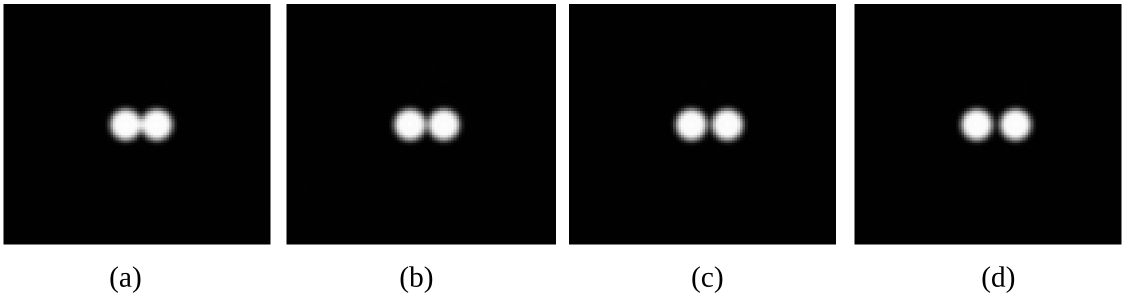


Figure 3.6 Images of two adjacent cylinders, 6 mm in diameter. (a) Cylinders are tangent. (b) Cylinders are positioned 0.5 mm apart. (c) Cylinders are positioned 1 mm apart. (d) Cylinders are positioned 1.5 mm apart. Note that, in image a, it is difficult to determine the boundary of each cylinder.

3.3 Summary

In this chapter, I discussed the appearance of images of object boundaries as gradual intensity gradients in a direction perpendicular to the boundary. I explained how this appearance, which is inherent in image acquisition, originates from principles of imaging systems. I included simulation results to illustrate the appearance of PVE in images, and to show that the gradients of two adjacent objects can combine to cause segmentation difficulties. The simulations also demonstrated that the blur associated with the gradual intensity falloff along the boundary of a cylinder is circular when the cylinder is oriented perpendicular to the imaging plane, and that blur is elliptic when the cylinder is slanted with respect to the image plane. However, independent of the orientation of the cylinder, the regular form of the blur suggested that **isointensity contours**, or contours formed along the boundary of objects through automated tracking of voxels with similar intensity values, have similar shape. Likewise, isolabel contours, which result from fine partition of the intensity range, and which are central to IVSeg’s operation, have similar shape.

In Chapter 4, I describe methods that I considered for comparing the shape of isolabel contours.

Chapter 4

Existing Methods for Contour Comparison

In this chapter, I explain the methods that I considered for comparing isolabel contours to identify salient regions. Researchers have used contour comparison for detection of objects in a variety of applications—for example, object recognition [64] and retrieval of images from databases [65]. In most of these applications, the user provided contour models, and the system found a contour in the image that was similar to the model. IVSeg does not search for objects that are similar to user-defined models; rather, it extracts quasi-concentric contours that surround object images, and compares pairs of adjacent contours to detect a contour that is different from its predecessor and that thus might suggest a transition from inside the object to the background. This difference, however, does not pertain to the contour-comparison task itself; thus I considered existing contour-comparison methods for IVSeg.

The two major factors in an algorithm for comparison of contours are the **contour representation**, which determines what information with respect to the contour will be compared, and the **distance measure**, which is the measure that will be used for comparison. I describe these two factors in Sections 4.1 and 4.2.

4.1 Contour Representation

A program can determine the extent to which contours are similar by computing the difference between coordinates of contours, or by computing the difference between more succinct representations of contours that are formed via feature extraction. Several features may be used to describe contours: position, shape at a variety of scales, size, and orientation. However, not all these features may be relevant for a particular comparison task. A contour-representation scheme must produce different representations for contours whose relevant features are different. A contour-representation scheme must be *invariant* to irrelevant features—that is, it must produce the same representation for objects that differ only in the irrelevant features. For example, for an application that searches for objects in an image database by matching query contours to contours of objects in the images, object size may be irrelevant. In this application, the representation of contours of objects that have similar shapes but different sizes should be similar. Invariant representations are typically obtained via **normalization**, where the characteristics of a contour are modified with respect to a standard. For comparison of contours, the standard often is defined to be one of the contours.

Pratt [25] and Gonzalez and Woods [17] surveyed a variety of methods for representing contours. I include here a description of the representations that I considered for representing contours in IVSeg.

4.1.1 List of Coordinates

A contour can be represented as a **list of coordinates** in the xy plane. This representation is simple and intuitive. Normalizations of position, size, and orientation require translation, scaling, and rotation transformations [18] of the list of coordinates, respectively. For data-intensive applications that require real-time response, the list of coordinates might be prohibitively expensive in computation.

4.1.2 Parametric Cubic Curves

Parametric cubic curves allow us to the derive a compact representation for a contour by approximating the contour with a *piecewise cubic polynomial* curve, or a contiguous sequence of smooth cubic curves [18]. We define each segment Q of the overall curve by a vector of cubic polynomials in the parameter t, $Q(t) = [x(t)\ y(t)]$, where t represents the distance along the curve from a reference point, and where

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x,$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y.$$

We can define $Q(t)$ in matrix notation,

$$Q(t) = T \cdot C, \tag{4.1}$$

where,

$$T = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix},$$

and

$$C = \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix}.$$

To find the polynomial $Q(t)$ for each segment, we use four known parameters to solve for the four polynomial coefficients. We can derive these parameters from geometric constraints on endpoints, tangent vectors, and continuity between curve segments. For example, the splines that IVSeg uses to represent contours (see Section 5.2.3) define geometric constraints on end points. We define C as a function of the constraining parameters $C = M \cdot G$, where M is a 4×4 *basis matrix*, and G is a 4×2 matrix of geometric constraints. Thus, we can rewrite Equation 4.1 as

$$Q(t) = T \cdot M \cdot G, \tag{4.2}$$

from which we observe that the task of finding $Q(t)$ reduces to that of finding M. The method used to find M is different for different kinds of parametric curves. In general, to find M, we need to state variants of the formulation in Equation 4.2 for known values, so M is the only unknown. For more detail, see Foley and colleagues [18]. The error associated with a parametric-curve representation of a given contour is inversely related to the number of segments used to approximate the contour. To obtain normalized contour approximations, we must normalize contour coordinates prior to the derivation of the parametric curves.

4.1.3 Turning Angle

To reduce the dimensionality of a contour's representation, we can represent a contour by simulating a travel along the contour and recording the **turning angle**, which is the direction taken at each step [65]; thus, the turning angle is another example of a parametric representation. The turning angle measures the angle of the tangent to the contour as a function of distance s from some reference point on the contour, and is defined by

$$\Phi(s) = \operatorname{atan}\left[\frac{dy(s)/(ds)}{dx(s)/(ds)}\right].$$

The discrete version of a turning-angles sequence is a piecewise linear chain code [17]. Algorithm developers typically normalize turning-angle representations by performing dynamic programming to register the target contour with a standard contour. This registration task can be costly because of the complexity of dynamic programming.

Researchers proposed reducing the size of a contour's representation by extracting features from the parametric representation at sample points. Example features include the first and second derivatives, and the sign of the curvature

$$k(s) = \frac{d\Phi(s)}{ds},$$

where $\Phi(s)$ is the turning angle. In Sections 4.1.4 through 4.1.6, I discuss additional features that can be used to represent contours.

4.1.4 Spatial Moments

Spatial moments are among the earliest features sets that researchers used to represent contours [66]. The pqth moment of order $(p+q)$ of an $N \times N$ image $f(x, y)$ is defined as

$$M_{pq} = \sum_{y=1}^{N} \sum_{x=1}^{N} x^p y^q f(x, y)\, dxdy \;\; ; \;\; p, q = 0, 1, \dots .$$

For example, for binary images that include a contour of M pixels, where contour pixels are marked by 1, the zero-order moment is equal to the area covered by the contour $M_{00} = M$. The center of mass of the contour has the coordinates $\bar{X} = \frac{M_{10}}{M_{00}}$, $\bar{Y} = \frac{M_{01}}{M_{00}}$. We can use $\bar{X}$ and $\bar{Y}$ to normalize moments with respect to position:

$$\overline{M}_{jk} = \sum_{x=1}^{N} \sum_{y=1}^{N} (X - \bar{X})^j (Y - \bar{Y})^k ,$$

We can normalize moments with respect to size by $\frac{\overline{M}_{jk}}{M}$. Reeves and colleagues [67] reported several additional forms of normalization, and compared the normalized moments' performance on matching of airplane silhouettes. For all experiments, about 18 moments produced the best results. Scassellati and colleagues [65] also used 18 moments to match objects from images in a database with user-defined contours. They compared the performance of several representation schemes, including those described in Sections 4.1.2 through 4.1.4, and found that none of the representation schemes was equivocally better than all others. Although moments allow for compact representations of contours, the relationship between moments and shape is not intuitive, and moments' sensitivity to outliers compromises the robustness of moment-based representations [68].

4.1.5 Fourier Descriptors

Fourier descriptors assume a parametric representation of a closed contour in the xy plane, whose length is $T = 2\pi$. A traversal of the contour more than once yields a pair of periodic projection functions, for the x and y dimensions, each of which can be represented as a Fourier series:

$$x(t) = A_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{2n\pi t}{T} + b_n \sin\frac{2n\pi t}{T}\right) \ ; \ t \geq 0 ,$$

where

$$A_0 = \frac{1}{T}\int_0^T x(t)\,dt ,$$

$$a_n = \frac{2}{T}\int_0^T x(t)\cos\frac{2n\pi t}{T}dt ,$$

$$b_n = \frac{2}{T}\int_0^T x(t)\sin\frac{2n\pi t}{T}dt .$$

The accuracy of representation obtained with Fourier descriptors depends on the number of harmonics at which the series is truncated. Kuhl and Giardina [69] showed how to compute the number of harmonics necessary to guarantee an error bound for a truncated Fourier approximation.

For discrete contours, we can use the **discrete Fourier transform (DFT)** to derive Fourier descriptors. Here, the number of descriptors is limited by the number of contour points. If the application requires only comparison of gross detail, low-frequency descriptors may be sufficient. To use the FFT, if the number of contour points is not a power of 2, we can zero-pad the sequence of points in each dimension to obtain a number of points which is a power of 2. Transformations on Fourier descriptors are derived directly from properties of the DFT (for details, see Gonzalez and Woods [17]).

Reeves and colleagues [67] compared Fourier descriptors to standard spatial moments; they found that, for matching airplane silhouettes, both descriptors produced good results. Both provide a global representation of contours. In some applications, however, local representations are necessary to allow comparison of parts of contours. For example, applications that perform object recognition from aerial photographs by matching contour models must be able to match parts of contours to identify objects of interest that may be occluded by other objects. Section 4.1.6 describes wavelet descriptors, which allow both global and local representation of contours.

4.1.6 Wavelets

The **wavelet decomposition** of a continuous function $g(x)$ is the expansion

$$g(x) = \sum_{i \in Z} \sum_{k \in Z} d_{(i)}(k) \psi(2^{-1}x - k) ,$$

where the basis functions are generated from an oscillatory function $\psi(x)$ via dilation (index i) and translation (index k), and $d_i(k)$ are wavelet coefficients that can be computed through filtering and sampling of $g(x)$ (Z is the set of positive natural numbers) [70]. Chuang and Kuo [36] demonstrated how to use spline wavelets to represent shape in deformable models. They compared spline-wavelet shape descriptors with Fourier descriptors for character-shape matching and for contour extraction from medical images; in both applications, wavelet descriptors were superior. They argued that the Fourier and wavelet transforms both decompose an input signal into a set of basis functions; however, the wavelet transform is more general in that it can use arbitrary basis functions, in contrast to the sinusoids used for the Fourier transform. Moreover, the wavelet transform provides a local-to-global description via its spatial- and frequency-localization characteristics (translation and dilation, respectively). Mumford [68] summarized the superiority of wavelet descriptors by saying that they allow for simultaneous expressive and compact representations of contours. One limitation of wavelets, however, is that they are not as intuitive as coordinates or turning angles.

4.2 Distance Measures

To compare contours, we must be able to measure the degree to which contours are similar. Although measuring contour similarity on the basis of size, position, or orientation is a well- defined task, measuring contour similarity on the basis of shape is an open problem. From a computational point of view, it is useful to capture our notion of shape similarity in a numeric figure that designates the closeness of shapes in some geometric space. We expect identical shapes to have a distance of zero, and different shapes to have a positive distance value, where the distance increases with an increase in shape difference as judged by our intuition [71]. In formal notation, a distance metric is a mapping $d{:}X \times X \to R$ if it satisfies the following conditions:

$$\begin{array}{ll} d(x, y) \geq 0 & \forall x, y \in X \\ d(x, y) = d(y, x) & \\ (x = y) \Rightarrow d(x, y) = 0 & \forall x, y \in X \\ d(x, y) + d(y, z) \geq d(x, z) & \forall x, y, z \in X \end{array}$$

The final equation—the triangle inequality—is necessary to ensure that the distance metric, when it is examined from a qualitative point of view, maintains transitivity for sets of shapes. Without this condition, it is possible to have cases where, for three shapes A, B, C, $d(A, B)$ and $d(B, C)$ are small, but $d(A, C)$ is large [72, 73]. Support for transitivity allows for efficient application of the distance metric for sets of contours, since we can infer similarity for the set by computing similarity for only sequential pairs.

Zikan [71] and Mumford [68] discuss a variety of metrics for measuring the distance between two sets of points A and B. Here are several examples:

1. The total **absolute Euclidean difference** between corresponding points:

$$D(A, B) = \sum_{i=0}^{k} |A_i - B_i|$$

2. The **minimum sum of Euclidean differences** between corresponding points when one contour is circularly shifted relative to the other contour:

$$D(A, B) = \min_{s \in [-r,r]} \sum_{i=0}^{k} |A_i - B_{\mathrm{mod}\,(i+s,\,k)}|$$

3. The **cardinality of the symmetric difference** between the point sets:

$$D(A, B) = |\langle A - B \rangle + \langle B - A \rangle|$$

4. The **Hausdorff distance** [74, 75]:

$$H(A, B) = \max(h(A, B), h(B, A))$$

   where $h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$, and $\| . \|$ is a norm on the points of A and B

5. The **transportation metric**: the minimum cost associated with transporting weighted hypothetical entities from points in one set to points in the other set [71]

A limitation of the Euclidean distance is that the latter requires a predefined point correspondence between the two point sets. The Hausdorff distance is sensitive to outliers; the cardinality of the symmetric difference metric is insensitive to outliers. The transportation metric allows for equal weighting of all points in a shape, and thus overcomes the limitations of the Hausdorff distance and the distance measured by the cardinality of the symmetric difference [68]. Mumford pointed out that several psychology studies demonstrated that the metric properties do not capture human intuition about similarity. This demonstration suggests that using metrics for automated recognition of similar shapes might result in program decisions that do not correspond well to human perception of similarity.

One useful measure that is not a metric is the distance computed by **nonlinear elastic matching**, which is the minimum sum of the Euclidean difference between corresponding points when the contours are nonlinearly warped with respect to one another. This measure is attractive for my application because the warping allows for recognition of shape similarity in the presence of nonlinear contour distortions due to imaging artifacts. This measure is not a metric because it does not satisfy the triangle inequality; however, Fagin and Stockmeyer show that, with a slight modification to the measure, a semitriangle inequality holds [73]. The **semitriangle inequality** is strongly related to the triangle inequality; it preserves similar intuitions with respect to the transitivity of qualitative shape differences.

4.3 Summary

In this chapter, I discussed methods for comparison of contours that I considered for IVSeg. I looked at methods that represent contour information and that calculate the difference between two representations. For my work, I chose to represent contours with sequences of turning angles because this representation is intuitive and had been demonstrated to be superior to several of the other contour-representation schemes for comparing contours with local distortions [65]. Similarly, I chose to measure the similarity of the sequences of turning angles with nonlinear elastic matching because that method accommodated for nonlinear distortions of shape due to imaging artifacts.

In Chapters 2 through 4, I have laid out the background for my work, and also explained some of the considerations for the choice of method elements within IVSeg. In Chapters 5 and 6, I describe the theoretical analyses and application considerations for the methods that I incorporated into IVSeg.

Chapter 5

Segmentation of 2D Sections Using Isolabel-Contour Maps

IVSeg segments medical image volumes by combining a 2D procedure, which identifies salient regions in each section, with a 3D procedure, which elicits a selection of an object from the user, and then groups regions that belong to the selected object from the entire sequence (Figure 1.7). Although it is possible both to identify and to select objects in 3D, the current version of IVSeg breaks down the segmentation task into a 2D part and a 3D part to simplify computation. I discuss the 2D component of IVSeg in this chapter, and the 3D component in Chapter 6.

The goal of the procedure for identifying salient regions is to portray objects of interest as coherent and smooth regions that a user can manipulate easily by clicking with a mouse. I observed that two characteristics of CTA data are likely to challenge existing segmentation methods. The first characteristic relates to the relative position of the vasculature and bone structures. Vessels and bones, both of which appear as bright structures in CTA sections, are sometimes so close to each other that, in some of the sections, they cannot be resolved by observation of intensity discontinuity alone. The second characteristic relates to the profound variation in the form of pathological structures across patients. This variation makes it difficult to use a priori information drawn from training data. Thus, I determined that a segmentation procedure for our application must have the ability to resolve complex objects with similar intensities that are close, or that touch, throughout the volume, without depending on extensive prior information. Similar requirements hold for segmentation tasks in other imaging modalities. The design of IVSeg's method for identifying salient regions addresses these challenges directly.

To identify salient regions in an image, IVSeg exploits the PVE that is inherent in image acquisition (see Section 3.1.2.4) in the following way. First, IVSeg removes noise from raw images with an anisotropic diffusion filter that preserves edges [76]. Then, IVSeg generates isolabel-contour maps by doing multilevel thresholding of the image intensity range with a fine partition. Finally, IVSeg extracts isolabel contours from the maps and compares their shapes to detect object boundaries (Figure 5.1). IVSeg builds on existing algorithms that use intensity thresholding: multilevel thresholding and connectivity. However, IVSeg addresses limitations associated with intensity thresholding by complementing intensity information with shape information that the program derives from the processed images.

This chapter describes IVSeg's method for identifying salient regions. Section 5.1 presents the contour-analysis theory that underlies the method. This theory pertains to the formation of isolabel-contour maps and their analysis for detection of object boundaries. Section 5.2 describes the algorithms that I incorporated into IVSeg; they are based on the contour-analysis theory. Section 5.3 analyzes of the performance of IVSeg's method for identifying salient regions, and describes two experiments to measure the method's robustness when program parameters change.

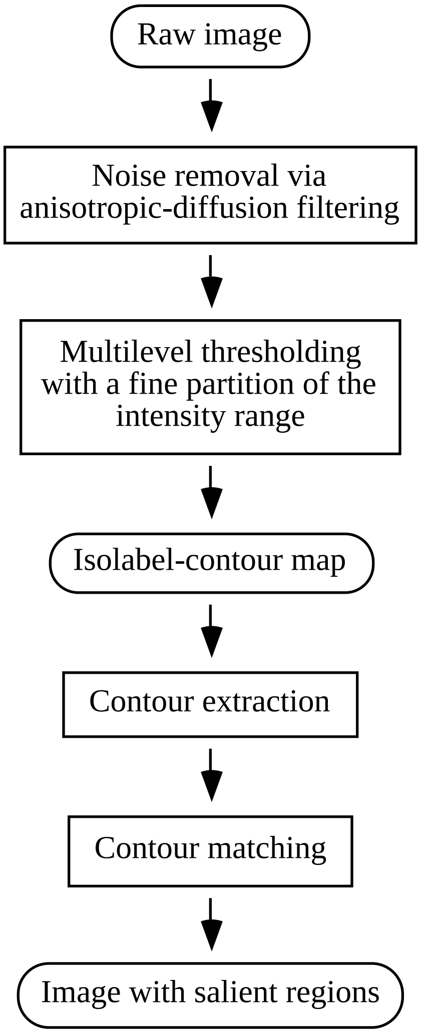

Raw image
Noise removal via anisotropic-diffusion filtering
Multilevel thresholding with a fine partition of the intensity range
Isolabel-contour map
Contour extraction
Contour matching
Image with salient regions

Figure 5.1 Steps that lead to the identification of salient regions in a 2D section. Noise removal is a preprocessing step. IVSeg matches the shape of contours to analyze isolabel-contour maps that it generates via multilevel thresholding.

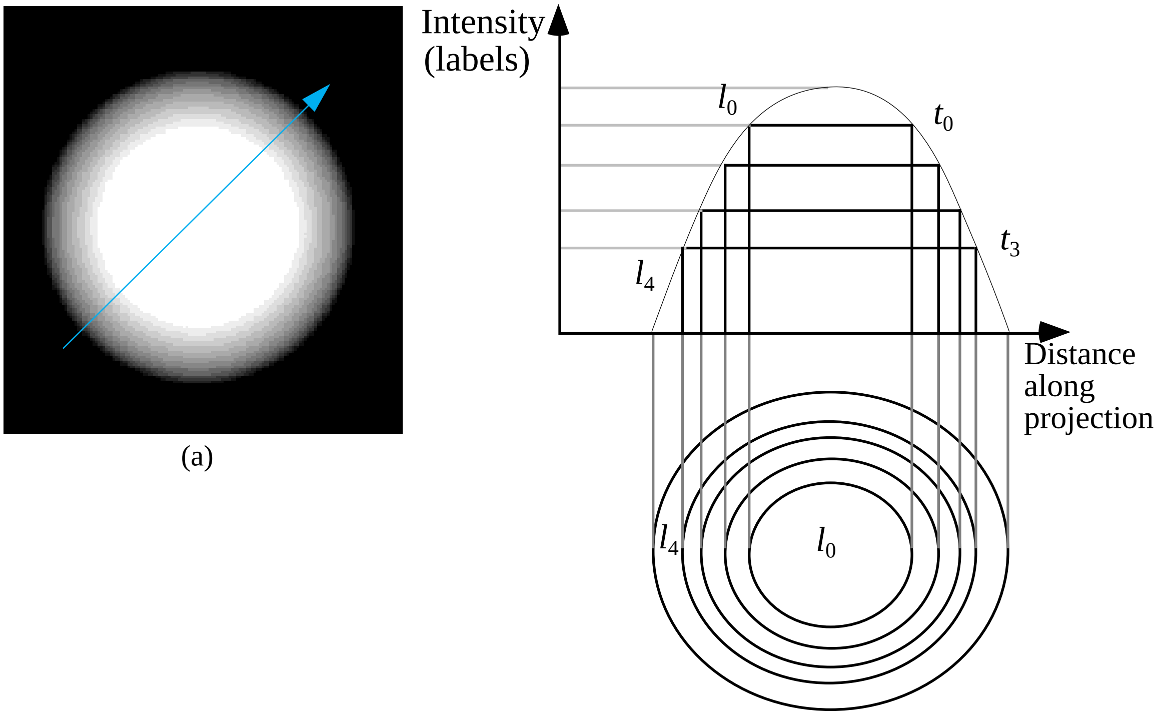
5.1 Theory

In this section, I explain how thresholding the intensity range of an image with a fine partition results in an isolabel-contour map. Then, I explore characteristics of objects' appearance in isolabel-contour maps, and I suggest that we can compare the shape of adjacent isolabel contours to detect object boundaries.

5.1.1 Formation of Isolabel-Contour Maps

Multilevel thresholding partitions the intensity range into bins, and assigns a distinct label to voxels associated with each intensity bin. The resulting image includes sets of connected voxels, such that the voxels in each set have a uniform label, and each voxel belongs to exactly one set. The voxels that lie on the boundary of such a set form an **isolabel contour**. In formal notation, the partition of the intensity range I results in a set of ordered distinct intensity subranges $s_j = [i_j, k_j)$, where $\bigcup s_j = I$. We assign, numeric labels l_j to voxels whose intensity is in the range s_j, where l_j reflects the relative intensity of s_j .

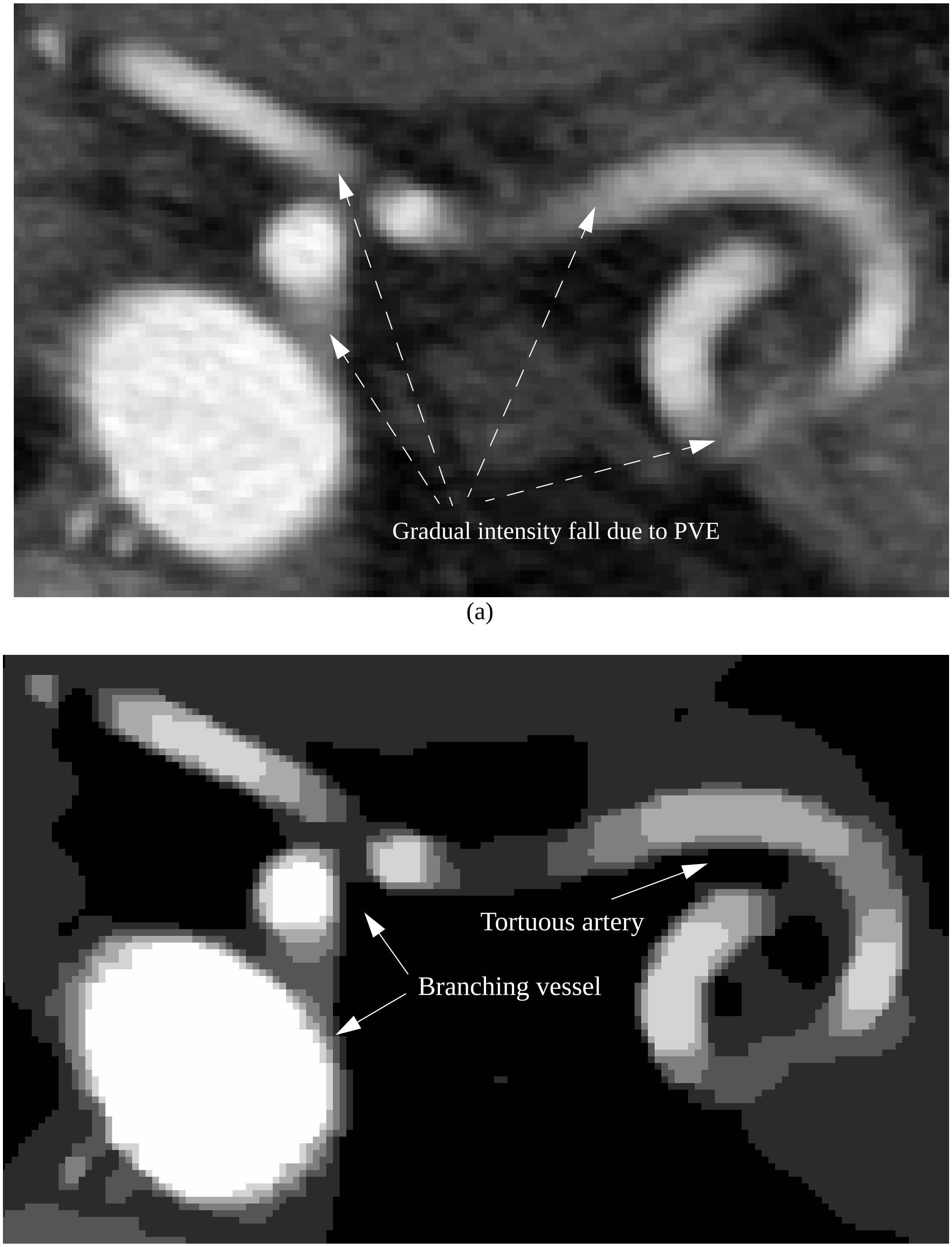
The proposed method differs from conventional thresholding in that it employs multilevel thresholding, not to segment structures based on intensity differences, but rather to generate pseudoconcentric contours near the boundaries of objects. Because partial-volume effects yield a gradual intensity fall across the boundaries of objects, multilevel thresholding with an intensity partition that is sufficiently fine results in labeled images such that the boundaries between regions with uniform labels form conspicuous contour patterns. Although the labeled images do not delineate isolabel contours explicitly, the resemblance of the contour patterns to isoelevation contours on topographical maps makes it reasonable to refer to the labeled images as *isolabel-contour maps* (Figures 5.2 to 5.4). If we scan an area within an isolabel-contour map from an object's center toward it's boundary, we see a distinct label pattern. Where the intensity gradient is monotonic in the raw image, the pattern of labels is monotonic as well. We observe dense contour patterns in areas of abrupt intensity gradients, and widespread contour patterns in areas of gradual intensity gradients.



(a)

(b)

Figure 5.2 A label pattern formed by thresholding of the intensity range with a fine partition. (a) A synthetic image of a circular object with a gradual intensity fall across the boundary and a line across the object. (b) The intensity profile of voxels along the line in part a, and the projections of voxels whose intensities are greater than or equal to the threshold intensities. Here, the l_i denote labels (l_0 corresponds to the highest range of intensities) and the t_i denote thresholds. Note the pattern of circular regions with decreasing labels. As the label index increases, the intensity range associated with the label decreases, and the distance of voxels from the center of the object increases.



(a)

(b)

Figure 5.3 A 3 × - magnified CTA section and its isolabel-contour map. (a) The raw section. Note the gradual intensity fall due to extensive PVE where the vessel is

oriented at a small angle with respect to the image plane. (b) The isolabel-contour map. The map resulted from a seven-threshold partition of the intensity range. Note the saddle-like patterns of isolabel contours for the branching and tortuous vessels.

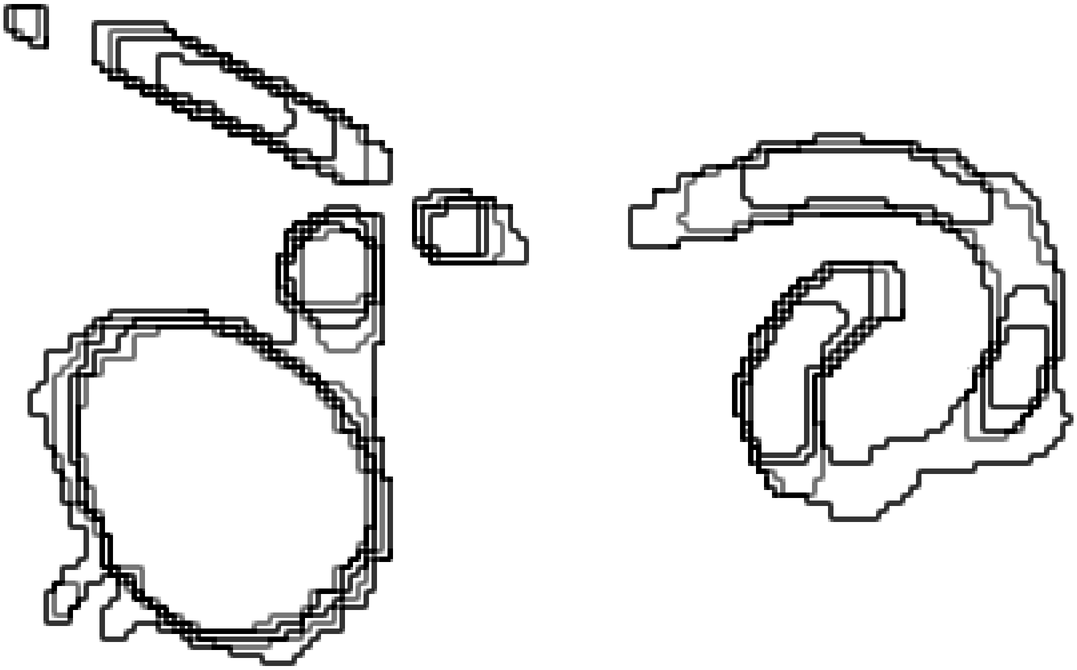


Figure 5.4 Contour patterns for the major structures in Figure 5.3. For clarity, the figure shows only the contours that correspond to the six highest thresholds, and displays the contours in a variety of grayscale values. Note that the contour patterns are dense along sharp intensity gradients and are more widespread where intensity fall is gradual.

5.1.2 Characteristics of Isolabel Contours

The shape of an isolabel contour that encloses a set of voxels depends on the shape of the object of which the voxels are an image, on noise, on intensity variations within the object image, and on manifestations of partial volume. I observed that, in general, isolabel contours that enclose voxels from only a single object have shapes that are similar to the shape of the object boundary, and thus, that these contours have similar shapes; similarly, isolabel contours that enclose voxels from different objects often have different shapes.

Isolabel contours that enclose voxels from the same object often also have similar orientation, but their size and position (center of mass) may vary, depending on the rate and uniformity of the intensity fall across the boundary of the object at various boundary points. These observations suggested that an automated procedure could search for similarity of shape in adjacent contours to determine the approximate location of an object's boundary.

Figures 5.5 and 5.6 show isolabel contours that resulted from multilevel thresholding of simulation data and patient data, respectively. Figure 5.5 shows that isolabel contours in the noiseless simulation data are fairly regular in that each contour adds a similar radial increment over the previous contour, and only one contour is a **terminal contour**—a contour that does not enclose any other contour. Figure 5.5 also demonstrates the geometric distortion associated with discrete representation of small circle-shaped objects: The isolabel contours of the 6 mm-diameter object are jagged. The isolabel contours in Figure 5.6, which are from patient data, are less regular than the isolabel contours in Figure 5.5 in that the former contours add variable radial increments—at various points on the contour—to the contours that they enclose. The irregularity of the contours for the patient data arise because the attenuation of anatomical objects is less homogeneous than the attenuation of the synthetic object, and because of noise. The preceding examples point out that irregularities and distortions associated with contours that enclose voxels of a single object might lead to a significant difference in the shapes of these contours; see, for example, the considerable difference in the shapes of contours 1 and 2 in Figure 5.6.

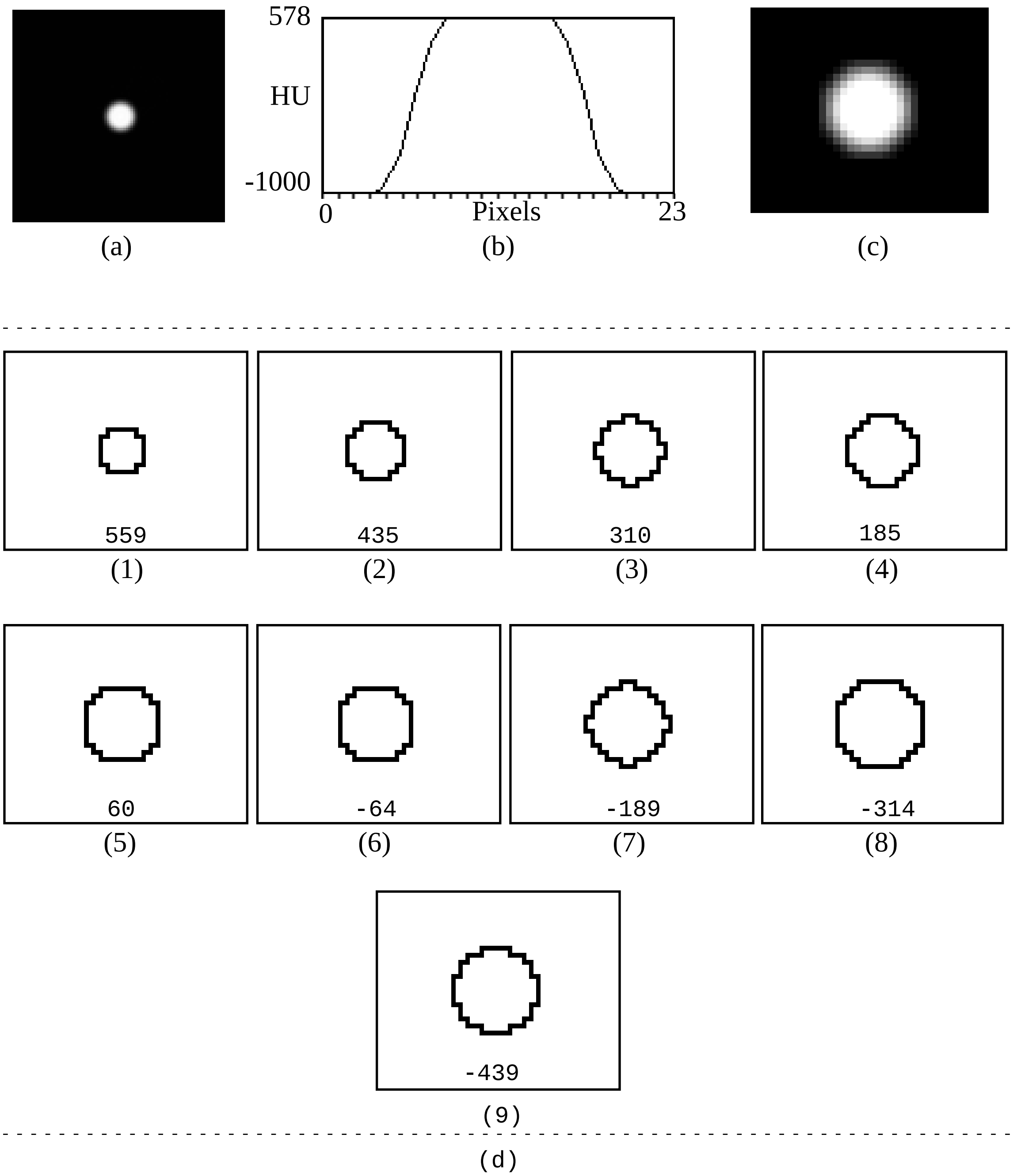


Figure 5.5 Isolabel contours in a noiseless CT simulation. (a) A section through a cylinder with a diameter of 6 mm that is oriented perpendicular to the imaging plane. (b) The intensity profile along a line that crosses the center of the cylinder section. Note that the pixel size in the simulation is 0.52 mm, and that there are 16 nonzero pixels along the crossing line, instead of the expected 12 pixels. The image

of the cylinder appears extended due to partial-volume effect. (c) The cylinder section after magnification by a factor of 3. (d) (1–9) Isolabel contours that result from thresholding the cylinder sections with 13 thresholds as labeled. The contours are ordered from the innermost contour to the outermost. Contour 1 is the terminal contour; because it does not enclose any other contours. The intensity of the voxels that are enclosed between each pair of contours is within the range of intensities specified by the corresponding thresholds.

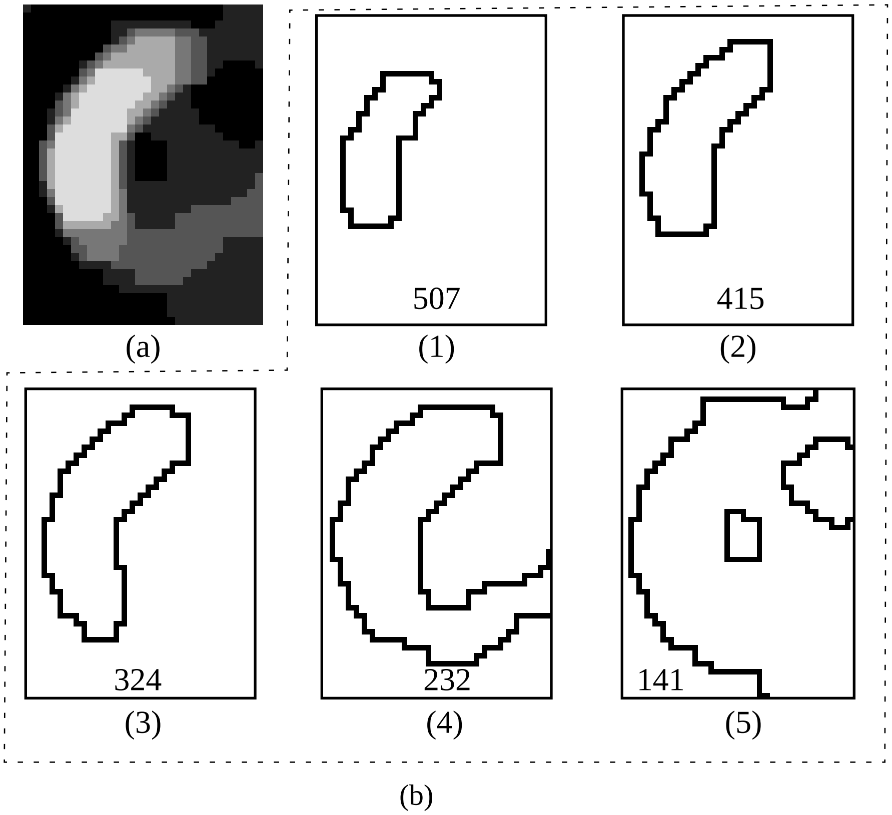


Figure 5.6 Isolabel contours from the CTA section in Figure 5.3. (a) An excerpt from the rightmost third of the isolabel-contour map. (b)(1–5) Several contours along the perimeter of part of the tortuous vessel. Note the shape similarity of contours 1 through 4, and the shape difference between contours 4 and 5.

A criterion for boundary detection that is based on only similarity in the shape of isolabel contours might result in false decisions. From preliminary tests, I learned that this criterion might result in false decisions where the program would delineate small insignificant regions as independent objects with independent boundaries; for example, small regions that emerge in CTA data due to correlated noise might have isolabel contours with similar shape. The challenge in applying contour matching to detection of object boundaries was to achieve good accuracy despite this potential for false decisions.

Another challenge is related to the ambiguity of saddle-like patterns, which resulted from partial volume effects. Saddle-like contour patterns may occur in tortuous structures whose plane of curvature is perpendicular to the image plane or in vessels that branch in the through-plane direction. Figure 5.3 shows examples of both. Saddle-like patterns may also occur in structures that are close to each other and have overlapping intensity distributions. Figure 5.7 shows an example for simulation data, where the contour encompasses voxels of two nearby objects. When a saddle-like pattern includes voxels of a single object, we would like to include all the voxels that are part of the pattern; when a saddle-like pattern includes voxels of different objects, we would like to distinguish between sets of voxels that belong to the saddle peaks. The method described in Chapter 6 allows us to make this distinction.

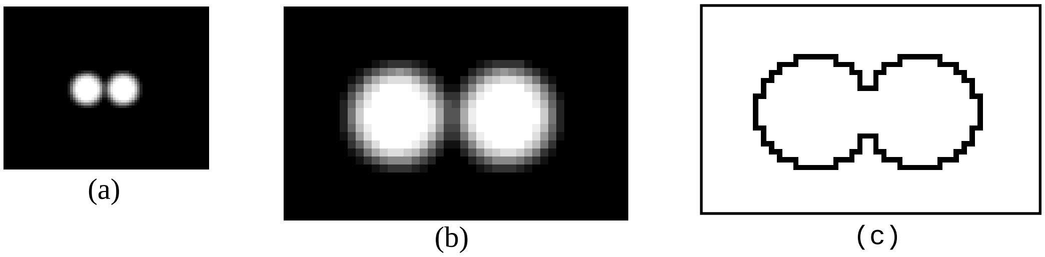


Figure 5.7 A simulation of a section through two cylinders of 6 mm diameter that are 1 mm apart. (a) The raw section. (b) The isolabel-contour map. (c) The isolabel contour that corresponds to an intensity threshold at -812 HU. The isolabel contour encompasses voxels of both cylinders.

We can describe the entire pattern of contours in an isolabel-contour map as a relationship of enclosure. This description allows us to manipulate easily voxels that related contours enclose. For any two isolabel contours in an isolabel-contour map, c_i, c_j, with sets of connected voxels that they encompass r_i, r_j, the following statement holds:

$$\langle r_i \cap r_j = \varnothing \rangle \vee \langle r_i \cap r_j = r_i \rangle \vee \langle r_i \cap r_j = r_j \rangle \, .$$

The statement is true because its negation implies that there are voxels with more than one label assignment, and that implication is inconsistent with generation of the isolabel-contour map via multilevel thresholding. We define an **enclosure** relation R_e over the image. If r_{i+1} is the set of voxels encompassed by contour c_{i+1}, and r_i is the set of voxels encompassed by contour c_i, then $(c_{i+1}, c_i) \in R_e$, or c_{i+1} encloses c_i, if and only if $r_{i+1} \supseteq r_i$.The enclosure relation allows us to produce hierarchical descriptions of contours and of the regions that they bind. Figure 5.8 illustrates a labeled image and a corresponding region-based tree structure.

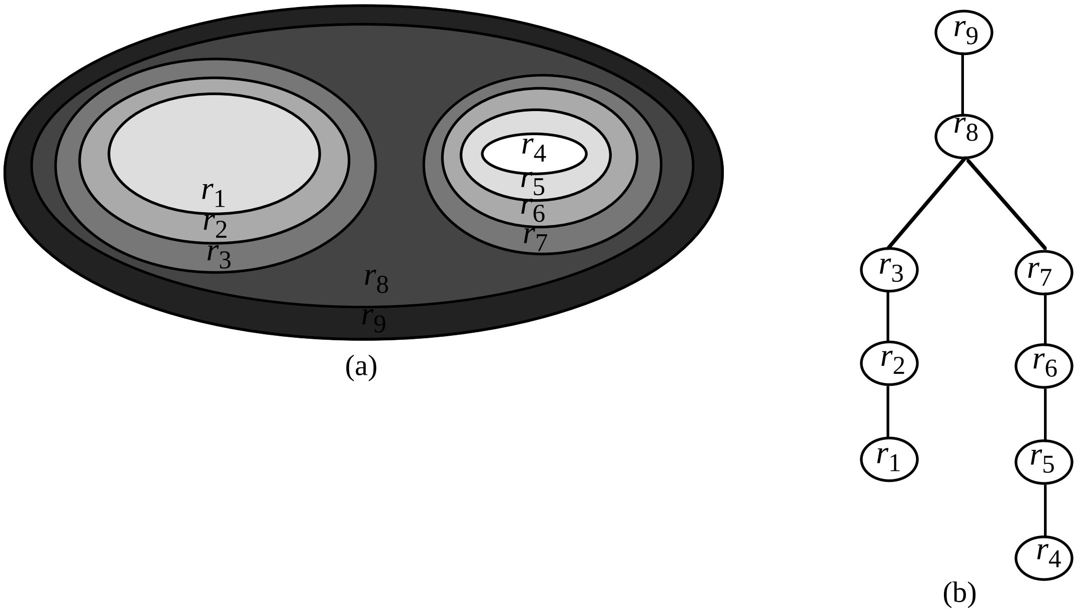


Figure 5.8 A hierarchical description of regions and their corresponding contours in an image. (a) A hypothetical labeled image. (b) The corresponding region tree. There are two objects in the hypothetical raw image: one is centered at region r_1;

the other is centered at region r_4. The two objects are represented by two separate branches in the tree structure.

5.1.3 Detection of Object Boundaries via Matching of Isolabel Contours

The characteristics of isolabel contours described in Section 5.1.2 suggest that we can analyze isolabel-contour maps to determine the approximate boundary of an object. This analysis is based on matches of **consecutive isolabel contours**, or two contours that are boundaries to the same set of connected voxels with a uniform label, where one of the contours encloses all voxels that the other encloses. Figure 5.9 shows an example of consecutive isolabel contours. Note that consecutive isolabel contours may consist of disjoint point sets, or may share a subset of points (Figure 5.9).

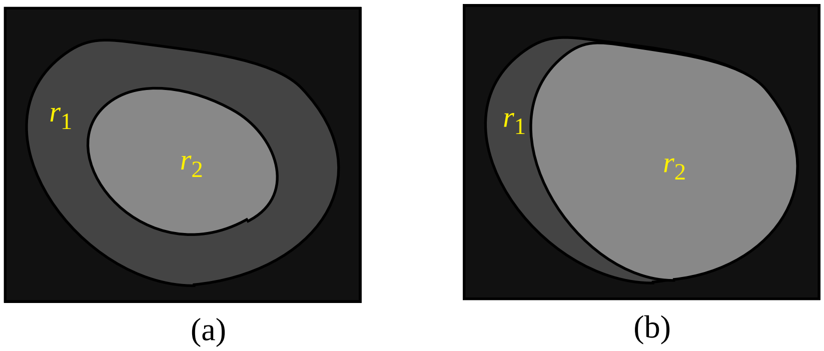


Figure 5.9 Consecutive isolabel contours. (a) The contours have disjoint point sets. (b) The contours share a subset of points. For both a and b, the two contours are boundaries to the set of connected voxels with a uniform label marked by r_1. Both contours enclose the set of connected voxels marked by r_2.

Given an ordered set of consecutive contours $C = c_1, \ldots, c_{n-1}, c_n$, and a distance measure $D(c_i, c_j)$ that assigns scores to pairs of consecutive contours c_i, c_j according to how well the contours match, C is a pattern of contours with similar shape if and only if, for every adjacent pair of contours $c_{i-1}, c_i \in C$, $D(c_{i-1}, c_i) < T_m$, where T_m is a predefined distance value. Based on the observation that large shape changes in sequences of consecutive contours are typically associated with a transition from within an object into the background, we search for cues in the form of a deviation from a pattern of contours with similar shape; these cues suggest the presence of an object boundary in the vicinity of where the deviation occurred.

I have presented the theoretical underpinning of IVSeg's method for identifying salient regions. Section 5.2 describes the algorithm that identifies approximate object boundaries by extracting and matching isolabel contours.

5.2 Application

In this section, I describe how IVSeg determines which contours to process. I then explain the details of processing the contours: extraction, representation, and matching.

5.2.1 Selection of Isolabel Contours

IVSeg uses the set of labels in the isolabel-contour map to guide the processing of consecutive contours. The program considers labels in monotonic order of associated intensity. Because isolabel contours do not cross (crossing isolabel contours would imply voxels with multiple labels at the cross points), this order guarantees that the program considers contours in a fixed direction with respect to the object center, and, consequently, that the program matches contours that are close to the object boundary. For example, in CTA sections, where the target objects are bright, as label values that are associated with contours decrease, the distance of contour points from the center of target objects increases (see Figure 5.2). Therefore, in considering labels in decreasing order, IVSeg matches consecutive contours from the innermost contour to the outermost contour, and thus it is guaranteed to cross the boundaries of target objects.

We can view the order in which IVSeg considers isolabel contours as a bottom-up, breadth-first traversal of the hierarchical representation of the contours from the contour map (Figure 5.8). IVSeg maintains a separate data structure for each path of consecutive contours within the representation tree. Each data structure includes the list of coordinates for the most recent contour that IVSeg considered for that path. The following procedure illustrates the course in which IVSeg considers contours for an arbitrary iteration j:

1. Given a current label l_j, find all regions with that label that have not been visited. The regions are set as roots of new subtrees. Establish a new path structure for the new subtrees.
2. For each existing path, find the contour that is associated with label l_{j+1}, and compare the contour to the preceding contour for that path.

The preceding procedure is adequate both for objects with intensity profiles that have a single peak and for objects with intensity profiles that have multiple peaks. For example, consider the object in Figure 5.10. The intensity profile for the object has more than one peak due to noise. IVSeg maintains a single path at l_0 andl_1; then, it establishes a new path at l_2. The program maintains two paths at l_3 and l_4; it merges the two paths into one parent path at l_5. I explain in Section 5.2.5 how IVSeg tracks the hierarchical relationship between regions.

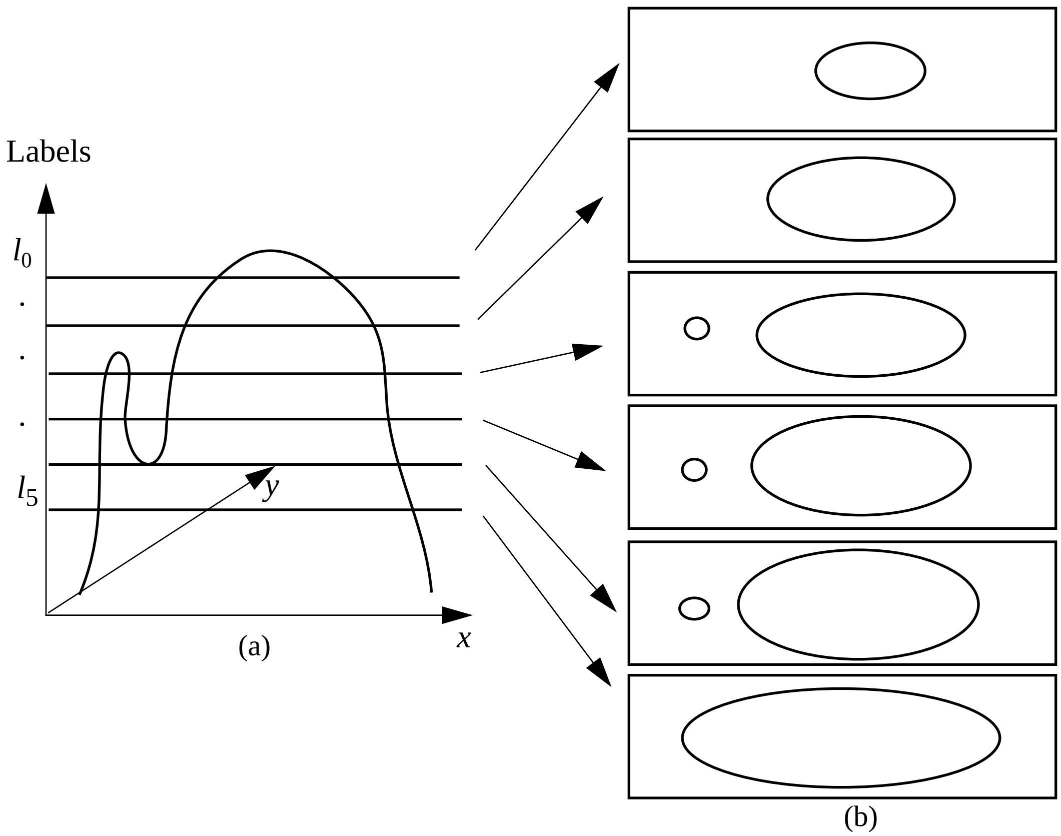


Figure 5.10 Contours that IVSeg extracts for an object with a nonmonotonic intensity profile. (a) The intensity profile along a section that is parallel to the *x*-intensity plane, and five intensity thresholds. (b) Contours that IVSeg extracts.

To compare the shape of contours, IVSeg extracts the contours, represents the contours in a way that incorporates shape information, and computes a measure of the distance between the contour representations. Sections 5.2.2 through 5.2.4 describe the details of the preceding subtasks.

5.2.2 Extraction of Isolabel Contours

During each iteration in the course of matching consecutive isolabel contours, IVSeg matches the current contour, which represents the current state of the object that the program reached in the previous iteration, to the next contour, which IVSeg extracts from the contour map. The program extracts the next contour by growing connected voxels with the specified label from seed points adjacent to the most recent isolabel contour from the outside (Figure 5.11). The voxels on the boundary of the enlarged region define the next isolabel contour. IVSeg extracts the voxels on the boundary by finding an initial voxel with the specified label that has an eight-neighbor[1] with a different label, and then tracking eight-neighbor–connected voxels with the specified label that have an eight-neighbor with a different label.

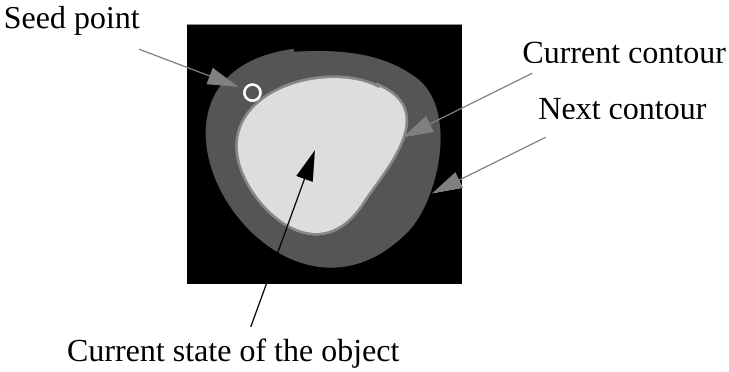


Figure 5.11 Results of applying connectivity to identify the next contour given the current contour.

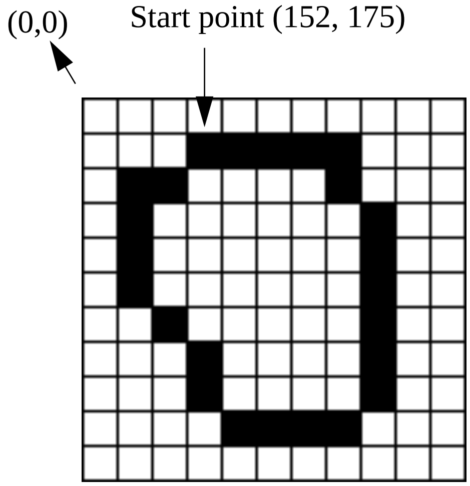
5.2.3 Representation of Contours

IVSeg decomposes contours into functions in parametric space with parameter t, $x(t), y(t)$; then, to allow for smoothing, it fits a cubic spline to a sample of equispaced domain points. It samples each spline, and creates a sequence of turning angles by recording the orientations of the line connecting each pair of adjacent sample spline coordinates [72]. Smoothing the isolabel contours via spline fitting reduces the effects of noise and geometric distortions (for small objects) on the results of contour matching. However, the degree of smoothing must be such that it does not eliminate object shape characteristics. In segmentation of CTA data, for a contour of n points, IVSeg samples $\lceil r \cdot n \rceil$ $0 < r < 1$ control points from the contour, and fits a spline to the resulting control-point sequence.

[1] Eight-neighbor voxels are oriented with respect to each other like the eight major directions on a compass.

IVSeg then samples $n + 1$ points from the continuous spline to create a sequence of turning angles, where each turning angle represents the slope of the line connecting two sequential spline points.

To reduce the sensitivity of contour matching to the selection of control points, IVSeg computes $\frac{1}{r}$ splines, each with a set of control points whose indices are offset by successive integers relative to the original set of control points. Then, the program averages the resulting coordinates for spline points that correspond in the original contour's parameter space—points that have the same *t* value. Figures 5.12 through 5.15 show an example of adjacent contours and their representation. In the current implementation of IVSeg, $r = 0.25$.



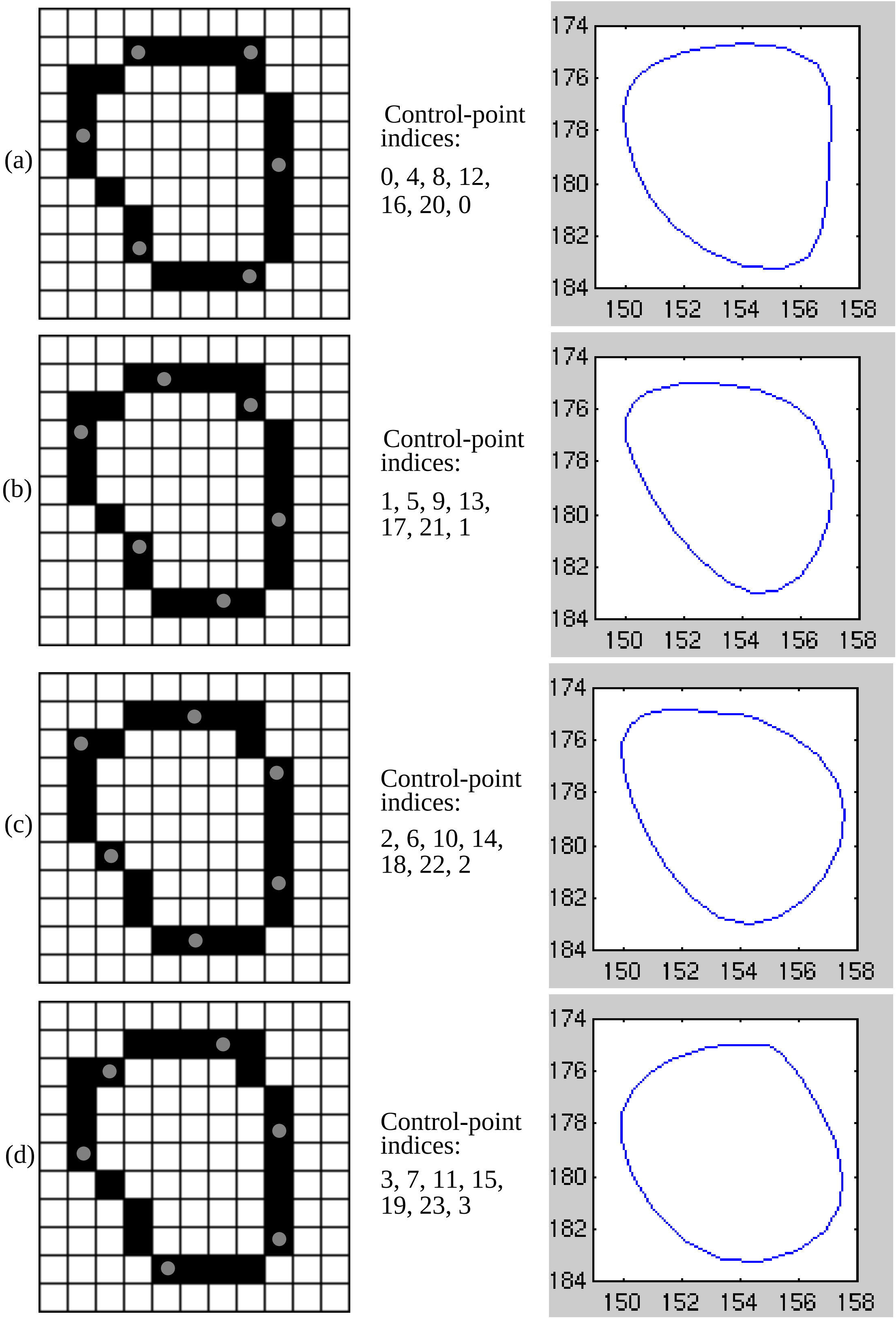
Contour points

t is an integer in the range $0 \le t \le 23$

$x(t)$ = {152, 153, 154, 155, 156, 156, 157, 157,
157, 157, 157, 157, 156, 155, 154, 153,
152, 152, 151, 150, 150, 150, 150, 151}

$y(t)$ = {175, 175, 175, 175, 175, 176, 177, 178,
179, 180, 181, 182, 183, 183, 183, 183,
182, 181, 180, 179, 178, 177, 176, 176}

Figure 5.12 An extracted isolabel contour (contour A) and its coordinates in a matrix coordinate system with the origin at the upper-left corner.



(a) Control-point indices:

0, 4, 8, 12, 16, 20, 0

(b) Control-point indices:

1, 5, 9, 13, 17, 21, 1

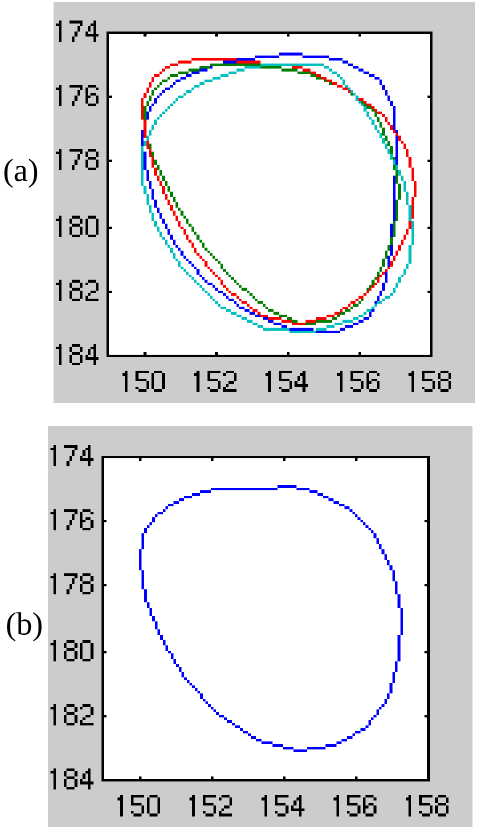
(c) Control-point indices:

2, 6, 10, 14, 18, 22, 2

(d) Control-point indices:

3, 7, 11, 15, 19, 23, 3

Figure 5.13 Splines fitted to the contour in Figure 5.12. (a) From left to right: control points, their indices, and a plot of sample spline points for offset 0. (b–d), same as a but for control-point offsets of 1 through 3, respectively.

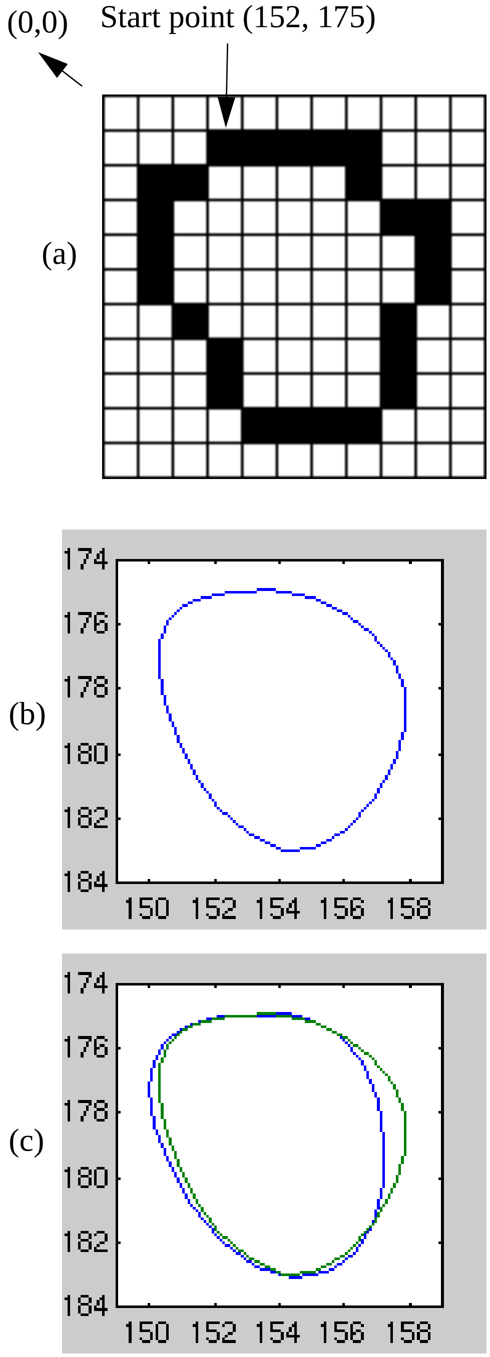


Average of sample spline points
t is an integer in the range $0 \le t \le 24$

$x(t)$ = {153.18, 153.60, 154.17, 154.92, 155.79,
156.53, 157.01, 157.23, 157.20, 156.90,
156.32, 155.44, 154.38, 153.25, 152.20,
151.30, 150.60, 150.16, 150.00, 150.12,
150.48, 150.99, 151.59, 152.21, 152.78}

$y(t)$ = {174.99, 174.95, 174.93, 175.07, 175.55,
176.43, 177.60, 178.91, 180.21, 181.40,
182.36, 182.95, 183.10, 182.76, 181.97,
180.86, 179.59, 178.34, 177.26, 176.41,
175.79, 175.39, 175.15, 175.02, 174.97}

Figure 5.14 Plots of four splines and plot of average spline. (a) All four splines superimposed. (b) The average spline and its coordinates (the coordinates are rounded to one-hundredth of a unit).



Contour points

t is an integer in the range $0 \le t \le 24$

$x(t)$ = {152, 153, 154, 155, 156, 156, 157, 158, 158, 158, 157, 157, 157, 156, 155, 154, 153, 152, 152, 151, 150, 150, 150, 150, 151}

$y(t)$ = {175, 175, 175, 175, 175, 176, 177, 177, 178, 179, 180, 181, 182, 183, 183, 183, 183, 182, 181, 180, 179, 178, 177, 176, 176}

Average of sample spline points

t is an integer in the range $0 \le t \le 25$

$x(t)$ = {153.11, 153.60, 154.20, 154.95, 155.87, 156.80, 157.50, 157.86, 157.85, 157.50, 156.88, 156.06, 155.10, 154.08, 153.06, 152.13, 151.38, 150.85, 150.51, 150.35, 150.35, 150.55, 150.93, 151.44, 152.02, 152.64}

$y(t)$ = {174.97, 174.95, 174.96, 175.12, 175.55, 176.24, 177.10, 178.09, 179.17, 180.32, 181.42, 182.35, 182.91, 182.98, 182.53, 181.70, 180.64, 179.50, 178.38, 177.37, 176.53, 175.90, 175.47, 175.20, 175.05 174.97}

Figure 5.15 The isolabel contour that follows the contour in Figure 5.13 (contour B). (a) The contour and its coordinates. (b) A plot of the average spline and its coordinates. (c) Superimposed plots of the average splines of the two isolabel contours.

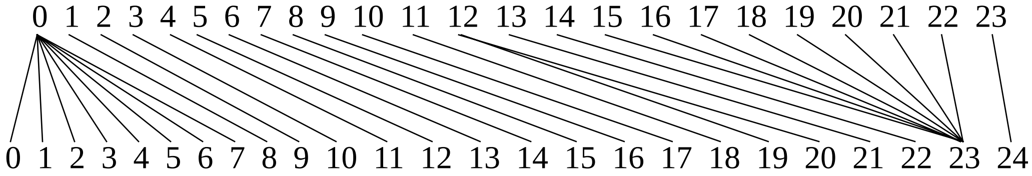
5.2.4 Distance Measure for Assignment of a Similarity Score

IVSeg assigns a similarity score to each pair of adjacent contours by performing a nonlinear elastic match between the respective sequences of turning angles [73]. A mapping between two sequences assigns all elements in one of the sequences to an element in the other sequence. The matching score is the minimum average absolute difference between corresponding sequence elements over all possible element mappings. The matching algorithm uses dynamic programming to find a mapping with a minimal score between the two sequences efficiently [77]. Figure 5.16 shows the turning-angle sequences for contours A and B and the mapping that produces the similarity score (0.04 radians).

t is an integer in the range $0 \le t \le 23$
$a(t)$ ={6.20, 6.25, 0.18, 0.50, 0.88, 1.18, 1.41, 1.59, 1.82, 2.12, 2.54, 3.01,
3.44, 3.78, 4.03, 4.21, 4.37, 4.57, 4.85, 5.24, 5.61, 5.90, 6.09, 6.19}

t is an integer in the range $0 \le t \le 24$
$b(t)$ ={6.24, 0.02, 0.21, 0.43, 0.64, 0.89, 1.22, 1.58, 1.87, 2.08, 2.30, 2.61,
3.08, 3.55, 3.87, 4.10, 4.28, 4.42, 4.55, 5.72, 5.02, 5.44, 5.80, 6.03, 6.15}

(a)



(b)

Figure 5.16 Turning-angle sequences and their optimal mapping. (a) Angle sequences for contour A ($a(t)$) and that for contour B ($b(t)$), in radians. (b) A mapping that produces the minimal matching score.

5.2.5 Contour Matching and Formation of Salient Regions

I now explain how IVSeg forms salient regions, which consist of connected voxels that have a nonzero grayscale code, via matching of isolabel contours. The similarity criterion that the contour-matching procedure uses allows us to detect isolabel contours that are likely to be good estimates for object boundaries. To maintain the location of the detected contours, IVSeg assigns a uniform grayscale code to voxels that have not been assigned a label previously and that are enclosed within patterns of similar contours. IVSeg updates the grayscale code to a new value when it encounters a contour that is not similar to the preceding contour. Figure 5.17 shows pseudocode for matching a pair of contours and for updating the grayscale code assigned to voxels that the contours enclose. The pseudocode assumes that the **current contour** and the **next contour** are two adjacent contours that IVSeg extracted.

```
let $c_i$ be the current contour
let $c_{i+1}$ be the next contour
let $D_i = D(c_i, c_j)$
let $l_c$ be an arbitrary grayscale code
if $(D_i > SDT)$ update $l_c$ to be a
    different grayscale code
assign $l_c$ to all voxels that the next contour encloses and that do not have a grayscale code
proceed to the next pair of extracted contours
```

Figure 5.17 Pseudocode for matching of isolabel contours. Here, $D\langle c_i, c_j\rangle$ is the distance between contours c_i, c_j, and SDT is a real-valued shape-difference threshold.

During the formation of salient regions, IVSeg tracks the hierarchical relationship between regions (see Section 5.1.2) in the following way. When IVSeg determines that a set of voxels belongs to a region a within a given path, the program checks the state of the voxels. If the voxels are marked as already belonging to a region b, the program assigns region b as a child of region a. If the voxels are not marked, the program tags the voxels to mark their inclusion within region a.

The hierarchical description of isolabel contours allows for efficient comparison of the shape of contours in an isolabel-contour map. To understand the point at which the hierarchical representation saves computation, we assume that there are NT thresholds, and therefore $NT - 1$ labels, and that each contour in the isolabel-contour map encloses directly[1] exactly q contours. Under this assumption there are q^{NT-2} terminal contours, which do not enclose any contours within. If we compare the shape of consecutive contours in an exhaustive manner, the total number of contour comparisons is

$$(NT - 2) \cdot q^{NT-2}. \tag{5.1}$$

If we exploit the hierarchical representation of the contours to avoid unnecessary comparisons, the number of comparisons is equal to the number of branches in the tree:

$$\sum_{i=1}^{NT-2} q^{i}. \tag{5.2}$$

The expressions in Equations 5.1 and 5.2 can both be viewed as sums of $NT - 2$ powers of q. All powers of q in Equation 5.2 are smaller than those in Equation 5.1, except for the final one, which is equal to the powers of q in Equation 5.1. Therefore,

$$\sum_{i=1}^{NT-2} q^{i} < (NT - 2) \cdot q^{NT-2}, \tag{5.3}$$

and we conclude that exploiting the contour hierarchy allows us to reduce the computation associated with comparison of consecutive isolabel contours. From Equation 5.3, we observe that we obtain the maximum reduction in computation,

$$(NT - 2) \cdot q^{NT-2} - \sum_{i=1}^{NT-2} q^{i}, \tag{5.4}$$

from a tree-structure representation when the branching factor is greater than 1 and is the same for every level in the tree. In practice, however, the branching factor varies through-

[1] Contour *a* directly encloses contour *b* if there is no other contour *c* that encloses *b* where *a* also encloses c.

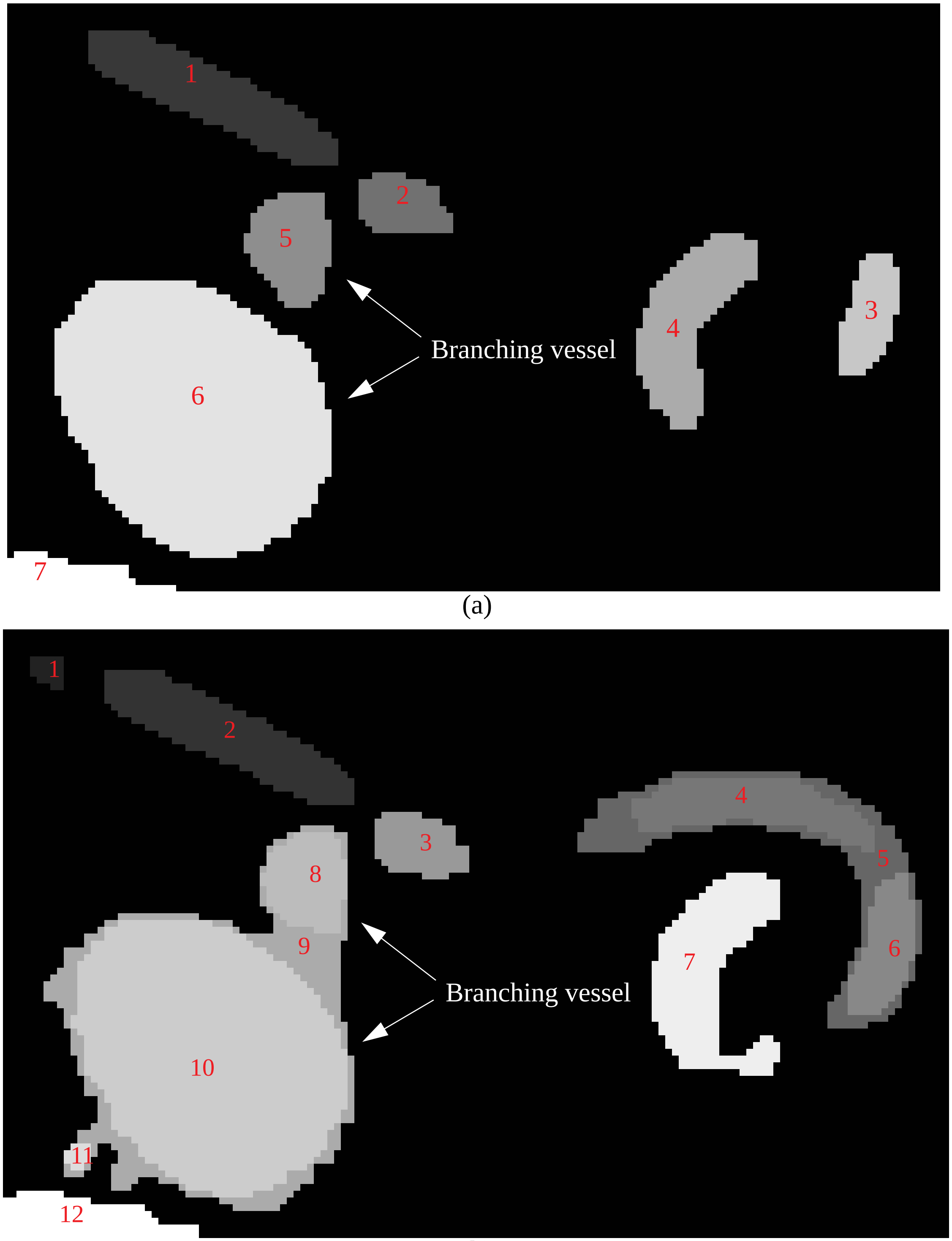
out the tree, and the difference between computation with and without the tree structure is smaller than the difference indicated by Equation 5.4.

For example, in Figure 5.8, the consecutive contours in the labeled image are (r_1, r_2) (r_2, r_3) (r_3, r_8) (r_4, r_5) (r_5, r_6) (r_6, r_7) (r_7, r_8) (r_8, r_9). With a straightforward comparison of contours, IVSeg processes the contour in the following order: (r_4, r_5) (r_5, r_6) (r_1, r_2) (r_6, r_7) (r_2, r_3) (r_7, r_8) (r_3, r_8) (r_8, r_9) (r_8, r_9)—a total of nine contour comparisons. The program processes contours r_8 and r_9 once for each branch stemming from r_8 in the region tree. However, with the hierarchical representation of contours, IVSeg tracks parent–child relations between regions, and terminates comparison of contours for child subtrees. Thus, the program processes each contour at most twice: once as the first contour in a pair, and once as the second contour. In this example, the contours are processed in the following order: (r_4, r_5) (r_5, r_6) (r_1, r_2) (r_6, r_7) (r_2, r_3) (r_7, r_8) (r_3, r_8) (r_8, r_9)—a total of eight contour comparisons, a savings of one-ninth of the contour comparisons. The preceding example is a simplification of the type of region hierarchies that we get in real CTA images, because there is only one branching point, and the number of branches is two.

To illustrate the computation savings, I ran IVSeg's procedure for identification of salient regions on a randomly sampled CTA case with 158 sections. I observed up to seven branches for a single branching point, and a total of 4662 branching points. Without the hierarchical representations, the program performed a total of 1003332 contour comparisons. With the hierarchical representation, the program performed 223182 contour comparisons. The hierarchical representation yielded a reduction in the number of contour comparisons by a factor of over four.

5.2.6 Interpretation of Images With Salient Regions

The contour-matching process results in an image with one or more salient regions (Figure 5.18). Each salient region is surrounded by a zero background (black in the figure) and consists of one or more sets of connected voxels with a different grayscale code. Multiple grayscale codes for voxels of a single structure result from multiple occurrences of contours with a significant shape difference within the structure. Note that the boundaries of salient regions in Figure 5.18 a correspond to contours in Figure 5.4. The differences between parts a and b in Figure 5.18 occur because, given the larger number of thresholds in b, more voxels are enclosed within contours of similar shape, and thus more voxels have nonzero values. Different grayscale codes for a single salient region—for example, in the branching structures—delineate substructures within salient regions, so that post-processes can make independent reference to individual substructures. To support postprocessing of salient regions, for example, see Section 6.2.1, IVSeg outputs a hierarchical description of the enclosure relation of sets of connected voxels with a uniform grayscale code. From a conceptual point of view, this description is similar to the description in Section 5.1.2.


(a)

(b)

Figure 5.18 Demonstration of salient regions. (a) Salient regions for seven thresholds. (b) Salient regions for 14 thresholds. The numbers indicate distinct grayscale codes.

Section 5.3 discusses the performance of IVSeg. I describe experiments that I ran to determine the sensitivity of IVSeg to parameter settings, and to compare the sensitivity of IVSeg and that of conventional thresholding.

5.3 Performance Analysis

Activation of IVSeg requires specification of values of four parameters: lower intensity (LI), upper intensity (UI), number of intensity thresholds (NT), and shape-difference threshold (SDT). Section 5.3.1 describes experiments that I conducted to test the sensitivity of IVSeg's results for the identification of salient regions to small changes in the parameters LI, UI, NT and SDT. Section describes a comparison of the sensitivity of IVSeg and conventional thresholding to the respective thresholds that the two methods use.

5.3.1 Sensitivity Analysis

I determined the sensitivity of the results of IVSeg's procedure for identification of salient regions to 5 percent changes in parameter settings. The parameters that I examined were LI, UI, NT and SDT. Through experimentation with four CTA cases, I determined those settings for the parameters that produced good results according to my subjective judgment: LI=49 HU, UI=598 HU, NT=170, SDT = 0.0475. My goal was to determine whether 5 percent changes in these parameters produced results that were significantly different from the results that I generated with the original parameter setting.

5.3.1.1 Study Design for Sensitivity Analysis

I selected a random set of 11 cases from a 1-year-old archive of abdominal–pelvic CTA cases. The number of sections in each case ranged from 152 to 198. I sampled three sections from each case, for a total of 33 sections. I activated IVSeg to identify salient regions in the collection of images for the original parameter settings; then I activated the program for 16 combinations of the four parameters at two levels each, where, for each parameter, each level constituted a 5-percent change, either above or below the original setting. I considered the results that IVSeg produced for the original parameter setting as the **baseline images**, and those for the 16 parameter-level combinations as the **treatment images.** Then, I compared each treatment image to the corresponding baseline image. I measured the degree of mismatch for every pair of corresponding images, and I performed an analysis of variance for a 4-way classification to determine whether any of the four parameters or parameter interactions explained differences in the mismatches between treatment images and baseline images. The null hypothesis was that *none of the parameters (LI, UI, NT, SDT), or any combination of these parameters, explained the differences in corresponding regions between the baseline images and the treatment images.*

5.3.1.2 Sensitivity Analysis Measures

I compared sets of images that IVSeg produced in the following way. Because IVSeg finds approximate boundaries for objects with an intensity range that the user specifies by indicating values for LI and UI, I compared the results for only objects in the prespecified intensity range. The sensitivity analysis concentrated on detection of images of arteries, whose intensity ranged from about 0 HU to 500 HU in CTA images. I activated a routine that used standard edits that experts generated, and that included only arteries, to select regions that IVSeg produced and that had the least amount of voxel mismatches with the expert delineated arteries. I generated masks from the selected regions, and compared the masks that resulted for images of each of the 16 treatment sets to the masks of images of the baseline set. For each region in one of the masks, I recorded whether it had a overlapping region in the corresponding mask. If there was a single overlapping region, I computed the **normalized voxel mismatch (NVM)**, or the total number of voxel mismatches divided by the number of voxels in the region of the baseline mask. I recorded regions that did not overlap with any region in the corresponding mask, and regions that overlapped with more than one region in the corresponding mask.

5.3.1.3 Sensitivity Analysis Results

The results of the analysis of variance showed that there was not sufficient evidence to reject the null hypothesis at $p < 0.05$, for all parameter combinations or for LI, UI, NT, and SDT alone. Table 5.1 shows the ***F* value** for each parameter or parameter combination, and the ***p* value**, or the observed significance level [78]. The *F* value is the ratio of the between-group variation and the within-group variation for the treatments; the *p* value is the probability of observing a value that is contradictory to the null hypothesis when the null hypothesis is true. The table shows the extent to which the parameter or parameter–level combinations explain the differences observed between the groups. From the table, we see that, if we reject the null hypothesis for each parameter or parameter combination—we assume that the parameter or parameter–level combination does not explain the differences in the NVMs for the 16 treatments—then we risk making errors at the rates specified in the third column. Where we do not have sufficient confidence to reject the null hypothesis, we accept it.

Table 5.1 *F* values for parameters and parameter interactions.

Parameter (or interaction)	*F* value	Pr > *F*
LI	1.01	0.3156
UI	0.78	0.3777
LI*UI	0.35	0.5552
NT	1.34	0.2471
LI*NT	1.48	0.2237
UI*NT	1.25	0.2639
LI*UI*NT	0.99	0.3189
SDT	1.56	0.2123
LI*SDT	0.83	0.3636
UI*SDT	1.87	0.1722
LI*UI*SDT	0.84	0.3605
NT*SDT	1.06	0.3030
LI*NT*SDT	1.08	0.2996
UI*NT*SDT	1.23	0.2668
LI*UI*NT*SDT	0.82	0.3654

Parameter interactions are designated by ’*’. Pr>*F* denotes the area under the curve of the *F* distribution that is to the right of the *F* value; Pr>*F* is the probability of rejecting the null hypothesis when it is true. The unit of analysis is a region. The dependent variable is NVM.

To draw valid conclusions from the analysis of variance test about the sensitivity of IVSeg’s results to changes in program parameters, I needed to verify that the power $(1-\beta)$ of the test was sufficient—that the test would have detected considerable ratios of the between-group variation and the within-group variation, if such ratios immerged from the data. The power calculation (see [79]) considered four factors—LI, UI, NT, and SDT—at two levels each, and 33 observations for each factor-level combination (this is the minimum number of observation assuming a single region per image). I assumed that the variance in the underlying populations was close to zero because NVM for baseline images relative to themselves was zero. The power calculations showed that, at $\alpha = 0.05$, $1-\beta$ was greater than 80 percent. I concluded that the analysis of variance had sufficient power to detect any existing significant effects due to 5-percent changes in IVSeg parameters.

I studied the distribution of NVM for the set of observations. The total number of observations (regions) was 1293. There were 1244 regions in the baseline images that had one overlapping region in the corresponding treatment image. For 416 (33.4 percent) of the overlapping regions the NVM was 0. For 882 (70.9 percent) of the overlapping regions, the error was smaller than 0.05. Figure 5.19 shows the histogram of the errors for the 1242 (99.8 percent) of the regions, for which the error was less than 1. There were 44 observations with mismatching objects, all of which originated from four different structures. There were five observations of regions that overlapped with more than one region in the corresponding treatment image, all of which originated from one structure.

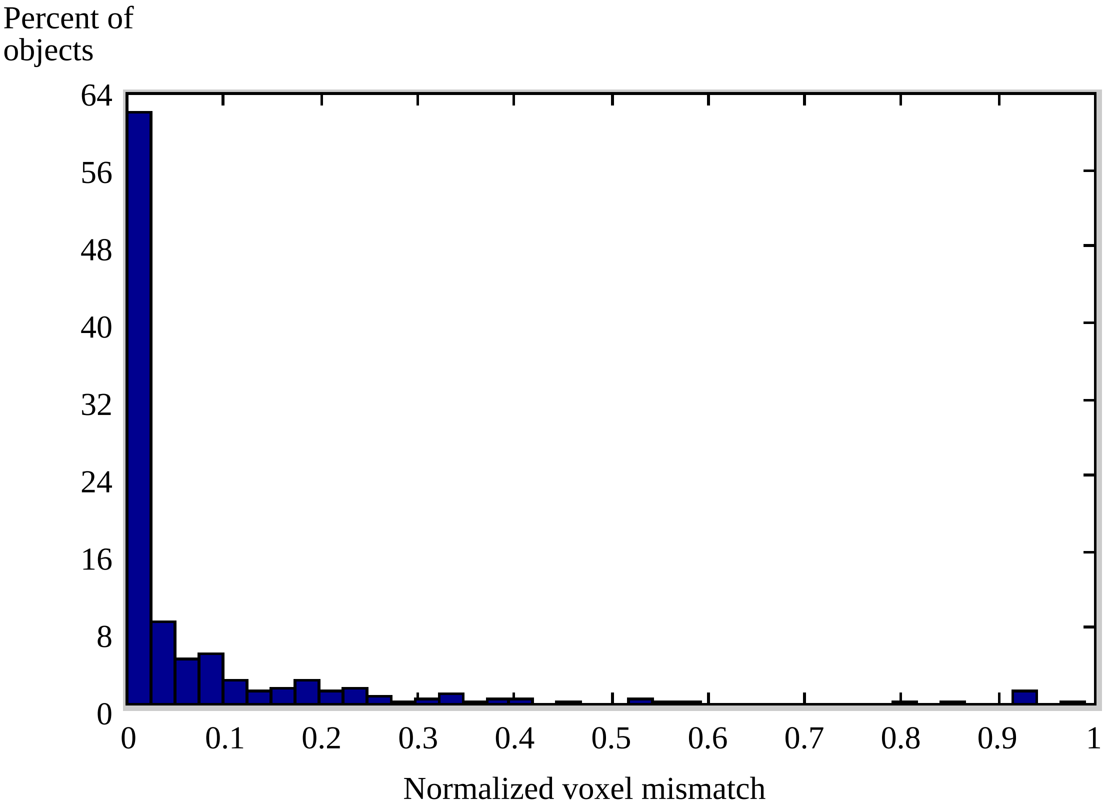


Figure 5.19 Histogram of the NVM for the 1242 objects that had an overlapping object in the corresponding image and for which $NVM \le 1$. Note that about 71 percent of the objects had a normalized-voxel–mismatch value smaller than or equal to 0.05.

5.3.1.4 Sensitivity Analysis Conclusions

The preceding analysis revealed that most of the regions within the treatment images differed by less than 5 percent from their counterparts in the baseline images. The analysis showed that IVSeg's procedure for identification of salient regions was not sensitive to 5-percent changes in the parameters LI, UI, SDT, and NT. These finding suggest that, for CTA data, the initial setting of LI, UI, SDT, and NT is likely to be adequate. Comparison of IVSeg's Procedure for Identification of Salient Regions and Intensity Thresholding

IVSeg's procedure for identification of salient regions and conventional thresholding both use a threshold value to make decisions about object membership of voxels. However, conventional thresholding performs thresholding in intensity space, whereas IVSeg performs thresholding in shape space. In Sections 5.3.1.5 to 5.3.1.8 I show that the results of IVSeg's procedure are less sensitive to threshold changes than are those of conventional thresholding.

5.3.1.5 Comparison Study Design

For this study, I used the same raw dataset and measures as in the sensitivity analysis (see Section 5.3). First, I ran IVSeg 11 times, each time with a different value for SDT. The values for SDT ranged from 0.035625 to 0.059375, and represented 5-percent increments over a predefined center-point value of 0.0475. The fixed parameter values were LI=49 HU, UI=598 HU, and NT=170. I generated masks that included blood-vessel regions only, as I had for the sensitivity analysis. I regarded the results that IVSeg produced for SDT=0.0475 as the baseline images, and the results that IVSeg produced for all other SDT values as treatment images. I computed the NVM for objects in corresponding baseline and treatment images.

Second, I ran a program for intensity thresholding for the same raw dataset 11 times, each time with a different intensity threshold. The intensity thresholds ranged from 150 HU to 250 HU, and represented 5-percent increments over a predefined center point of 200 HU. Then, I generated masks that included blood-vessel regions only, as I had for the sensitivity analysis. I regarded the results from the threshold of 200 HU as the baseline images, and the results from all other threshold values as treatment images. I computed the NVM for objects in corresponding baseline and treatment images.

Third, for each increment, I performed a paired *t* test to test the hypothesis that *the mean NVM (MNVM) values for IVSeg were smaller than those for conventional thresholding*. The paired *t*-test compared the MNVM for corresponding images, one resulting from intensity thresholding and the other resulting from activation of IVSeg. I considered for the test only those regions that had overlapping counterparts in the baseline images (94 percent for conventional thresholding results, and 99 percent for IVSeg's results).

5.3.1.6 Comparison Results

The paired *t*-test showed that, for all increments relative to the center-point value, the MNVM was smaller for IVSeg's images than for the images generated with conventional thresholding. From Table 5.2, we see that the differences between MNVMs of corresponding images for positive increments were statistically significant at $p < 0.001$. Figure 5.20 shows a summary plot of the results. In general, the mean difference between MNVMs that resulted from the two methods was greater for negative increments than for positive increments. For negative increments, the mean differences and the standard error of the mean increased with the magnitude of the increment. For positive increments, there was merely a trend of increasing differences with increasing increments.

Table 5.2 Results of paired *t*-test for comparison of conventional thresholding and IVSeg.

Percent increment	Mean of differences	Standard error	*T* value	Prob >|*T*|
-25	13.7001805	9.8211780	1.3949631	0.1723
-20	12.3151830	9.4384874	1.3047835	0.2010
-15	11.5250592	8.8199836	1.3066985	0.2003
-10	11.3400731	8.2644172	1.3721564	0.1793
-5	9.6773792	7.6458397	1.2657052	0.2145
5	0.1235062	0.0325463	3.7947846	0.0006
10	0.1663892	0.0386262	4.3076737	0.0001
15	0.2217254	0.0452144	4.9038684	0.0001
20	0.2428929	0.0451164	5.3837000	0.0001
25	0.2399278	0.0439518	5.4588853	0.0001

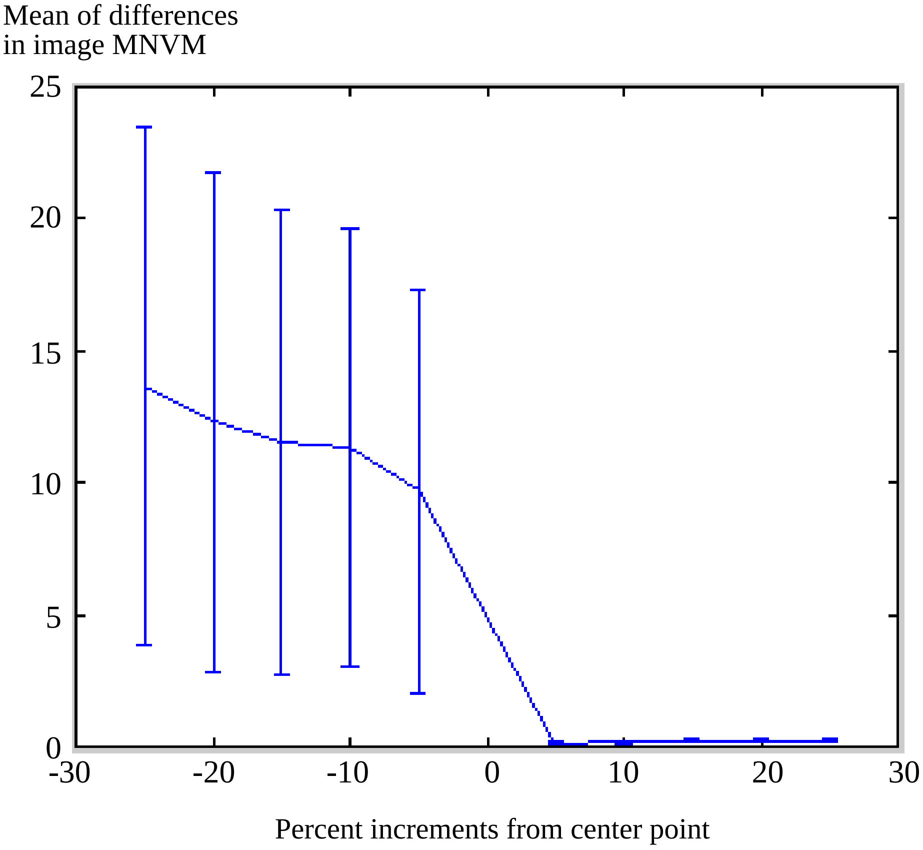


Figure 5.20 Error-bar plot of mean difference of image MNVM.

5.3.1.7 Comparison Conclusions

The sensitivity-comparison experiment showed that IVSeg's method for finding object boundaries in shape space was less sensitive to changes in threshold values than was conventional thresholding in intensity space. Because of large standard-error values for negative increments, the results are statistically significant for only positive increments. The large standard-error values are caused by considerable leaks into large bone structures in several threshold images—for example, see Figure 5.21. For negative increments, due to the large standard error, if we reject the null hypothesis that the two methods' sensitivities are not different, we risk making an error with Prob>|*T*|, as specified in Table 5.2.

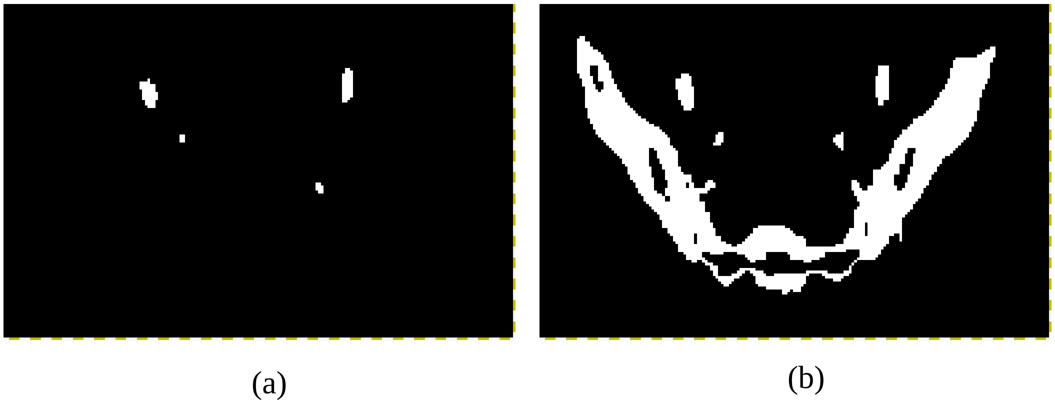


Figure 5.21 Leaks from fine branches of the iliac arteries into the pelvic bone. (a) The baseline image. (b) A corresponding treatment image with a leak.

5.3.1.8 Repeat of Statistical Analysis for Observations without Leaks

In the comparison of IVSeg's procedure for identification of salient regions and intensity thresholding, results were not statistically significant for negative increments because of large standard errors that resulted from leaks. I repeated the statistical analysis for all the structures for which there were no leaks into adjacent structures in any of the observations (I excluded a total of 51 observations). Table 5.3 shows the results of the repeat paired *t* -test; Figure 5.22 shows an error-bar plot for the results of the repeat test. From the table and error-bar plot, we see that, in general, the mean differences and standard error of the mean for MNVM increased with an increase in the magnitude of the increment. The repeat statistical analysis showed that, where leaks to adjacent structures did not occur, IVSeg's method for finding object boundaries in shape space was less sensitive to changes in threshold values compared to conventional thresholding in intensity space. These results were statistically significant at $p < 0.025$ for all increments.

Table 5.3 Results of paired *t*-test for repeated comparison of conventional thresholding and IVSeg (for structures without leaks).

Percent increment	Mean of differences	Standard error	*T* value	Prob> |*T*|
-25	0.9838334	0.4080844	2.4108574	0.0218
-20	0.3413646	0.1000427	3.4121885	0.0018
-15	0.2854101	0.0871485	3.2749844	0.0025
-10	0.2389239	0.0742975	3.2157736	0.0030
-5	0.1093748	0.0438471	2.4944592	0.0180
5	0.1104296	0.0312476	3.5340196	0.0013
10	0.1541976	0.0377041	4.0896801	0.0003
15	0.2101106	0.0446525	4.7054570	0.0001
20	0.2240587	0.0444016	5.0461802	0.0001
25	0.2201652	0.0432196	5.0941033	0.0001

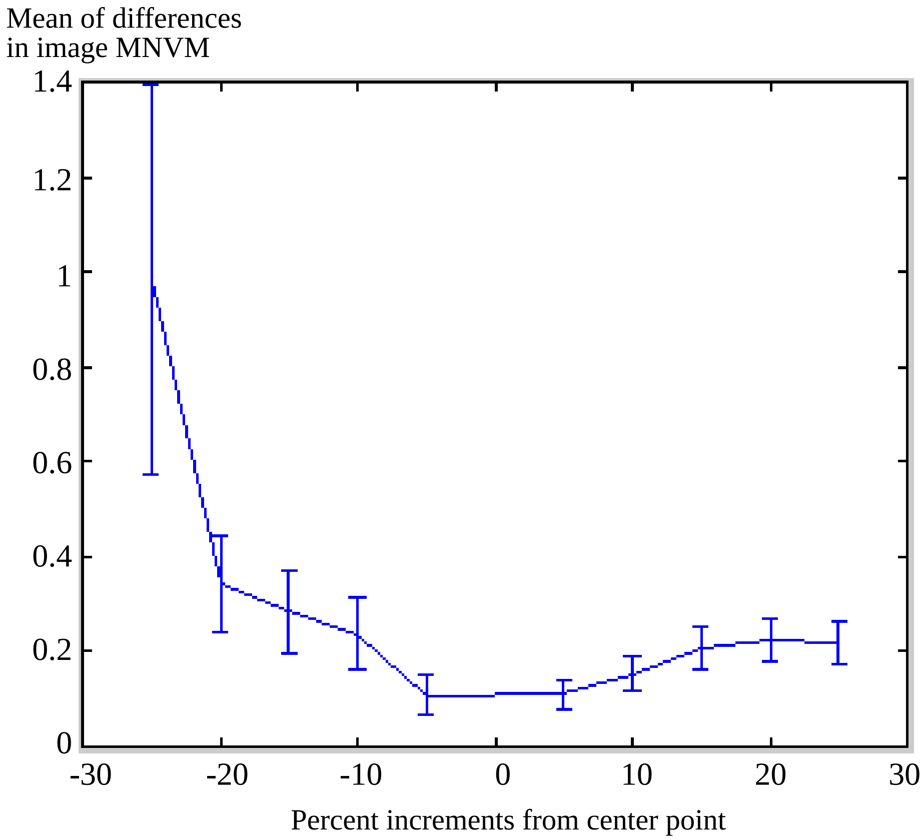


Figure 5.22 Error-bar plot of mean difference of image MNVM for the repeat statistical analysis.

5.4 Summary

In this chapter, I described the theoretical foundation and the applications of the methods that IVSeg uses to find coherent regions. Then, I examined the performance of the method in two ways. First, I tested the sensitivity of the method to 5-percent changes in parameter values; I found that the method was not sensitive to these changes. Second, I compared the sensitivity of IVSeg's method to changes in the value of SDT to the sensitivity of conventional thresholding to changes in the intensity threshold; I found that IVSeg's method was more robust than conventional thresholding to changes in the respective threshold values.

The salient regions that the analysis of isolabel-contour maps produces are amenable for direct manipulation by the user; for example, the user can easily generate object masks by selecting relevant salient regions with a mouse click. However, IVSeg groups salient regions programmatically. In Chapter 6, I describe IVSeg's method for reconstructing vessel structures by grouping regions from segmented sequences of 2D sections.

Chapter 6

Dynamic Region-Overlap Criteria for Extraction of Vessel Structures from a Sequence of 2D Sections

IVSeg's analysis of contour maps results in images with salient regions that the user can easily manipulate by clicking with a mouse. Editing via manual selection of individual regions is likely to save editing time compared to manual tracing. However, IVSeg achieves greater time savings by having the user indicate the target vessel with only a few mouse clicks, and by grouping regions that belong to the target vessel (Figure 6.1). IVSeg selects related regions by examining the connectivity of the regions (see Section 1.2.4) in the through-plane direction. To avoid leaks between unrelated objects whose regions overlap in the through-plane direction (Figure 6.2), IVSeg complements connectivity with region-overlap criteria, which are based on the expected overlap between regions from consecutive sections that belong to the same vessel. IVSeg computes these criteria dynamically from geometric parameters that the program estimates from the input data.

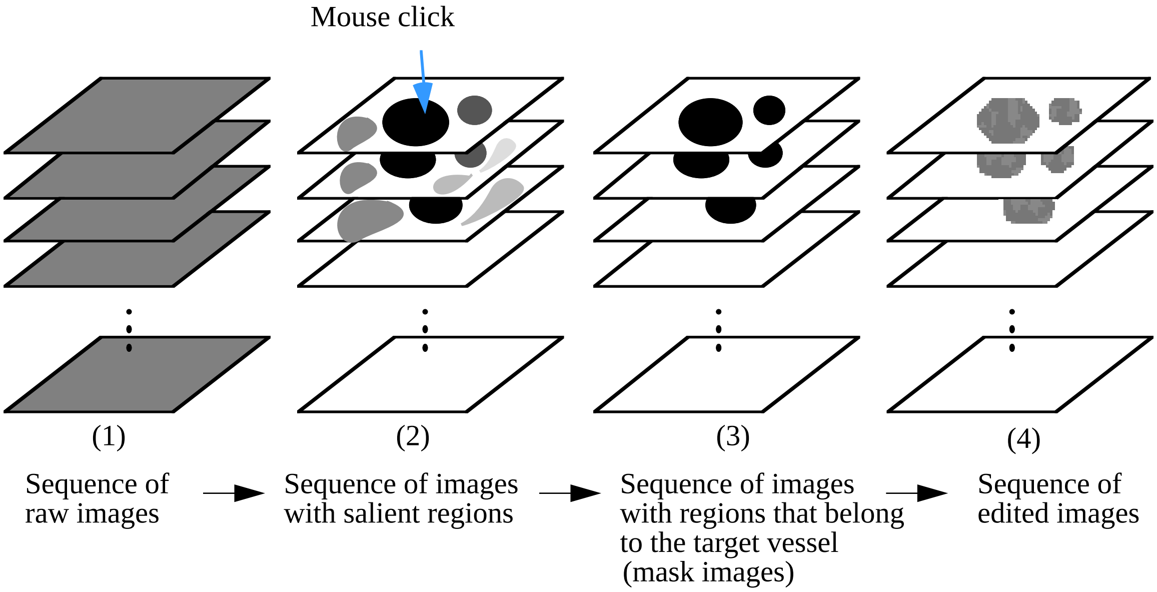

Mouse click
(1)
(2)
(3)
(4)
Sequence of raw images
Sequence of images with salient regions
Sequence of images with regions that belong to the target vessel (mask images)
Sequence of edited images

Figure 6.1 Intermediate results in the course of IVSeg's operation (see corresponding elliptic boxes in the flow chart of Figure 1.7). To form the sequence of edited images on the right, the user clicks on one of the dark regions within the first image on the second column. The region-grouping procedure allows the transition from the second sequence to the third sequence.

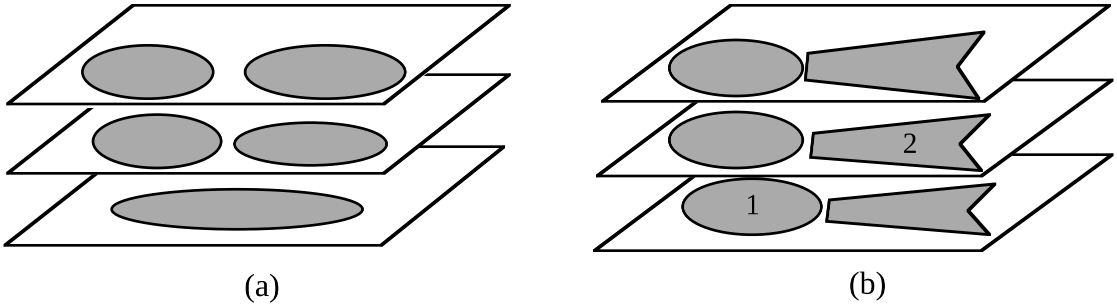


Figure 6.2 Hypothetical connected regions in consecutive CTA sections. (a) A bifurcation within a single vessel. Application of conventional connectivity within the object would correctly include all regions. (b) Very close objects with similar intensity—for example, the aorta (elliptic regions) and the spine (polygonal regions). Because regions that belong to different objects are connected in the *z*

direction—for example, region 1 and region 2—conventional connectivity will cause a leak from one object into the other.

The organization of the current chapter follows. Section 6.1 presents the theory that underlies the use of dynamic region-overlap criteria for vessel extraction. The theory pertains to the formalization of region-overlap criteria and to the applicability of these criteria to analysis and grouping of segmented regions in sequences of medical images. Section 6.2 describes algorithms, based on the preceding theory, that I incorporated into IVSeg's region-grouping procedure. Section 6.3 reports an experiment that I conducted to test the robustness of the region-grouping procedure with respect to change of program parameters.

6.1 Theory

The following analysis of the relationship between regions in consecutive sections is based on a framework that models vessel structures as a composition of cylindrical components of various sizes that are oriented along the principal axis of the vessel (Figure 6.3), in accordance with the generalized-cylinder modeling scheme [45, 48, 49]. The framework seems appropriate for modeling vessels in CTA volumes because, with the exception of branching vessels and vessels for which diameter changes rapidly, blood vessel are cylindrical. Like planes that intersect a cylinder's axis at only one point, planes that intersect the vessel axis at only one point typically have elliptic shape.

For elliptic regions in consecutive images with salient regions, we postulate that the regions result from intersections of consecutive image planes with the same vessel, and that therefore the voxels within these regions belong to the same vessel. To test this hypothesis, we fit a compositions of cylinders to the regions and then check whether the regions overlap to the extent that we would expect for sections of the fitted cylinders. In Section 6.1.2 I demonstrate that we can apply this method to all voxels within a vessel structure by showing that any vessel voxel belongs to an elliptic region within one of three orthogonal reformations of the data.

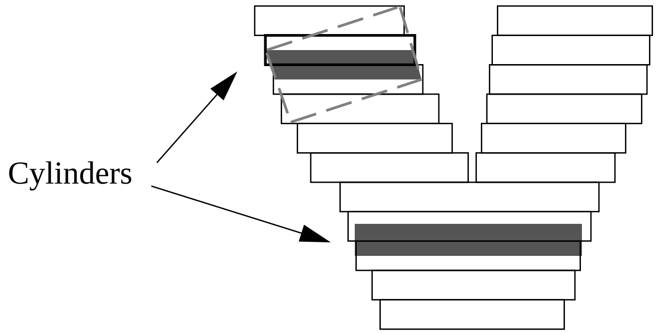


Figure 6.3 Modeling of a vessel structure as a composition of small cylinders. The transparent rectangles represents salient regions. The gray polygons represent the volume that is enclosed between intersections of the image planes with a cylinder model that we fit to salient regions in consecutive images.

The following discussion explains the dynamic overlap criteria that I suggest for analysis of regions in consecutive sections. Section 6.1.1 describes how we can determine whether a region is elliptic and thus is amenable to modeling as a cylinder section and to application of the overlap criteria. Section 6.1.2 shows that, for voxels within regions that are nonelliptic in the acquisition plane, we can find elliptic regions in orthogonal reformations of the data, and we can apply the overlap criteria for the latter ellipses. Section 6.1.3 provides a formal expression for the overlap between cylinder sections and explains how we derive dynamic overlap criteria using this expression.

6.1.1 Testing Whether a Region Is Elliptic Via an Elliptic Approximation of a Contour

To determine whether a region is elliptic, we can approximate the shape of the contour with an ellipse and compute the approximation error. We can define the ellipse in terms of the first harmonic of two Fourier series expansions—one expansion for each of the contour projections along the x and y axes [69]. The Fourier series expansion for the projection $x(t)$ is

$$x(t)\left(= A_0 + \sum_{n=1}^{T-1}\left(a_n\cos\frac{2n\pi t}{T} + b_n\sin\frac{2n\pi t}{T}\right)\right); \quad t \ge 0\,; \tag{6.1}$$

the expansion for the projection $y(t)$ is,

$$y(t)\left(= C_0 + \sum_{n=1}^{T-1}\left(c_n\cos\frac{2n\pi t}{T} + d_n\sin\frac{2n\pi t}{T}\right)\right); \quad t \ge 0\,; \tag{6.2}$$

where T is the period, n are harmonics, and

$$A_0 = \frac{1}{T}\sum_{t=0}^{T-1} x(t)\ ,$$

$$a_n = \frac{2}{T}\sum_{t=0}^{T-1} x(t)\cos\frac{2n\pi t}{T}\,,$$

$$b_n = \frac{2}{T}\sum_{t=0}^{T-1} x(t)\sin\frac{2n\pi t}{T}\,,$$

$$C_0 = \frac{1}{T}\sum_{t=0}^{T-1} y(t)\ ,$$

$$c_n = \frac{2}{T}\sum_{t=0}^{T-1} y(t)\cos\frac{2n\pi t}{T}\,,$$

$$d_n = \frac{2}{T}\sum_{t=0}^{T-1} y(t)\sin\frac{2n\pi t}{T}\,.$$

This representation can be viewed as a decomposition of the contour into a sum of rotating phasors, each of which defines an ellipse [20]. Thus, the pair of corresponding terms for $n = 1$ in the summations of Equations 6.1 and 6.2 defines an ellipse.

We compute the truncated expansion of the projection of a contour on the x axis from Equation 6.1 by computing a_1 and b_1 using the formulas

$$a_1 = \frac{T}{2\pi^2}\sum_{p=1}^{k}\frac{\Delta x_p}{\Delta t_p}\left[\cos\frac{2\pi t_p}{T} - \cos\frac{2\pi t_{p-1}}{T}\right],$$

$$b_1 = \frac{T}{2\pi^2} \sum_{p=1}^{k} \frac{\Delta x_p}{\Delta t_p} \left[\sin \frac{2\pi t_p}{T} - \sin \frac{2\pi t_{p-1}}{T} \right],$$

where k is the number of single-coordinate projections that we obtain via a walk along the contour, Δt is the time it takes to complete a single step. In our case $\Delta t_i = 1$, Δx_i are the changes in the x coordinate with each step, and

$$t_p = \sum_{i=1}^{p} \Delta t_i .$$

We compute the truncated expansion of the projection of a contour on the y axis from Equation 6.2 by computing c_1 and d_1 in a similar manner.

We compute the magnitude of the axes using the following procedure [69]: Given a pair of Fourier series expansions of a contour at an arbitrary starting point $t = 0$, let the coefficients that correspond to the first harmonic in the two expansions be a_1, b_1 and c_1, d_1. The terms

$$x_1 = a_1 \cos\theta + b_1 \cos\theta ,$$

$$y_1 = c_1 \cos\theta + d_1 \cos\theta ,$$

where $\theta = \frac{2\pi t}{T}$, and $0 \le t \le T$ define an ellipse, and $(x_1(t), y_1(t))$ denote the position of the rotating phasor at time t on the ellipse. To compute the magnitude of the semimajor axis of the ellipse, we derive a pair of new expansions, from a starting point $t^* = 0$ on the contour, such that the phasor is aligned with the ellipse semimajor axis. Then, we evaluate an expression for the magnitude of the phasor at $t^* = 0$.

To derive the new expansions, we compute the angular rotation that maximizes the magnitude of the first harmonic phasor $E = \sqrt{x_1^2 + y_1^2}$ by setting the first derivative of the magnitude equal to zero, yielding

$$\theta_1 = \frac{1}{2}\left(\operatorname{atan} \frac{2(a_1 b_1 + c_1 d_1)}{a_1^2 + c_1^2 - b_1^2 - d_1^2} \right).$$

Then, we derive the Fourier coefficients for the new expansions:

$$\begin{bmatrix} a_1^* & c_1^* \\ b_1^* & d_1^* \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} a_1 & c_1 \\ b_1 & d_1 \end{bmatrix} .$$

The ellipse is defined in terms of the new coefficients by

$$x_1^*(t^*) = a_1^* \cos\frac{2\pi t^*}{T} + b_1^* \sin\frac{2\pi t^*}{T} , \qquad (6.3)$$

$$y_1^*(t^*) = c_1^* \cos\frac{2\pi t^*}{T} + d_1^* \sin\frac{2\pi t^*}{T} . \qquad (6.4)$$

We can find the magnitude of the ellipse semimajor and semiminor axes by evaluating Equations 6.3 and 6.4 at $t^* = 0$ and $t^* = \frac{T}{4}$, respectively:

$$E^*(0) = \sqrt{x_1^*(0)^2 + y_1^*(0)^2} = \sqrt{a_1^{*2} + c_1^{*2}} ,$$

$$E^*\left(\frac{T}{4}\right) = \sqrt{x_1^*\left(\frac{\pi}{2}\right)^2 + y_1^*\left(\frac{\pi}{2}\right)^2} = \sqrt{b_1^{*2} + d_1^{*2}} .$$

In the following discussion, I show that the ellipse that is defined by Equations 6.3 and 6.4 is an optimal elliptic approximation to the contour. For simplicity, the discussion uses the Fourier series decomposition of the contour into a sum of exponential terms.

Theorem: The elliptic approximation to a contour that is defined in terms of the first harmonic of two Fourier series expansions—one expansion for each of the contour projections along the x and y axes—is an optimal approximation: The approximation error is minimal. The error is the second norm of the differences of the contour and ellipse functions.

Proof: Let the contour coordinates be

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{n=0}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta nt} . \tag{6.5}$$

Let the optimal ellipse be

$$M_{E_1} = \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t} , \tag{6.6}$$

Assume another ellipse that is related to the optimal ellipse via a sequence of three spatial transformations, without loss of generality—a translation, scaling, and rotation:

$$M_{E_2} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \left(\begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t} \right) . \tag{6.7}$$

From Equation 6.7 we get

$$M_{E_2} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t} . \tag{6.8}$$

The error for the contour approximation via M_{E_1} is

$$\left\| M_c - M_{E_1} \right\| = \left\| \begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \sum_{n=2}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta nt} \right\| . \tag{6.9}$$

If we rewrite the expression for M_c,

$$M_c = M_{E_2} + M_{E_1} - M_{E_2} + \begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \sum_{n=2}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta nt} ,$$

we get the following expression for the error of the contour approximation via M_{E_2}:

$$\left\|M_c - M_{E_2}\right\| = \left\|M_{E_1} - M_{E_2} + \begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \sum_{n=2}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta n t}\right\|. \tag{6.10}$$

We show in Appendix 1 that the right hand side of Equation 6.10 is a sum of orthogonal exponential terms, and therefore

$$\left\|M_{E_1} - M_{E_2} + \begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \sum_{n=2}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta n t}\right\| = \left\|M_{E_1} - M_{E_2}\right\| + \left\|\begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \sum_{n=2}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta n t}\right\|.$$

We conclude that

$$\left\|M_c - M_{E_1}\right\| < \left\|M_c - M_{E_2}\right\|.$$

In Section 6.1.2, I show that cylindrical components are guaranteed to have elliptic sections in at least one of three orthogonal reformation of the data.

6.1.2 A Proof that Right Cylinders with Circular Bases Have Elliptic Sections

We observe that planes that intersect the axis of a right cylinder with a circular base result in elliptic regions. Nonelliptic regions in the acquisition plane result from one, and only one, of the following situations:

1. The regions belong to an object whose shape is different from a right cylinder with circular bases.
2. The regions belong to a right-cylinder object, but the acquisition plane does not cross the cylinder axis at only one point.

Figure 6.4 illustrates situation 2: For in-plane branches and in-plane bends of tortuous vessels, which can be modeled as compositions of right cylinders, but whose shape is nonelliptic in the acquisition plane, the acquisition plane intersects the vessel axis at multiple points or at no points at all.

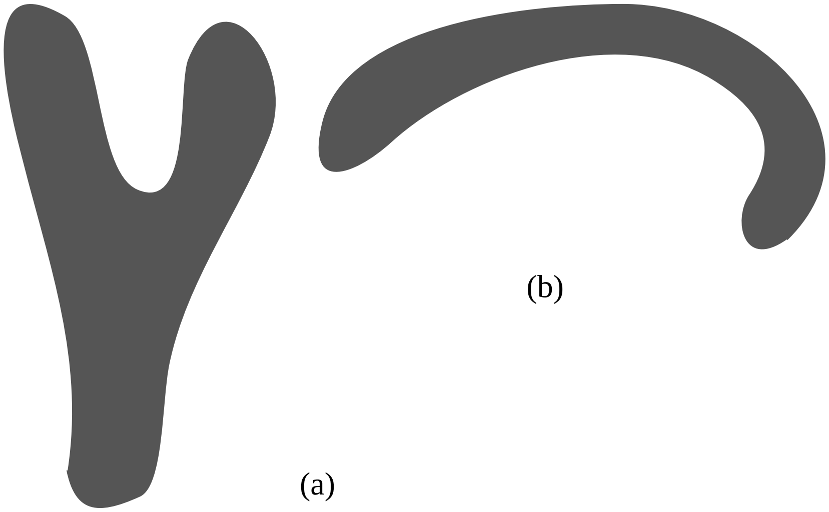


Figure 6.4 Nonelliptic sections of hypothetical tortuous blood vessels. (a) An in-plane branch. (b) An in-plane bend.

Intersection of the acquisition planes with in-plane bends and in-plane branches results in nonelliptic regions. However, intersection with these substructures of hypothetical planes whose orientation is orthogonal to the image plane results in elliptic regions (Figure 6.5). I support the preceding statement with a proof. To simplify the discussion, I formulate the proof for infinite cylinders. If we assume that we can use cylinder components of arbitrary length to represent the vessel structure, we can adapt the proof to hold for cylindrical components of finite length.

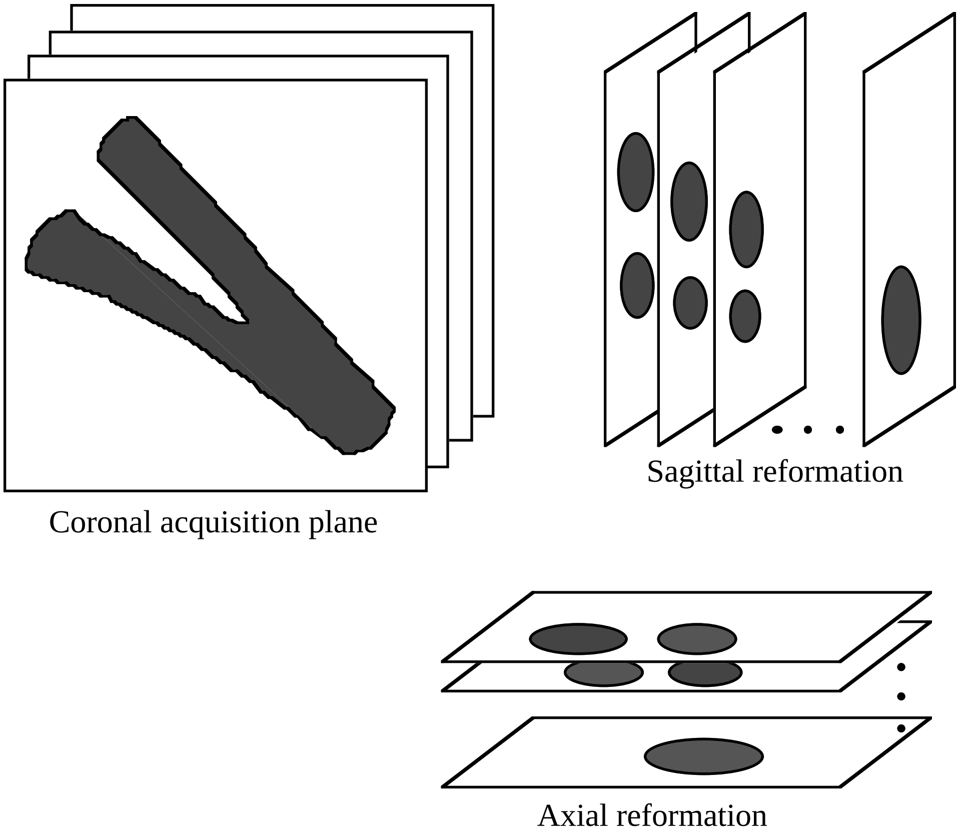


Figure 6.5 Sections of an in-plane branching vessel in orthogonal reformations[1] of the data. Note that the vessel cross sections are nonelliptic in the acquisition plane and are elliptic in the orthogonal reformations of the data.

Lemma: Given an infinite right cylinder that has a circular base and an arbitrary infinite plane that intersects the cylinder, the region of intersection is elliptic if and only if the plane intersects the cylinder axis at one and only one point (proof omitted).

Theorem: Assume that we have a point s inside an infinitely long cylinder within an arbitrary coordinate system xyz, and three vectors $\underset{\sim}{x}', \underset{\sim}{y}', \underset{\sim}{z}'$ that intersect at s and are parallel to the coordinate axes (Figure 6.6). The intersection of the cylinder with the three planes that the vectors $\underset{\sim}{x}', \underset{\sim}{y}', \underset{\sim}{z}'$ define results in at least one elliptic cylinder section.

[1] A reformation of a dataset is the process of rearranging the dataset in a new coordinate system.

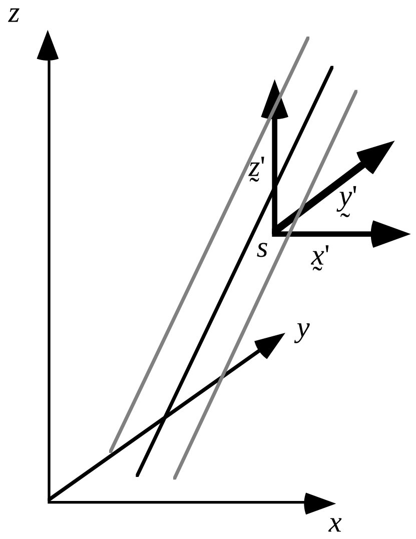


Figure 6.6 A cylinder in an arbitrary coordinate system.

Proof: We can characterize the conditions for intersection between the cylinder axis and an arbitrary plane as follows. Assume that the cylinder axis passes through a point $p = (x_1, y_1, z_1)$ and has the vector equation $\underset{\sim}{r} = k\underset{\sim}{v}$, where $\underset{\sim}{r} = (x - x_1, y - y_1, z - z_1)$ is the vector from p to any other point (x, y, z) on the axis, $k \neq 0$ is a factor of proportionality, and $\underset{\sim}{v} = (a, b, c)$ is the direction vector for the axis. Assume that an arbitrary plane passes through a point $q = (x_2, y_2, z_2)$ and is perpendicular to a vector $n = (A, B, C)$ that passes through q. The vector equation of the plane is $\underset{\sim}{n} \bullet \underset{\sim}{r} = 0$, where $\underset{\sim}{r} = (x - x_2, y - y_2, z - z_2)$ is the vector that the directed line segment from q to any arbitrary point (x, y, z) in the plane represents. All points (x, y, z) at the intersection of the plane and the cylinder axis satisfy

$$(x - x_1, y - y_1, z - z_1) = k\underset{\sim}{v}, \qquad (6.11)$$

and

$$(x - x_2, y - y_2, z - z_2) \bullet (A, B, C) = Ax + By + Cz + D = 0. \qquad (6.12)$$

From Equations 6.11 and 6.12, we get

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} ka - x_1 \\ kb - y_1 \\ kc - z_1 \end{bmatrix} = k\underset{\sim}{v} + p\,, \tag{6.13}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \bullet \underset{\sim}{n} = -D\,. \tag{6.14}$$

From Equations 6.13 and 6.14, we get

$$(k\underset{\sim}{v} + p) \bullet \underset{\sim}{n} = -D\,,\ k\underset{\sim}{v} \bullet \underset{\sim}{n} + p \bullet \underset{\sim}{n} = -D\,. \tag{6.15}$$

From Equations 6.12 and 6.15, we get

$$k\underset{\sim}{v} \bullet \underset{\sim}{n} = \underset{\sim}{n} \bullet (q - p)\ . \tag{6.16}$$

We assume that $\underset{\sim}{n} \neq 0$ and $\underset{\sim}{v} \neq 0$. We can differentiate two cases:

1. For $\underset{\sim}{v} \bullet \underset{\sim}{n} = 0$—that is, for cases where $\underset{\sim}{v}$ is parallel to the arbitrary plane—if $q = p$, Equation 6.16 holds for all k; thus, we conclude that the plane intersects all cylinder-axis points. If $q \neq p$, Equation 6.16 does not hold for any k; thus, we conclude that the plane does not intersect any cylinder-axis point.
2. If $\underset{\sim}{v} \bullet \underset{\sim}{n} \neq 0$,

$$k = \frac{\underset{\sim}{n} \bullet (q - p)}{\underset{\sim}{v} \bullet \underset{\sim}{n}}\,,$$

and we conclude that the arbitrary plane intersects exactly one cylinder-axis point.

The three planes defined by $\underset{\sim}{x}', \underset{\sim}{y}', \underset{\sim}{z}'$ are orthogonal, and therefore $\underset{\sim}{x}' \bullet \underset{\sim}{y}' = 0$, $\underset{\sim}{y}' \bullet \underset{\sim}{z}' = 0$, $\underset{\sim}{x}' \bullet \underset{\sim}{z}' = 0$. There are three possible mutually exclusive spatial relations between $\underset{\sim}{v}$ and $\underset{\sim}{n}_i \in \{\underset{\sim}{x}', \underset{\sim}{y}', \underset{\sim}{z}'\}$:

a. $\underset{\sim}{v} \| \underset{\sim}{n}_i \Rightarrow (\underset{\sim}{v} \bullet \underset{\sim}{n}_i \neq 0) \wedge (\underset{\sim}{v} \bullet \underset{\sim}{n}_j = 0) \wedge (\underset{\sim}{v} \bullet \underset{\sim}{n}_k = 0)$, $i \neq j \neq k$; therefore, from cases 1 and 2, one plane intersects the cylinder axis at exactly one point.

b. $\underset{\sim}{v} \perp \underset{\sim}{n}_i \Rightarrow (\underset{\sim}{v} \bullet \underset{\sim}{n}_i = 0) \wedge (\underset{\sim}{v} \bullet \underset{\sim}{n}_j \neq 0) \wedge (\underset{\sim}{v} \bullet \underset{\sim}{n}_k \neq 0)$, $i \neq j \neq k$; therefore, from cases 1 and 2, two planes intersect the cylinder axis at exactly one point.

c. $\underset{\sim}{v}$ is not orthogonal to any of $\underset{\sim}{x}', \underset{\sim}{y}', \underset{\sim}{z}'$. Thus $(\underset{\sim}{v} \bullet \underset{\sim}{n}_i \neq 0) \wedge (\underset{\sim}{v} \bullet \underset{\sim}{n}_j \neq 0) \wedge (\underset{\sim}{v} \bullet \underset{\sim}{n}_k \neq 0)$, $i \neq j \neq k$; therefore, from case 2, all three planes intersect the cylinder axis at exactly one point.

Because one of a, b, or c must hold, at least one of the planes intersects the cylinder axis at exactly one point. From the lemma, the intersection of the cylinder and the planes that the vectors $\underset{\sim}{x}', \underset{\sim}{y}', \underset{\sim}{z}'$ define results in at least one elliptic section.

Q.E.D.

Corollary: In a sequence of images of blood vessels, each vessel voxel belongs to an elliptic region in at least one of three reformations of the data.

Section 6.1.3 presents and demonstrates a formal expression for the overlap of sections through a right cylinder with a circular base.

6.1.3 The Overlap of Consecutive Sections Through a Right Cylinder

The overlap of elliptic regions that result from the intersection of a cylinder with consecutive image planes is a function of the cylinder diameter, the slant angle, the section thickness, and the spacing between sections. Figure 6.7 shows a 2D projection of sections across a cylinder from a direction perpendicular to the direction of the cylinder slant.

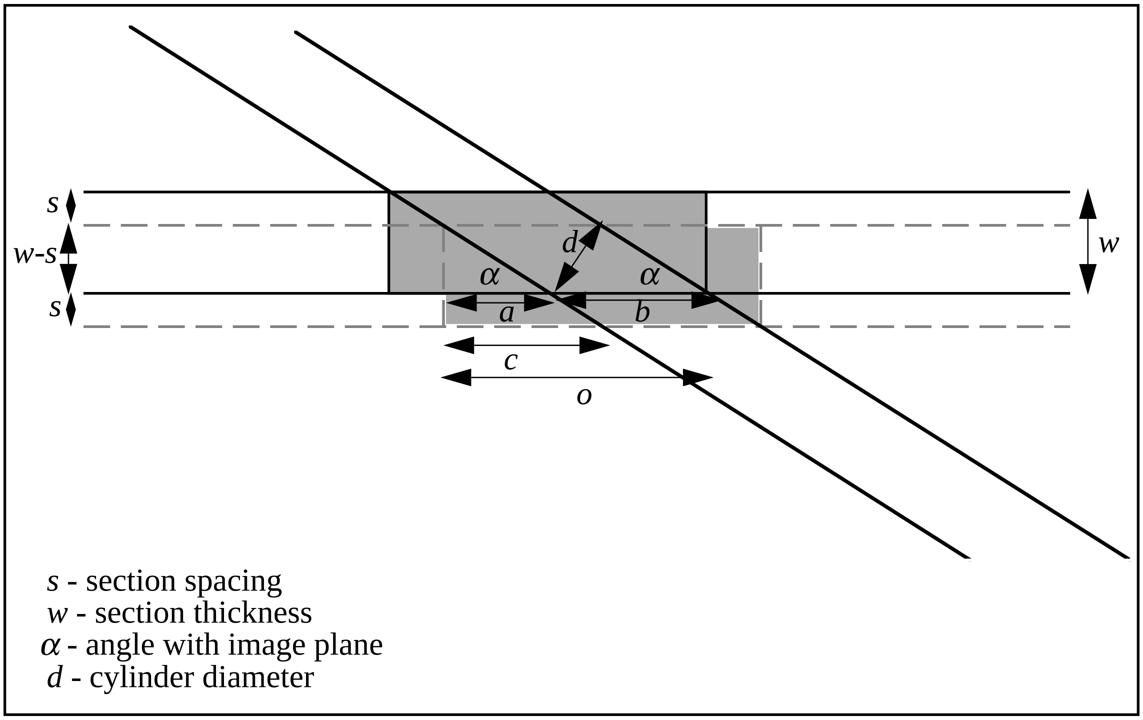


Figure 6.7 Overlap between consecutive cylinder sections, viewed from a direction perpendicular to the direction of the cylinder slant. The gray slabs indicate the extent of the cylinder section along the direction of the slant. The variables a, b, c, and o represent distances.

6.1.3.1 Derivation of the Relative-Overlap Expression

We can calculate the ellipse overlap by examining the overlap of the semimajor axes of the elliptic regions, and then deriving the area of ellipse overlap from the semimajor axes overlap. Assume that a cylinder of diameter d intersects two consecutive sections at an angle α (see Figure 6.7). Let the section width be w, and the section spacing be s. The axes of the ellipses are scaled as a result of window and level settings that the user sets for image display. Let the scaling factor be k. We can compute the slab overlap via the following derivation:

$$o = k(a+b) = k\left(\frac{w-s}{|\tan\alpha|} + \frac{d}{|\sin\alpha|}\right) = k\left(\frac{(w-s)|\cos\alpha| + d}{|\sin\alpha|}\right). \tag{6.17}$$

The semimajor axis of the ellipse is

$$e = k\left(\frac{b+c}{2}\right) = k\left(\frac{d + w|\cos\alpha|}{2|\sin\alpha|}\right); \tag{6.18}$$

the semiminor axis is

$$f = k\left(\frac{d}{2}\right). \tag{6.19}$$

Because the ellipses in consecutive sections are identical, we can compute the area of one quadrant of the total overlap, and then multiply by 4 to obtain the total overlap (Figure 6.8).

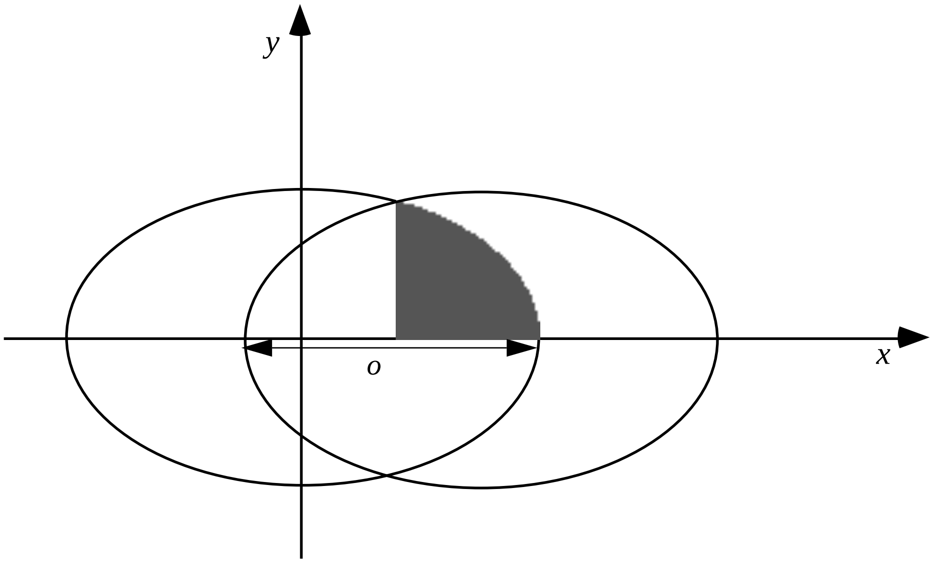


Figure 6.8 Intersecting ellipses, top view.

Without loss of generality we assume that the center of the left ellipse is the origin, that the semimajor axis is aligned with the x axis, and that the semiminor axis is aligned with the y axis. We use the equation of an ellipse centered at the origin,

$$\frac{x^2}{e^2} + \frac{y^2}{f^2} = 1,$$

to derive the function that describes the right quadrant of the ellipse boundary:

$$y = \sqrt{\left(1-\frac{x^2}{e^2}\right)\cdot f^2}.$$

The area of overlap is then

$$A = 4\cdot\int_{\left(e-\frac{o}{2}\right)}^{e}\sqrt{\left(1-\frac{x^2}{e^2}\right)\cdot f^2}\,dx = 4\cdot\frac{f}{e}\int_{\left(e-\frac{o}{2}\right)}^{e}\sqrt{(e^2-x^2)}\,dx = ""$$

$$= 2\cdot\frac{f}{e}\cdot\left(x\sqrt{e^2-x^2}+e^2\cdot\operatorname{asin}\frac{x}{e}\Big|_{\left(e-\frac{o}{2}\right)}^{e}\right).$$

After substitution for x, we get

$$A = 2\frac{f}{e}\left(\left(e\sqrt{e^2-e^2}+e^2\operatorname{asin}\frac{e}{e}\right)-\left(\left(e-\frac{o}{2}\right)\sqrt{e^2-\left(e-\frac{o}{2}\right)^2}+e^2\operatorname{asin}\frac{\left(e-\frac{o}{2}\right)}{e}\right)\right),$$

and further

$$A = 2f\left(e\left(\frac{\pi}{2}-\operatorname{asin}\left(1-\frac{o}{2e}\right)\right)-\left(e-\frac{o}{2}\right)\sqrt{\frac{o}{e}\left(1-\frac{o}{4e}\right)}\right).$$

The relative overlap (RO) between cylinder regions in consecutive section is the area of intersection divided by the area of one of the ellipses

$$RO = \frac{A}{ef\pi} = 1-\frac{2}{\pi}\left(\operatorname{asin}\left(1-\frac{o}{2e}\right)+\left(1-\frac{o}{2e}\right)\sqrt{\frac{o}{e}\left(1-\frac{o}{4e}\right)}\right). \tag{6.20}$$

After we substitute for the expressions o, e from equations 6.17 and 6.18 in Equation 6.20 (see derivation in Appendix 2), we get

$$RO = 1-\frac{2}{\pi}\left(\operatorname{asin}\left(\frac{s|\cos\alpha|}{d+w|\cos\alpha|}\right)+\frac{s|\cos\alpha|}{d+w|\cos\alpha|}\sqrt{1-\left(\frac{s|\cos\alpha|}{d+w|\cos\alpha|}\right)^2}\right). \tag{6.21}$$

Note that RO is independent of the scaling factor k. A limits analysis (see Appendix 2) tells us that RO approaches 1 when $\frac{s}{w}$ approaches 0, when α approaches $\frac{\pi}{2}$, and when d approaches infinity. RO is undefined when $s > \frac{d}{|\cos\alpha|}+w$.

6.1.3.2 Demonstration of Elliptic-Region Overlap

To demonstrate the preceding analytic derivation, I performed a simulation of overlap between sections of an infinite cylinder given a set of values for d, α, w, and s. To verify that RO for the following example is reasonable, I obtained overestimated and underestimated RO, then I showed that the analytic RO is bounded by the estimated ROs. To calculate the estimate ROs I performed the following:

1. I scan converted an ellipse (see [18], p. 90) with semiaxes derived from Equations 6.18 and 6.19 to generate a bitmap of the ellipse (Figure 6.9). For the simulation data, I verified that the area of the resulting ellipse bitmap, b_{large}, was larger than the area of the analytic ellipse. From b_{large}, I generated another bitmap, b_{small}, whose area was smaller than that of the analytic ellipse, by setting all pixels on the perimeter of the ellipse to the background value.
2. I calculated the shift in the center of mass of two hypothetical ellipses in consecutive sections with Equations 6.17 and 6.18 (that is, $h = 2e - o$), and derived the corresponding floor and ceiling values, h_{small} and h_{large}, respectively.
3. I calculated the overestimated RO by counting the number of foreground pixels in the intersection of b_{large} and b_{large} shifted by h_{small}, and dividing the result by the number of foreground pixels in b_{small}. I calculated the underestimated RO by counting the number of foreground pixels in the intersection of b_{small} and b_{small} shifted by h_{large}, and dividing the results by the number of foreground pixels in b_{large}.

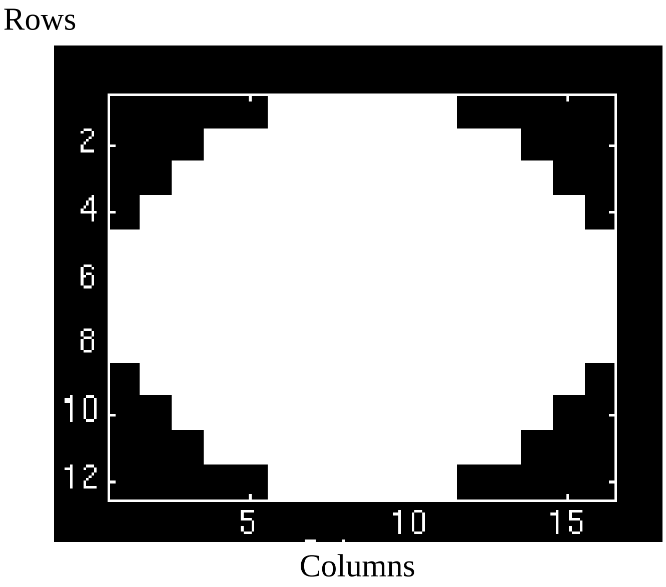


Figure 6.9 A filled scan-converted ellipse that corresponds to a cylinder section, where the cylinder has a diameter of 10 mm, the section thickness is 3 mm, and the cylinder is oriented 60° relative to the image plane. The section thickness, the orientation angle, and the roundoff error associated with scan conversion, all contribute to the length of the ellipse axes in the scan-converted image. Note that the ticks in the figure designate pixel centers.

The values that I used for the simulations were $d = 10$ mm, $\alpha = \frac{\pi}{3}$ radians, $w = 3$ mm, $s = 3$ mm, and constant window and level settings. In each simulation, I varied one of the parameters over a range of values while keeping the remainder of the parameters fixed. Figure 6.10 shows RO curves that result from these simulations. The four plots confirm our expectation that the RO for analytic ellipses would be bound by the overestimated and underestimated RO values. The plots demonstrate the sensitivity of RO to changes in d, α, w, and s, and thus provide justification for dynamic computation of the overlap criteria. The plots corroborate the results of the limits analysis (see Appendix 2)—namely, that RO approaches 1 when $\frac{s}{w}$ approaches 0, when α approaches $\frac{\pi}{2}$, and when d approaches infinity.

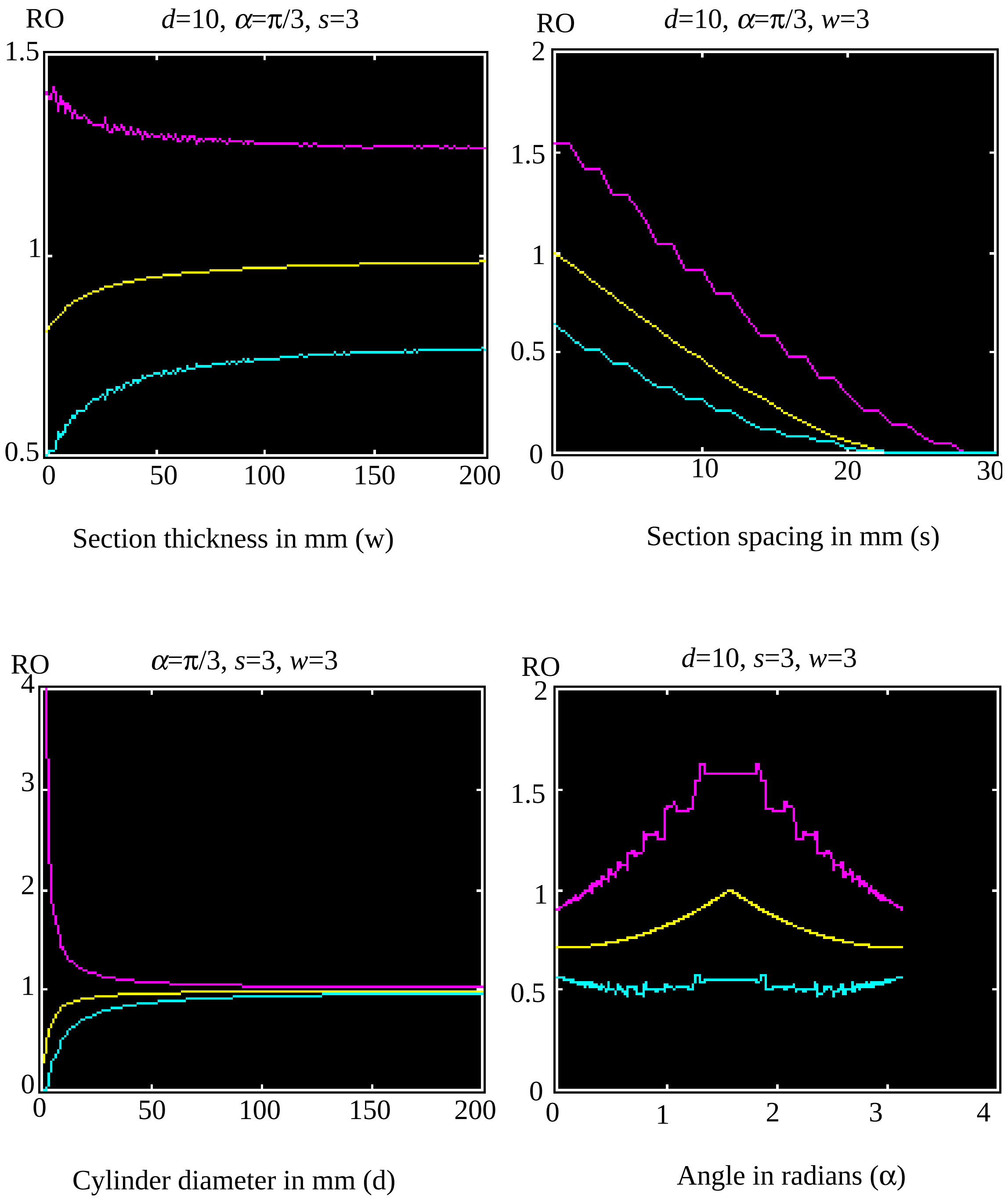


Figure 6.10 Relative-overlap (RO) curves as a function of changes in section thickness, section spacing, cylinder diameter, and the angle with respect to the imaging plane. The magenta curves represent the overestimated RO, the yellow curves represent the analytic RO, and the cyan curves represent the underestimated

> RO. The large overestimate RO values result from a large relative error that is associated with the small ellipse size.

Section 6.2 describes how IVSeg applies the region-overlap criteria to extract entire vessel structures.

6.2 Application

IVSeg's region-grouping procedure accepts as input a sequences of images for which a previous segmentation method identified salient regions in 2D, and a specification of one or several salient regions that belong to the target vessel (Figure 6.11). To extract additional regions that belong to the target vessel, the program searches throughout the volume of images with salient regions for elliptic regions that are connected to elliptic regions that the program has already identified as part of the target vessel. For each pair of elliptic regions in consecutive sections, the program fits a cylinder to the regions, and checks whether the regions exhibit overlap that is at least as large as the amount of overlap that we expect for sections of the cylinder. If the regions exhibit sufficient overlap, the program extracts their voxels.

The following sections describe the details of IVSeg's region-grouping procedure. Section 6.2.1 describes IVSeg's search for elliptic regions, which we assume are sections of a vessel that we can model adequately with cylinders. Section 6.2.2 explains how IVSeg fits a cylinder to the elliptic regions, and how the program determines what overlap between the elliptic regions it expects to observe.

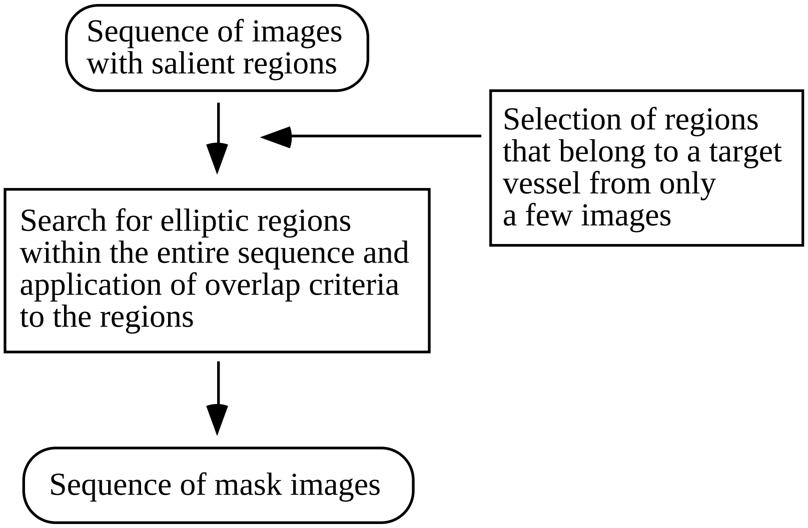

Sequence of images with salient regions
Selection of regions that belong to a target vessel from only a few images
Search for elliptic regions within the entire sequence and application of overlap criteria to the regions
Sequence of mask images

Figure 6.11 Steps in IVSeg's procedure for region grouping.

6.2.1 Search for Elliptic Regions That Might be Amenable to Application of the Overlap Criteria

To test whether a region is elliptic, IVSeg fits an ellipse to the region and compares the fit error to a predefined allowable ellipse-fit error (AEE; see expression for fit error in Section 6.1.1). Voxels that belong to a target vessel do not necessarily belong to elliptic regions in the acquisition planes. However, by modeling vessels with compositions of cylinders, we can assert that blood-vessel voxels belong to elliptic regions in at least one of three orthogonal reformations of the data (see Section 6.1.2). Based on this assertion, IVSeg searches for ellipses to which it could apply overlap criteria in three orthogonal reformations of the data, to ensure that it considers all voxels that belong to a target vessel.

IVSeg generates orthogonal reformations of the input sequence, suppresses noise, and identifies salient regions in each reformation, prior to region grouping. The program stores on a queue an object data structure for each elliptic region that is part of the target vessel. The program processes all elements in a queue until the queues is empty. It maintains three queues—one for each reformation, and processes the queues in an iterative manner until *all* queues are empty. A single iteration may not be sufficient because processing the elements of a given queue may result in addition of new elements to the other queues. Figure 6.12 shows the processing that the program performs for a region that it has placed on a queue.

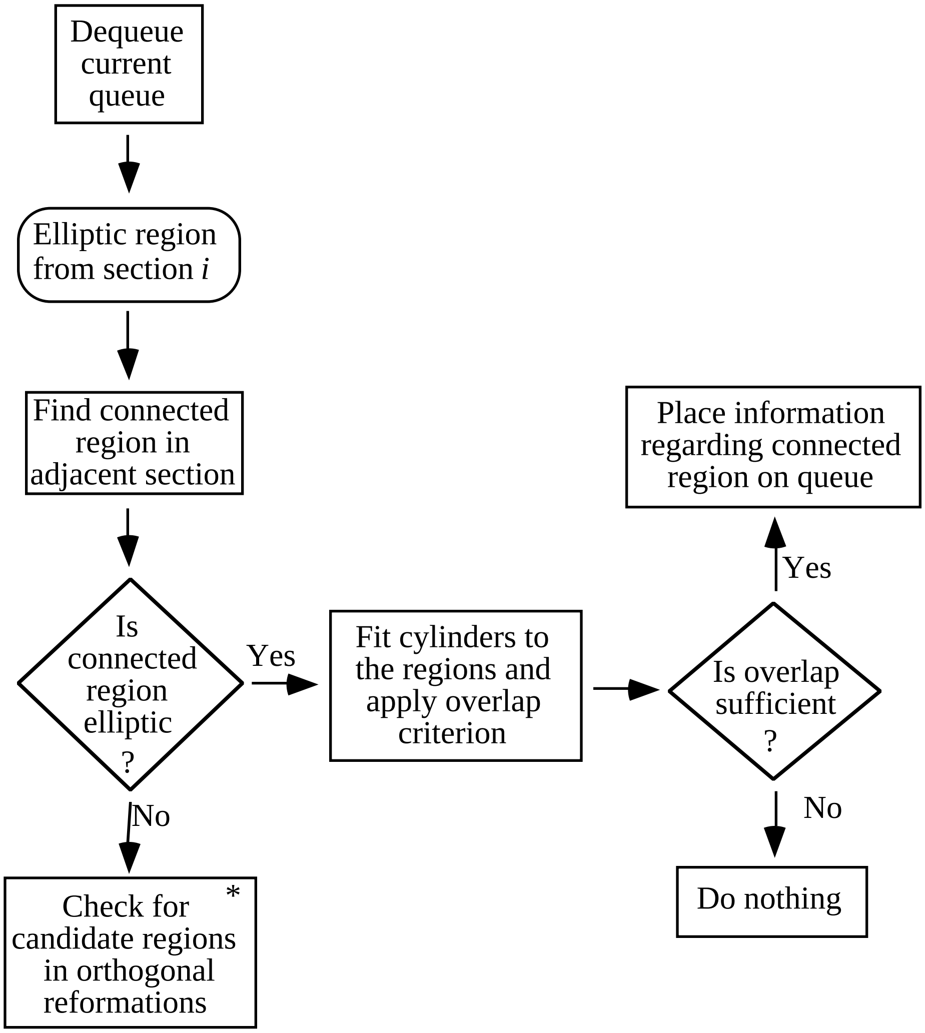

Dequeue current queue
Elliptic region from section *i*
Find connected region in adjacent section
Is connected region elliptic ?
Yes
No
Fit cylinders to the regions and apply overlap criterion
Is overlap sufficient ?
Yes
No
Place information regarding connected region on queue
Do nothing
Check for candidate regions in orthogonal reformations
*

Figure 6.12 Application of overlap criteria to test a new region. The box with the asterisk is expanded in Figure 6.13.

Given a dequeued elliptic region that belongs to a target vessel—either a region that the user selected or a region that the program has already determined to be part of the target vessel—IVSeg searches for connected regions in an adjacent section. For a connected region that is elliptic, the program attempts to fit a cylinder to the regions and to apply the overlap criterion for that cylinder (see Section 6.2.2). For a connected region that is nonelliptic, the program searches in orthogonal images with salient regions for elliptic regions that include any of the voxels of the nonelliptic region. IVSeg regards a new elliptic region as a candidate region for further processing only if, in previous iterations, the program already tagged for inclusion a sufficient ratio—the required tagged ratio (RTR)—of the elliptic region's voxels. The RTR ensures that there is substantial evidence that the new elliptic region includes target voxels (Figure 6.13).

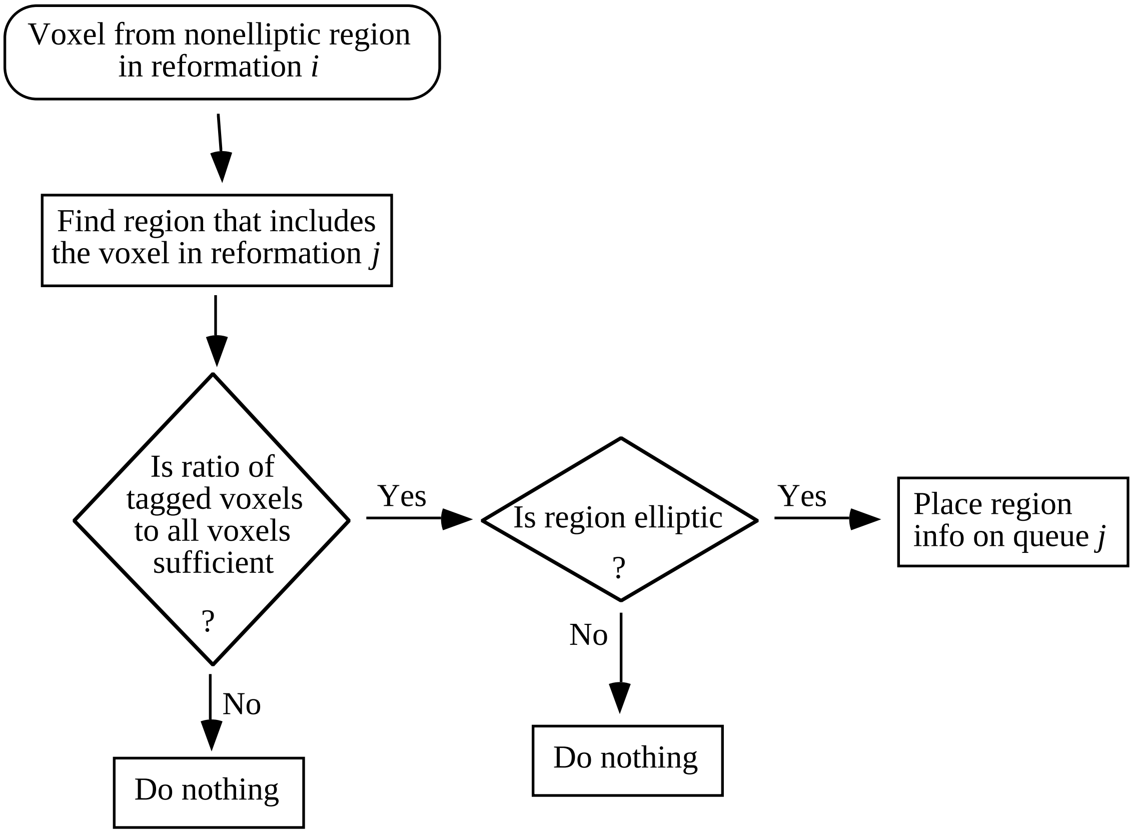

Voxel from nonelliptic region in reformation *i*
Find region that includes the voxel in reformation *j*
Is ratio of tagged voxels to all voxels sufficient ?
Yes
Is region elliptic ?
Yes
Place region info on queue *j*
No
No
Do nothing
Do nothing

Figure 6.13 Procedure for identifying candidate regions in an orthogonal reformation of the sequence.

IVSeg uses the hierarchical representation of an image with salient regions (see Section 5.2.6). It processes a salient region from its root in a breadth-first tree traversal. For a given step, if the step succeeds for a parent region, IVSeg assumes that the step succeeded for all the subtrees included within the boundary of the parent region, and does not apply the step to the subtrees. For example, if the ellipse-fit error for a parent region is smaller that AEE, IVSeg does not check whether any of its children are elliptic, and continues to process all voxels within the parent region. Similarly, if the ratio of tagged voxels to all voxels in a parent region is greater than the RTR, IVSeg does not check the ratio of tagged voxels for any of that region's children. Note that, for a given parent region, IVSeg checks the ratio of tagged voxels for the difference set of the parent and its children, or for voxels that the parent includes but the children do not. This way of assessing the ratio of tagged voxels reduces the chance of getting a count that is greater than the RTR for a region that is not part of the target vessel.

6.2.2 Dynamic Estimation of Expected Overlap for Elliptic Regions

IVSeg derives and applies overlap criteria for only those elliptic regions that are in consecutive sections. IVSeg distinguishes three possible mutually exclusive situations:

1. The axes of the elliptic regions have the same length and orientation; we assume that the regions originated from a noncurving vessel segment, which we model with a single cylinder (Figure 6.14a).
2. The major axes have different lengths but the minor axes have the same length and orientation; we assume that the regions originated from a vessel that is curving in the through-plane direction. We model the vessel with a composition of cylinders (Figure 6.14b).
3. Neither situation 1 nor situation 2 holds. In this case, we conclude that the regions do not belong to the same vessel.

For situation 3, IVSeg does nothing. For both situations 1 and 2, IVSeg fits a cylinder to the smaller ellipse for the following reasons. For situation 1, since the ellipses have similar sizes and orientations, we expect that the cylinder fit would yield the same result, independent of which ellipse we use. For situation 2, each of the regions assumes a different orientation of the vessel axis with respect to the image plane. By fitting a cylinder to the ellipse with the smaller major axis, IVSeg assumes a larger angle with respect to the image plane, and thus employs a greater RO requirement (see Figure 6.10). IVSeg assumes situations 1 or 2 if the ratio of the length of the minor axes is within a predefined allowable axis ratio range (AARR), and if the absolute orientation difference of the minor axes is smaller than a predefined allowable orientation angle difference (AOAD). Otherwise, it assumes situation 3.

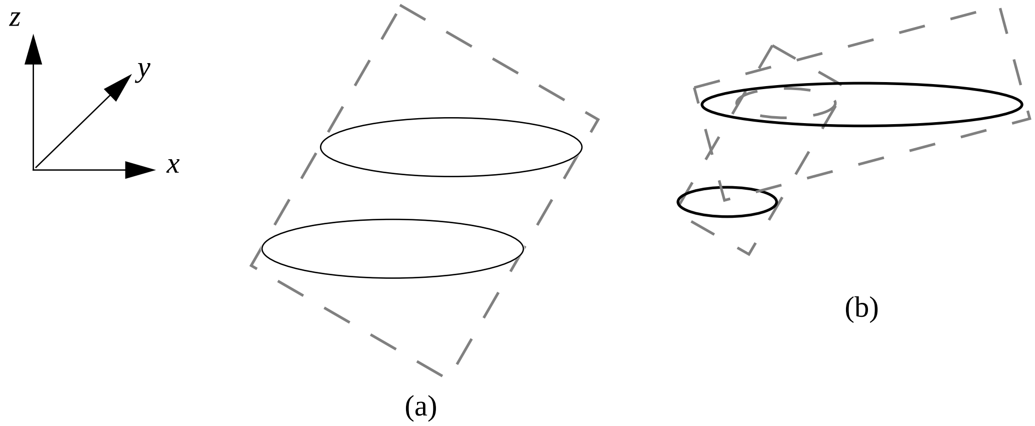


Figure 6.14 Cylinder fit to elliptic regions in consecutive sections. We obtained the view by cutting through the cylinders along the *xz* plane. The ellipses are in the *xy* plane. (a) Fit of a single cylinder to two elliptic regions for which both axes have similar length and orientation. (b) Fit of two cylinders to two elliptic regions whose minor axes have a similar length and orientation.

When IVSeg fits a cylinder to an ellipse, it assumes that the ellipse in question is a section through a right cylinder with a circular base that has been rotated from an upright position about an axis that intersects the cylinder's principal axis and is orthogonal to the principal axis. For an ellipse with a major axis j and a minor axis i, the cylinder diameter is i, and the cylinder orientation relative to the image plane—α—satisfies the equation $\frac{i}{j} = \sin\alpha$ (Figure 6.15). Therefore, the orientation of the cylinder is $\operatorname{asin}\frac{i}{j}$.

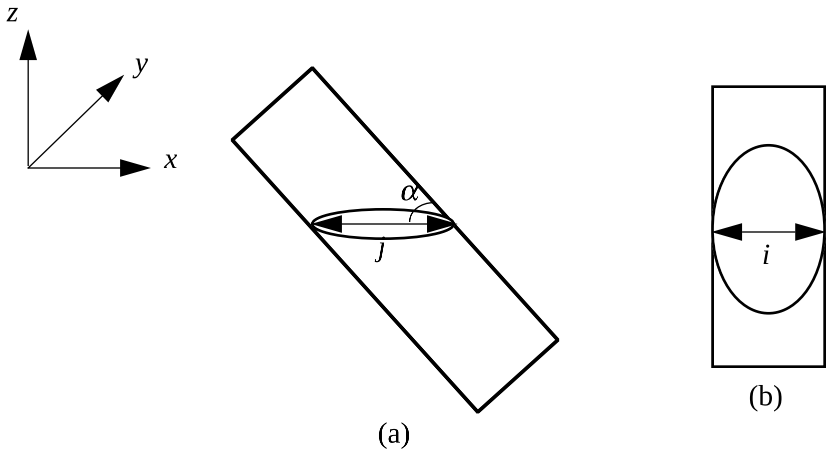


Figure 6.15 A slanted right cylinder and its elliptic cross section. (a) The cylinder axis is in the *xz* plane. We obtain the view by cutting through the cylinder along the *xz* plane. The ellipse is in the *xy* plane. (b) A view from the direction of the slant.

IVSeg uses the diameter, orientation, and the scanning parameters (section thickness and spacing) to calculate the expected RO between the intersections of consecutive image planes with the cylinder, according to Equation 6.21. IVSeg compares the expected RO to the actual RO between the regions; it computes the latter by dividing the number of overlapping voxels by the number of voxels in the smaller region. If the actual RO is greater than or equal to the expected RO, IVSeg determines that the regions belong to the same object; otherwise, IVSeg determines that the regions belong to different objects.

6.3 Sensitivity Analysis

In this section, I describe an experiment that I conducted to determine the sensitivity of the results of IVSeg's region-grouping procedure to 5-percent changes in parameter settings. I examined these parameters:

1. RTR—the ratio of the voxels within a region that IVSeg already identified as part of the target structure to the total number of voxels within the region. IVSeg regards elliptic regions in orthogonal reformations of the data as candidate for application of the overlap criteria only if the ratio of tagged voxels to all voxel is greater than RTR (see Section 6.2.1).
2. AEE—the allowable error between a region's boundary the ellipse that the IVSeg fits to the boundary. AEE is the average coordinate difference between corresponding points on the spline and the ellipse that the program fits to the boundary. IVSeg regards regions with an error smaller than AEE as elliptic, and regions with larger errors as nonelliptic (see Section 6.2.1).
3. AARR—the ratio between minor axes of ellipses in consecutive sections. IVSeg verifies the amount of overlap for only those ellipses whose minor-axis ratio is within AARR (see Section 6.2.2).
4. AOAD—the maximum orientation difference between minor axes of ellipses in consecutive sections. IVSeg verifies the amount of overlap for only those ellipses whose minor-axes orientation differs by less than AOAD (see Section 6.2.2).

Through experimentation with three CTA cases, I determined settings for the parameters that produced good results according to my subjective judgment. The settings were AEE=3.75 pixels; AARR=0.80-1.2; AOAD=0.6 radians; RTR=0.4. My goal was to determine whether 5-percent changes in these parameters produced results that were significantly different from the results that I obtained with the original parameter settings.

6.3.1 Study Design for Sensitivity Analysis

I selected randomly 16 cases from a 1-year-old archive of abdominal-pelvic CTA cases. The number of sections in the cases ranged from 128 to 198. For each case, I reformatted the data to obtain sequences of sections in the coronal and sagittal orientations. Then, I used IVSeg to identify salient regions in all three orientations. I assigned randomly to each case one of these structures: aorta, superior mesenteric artery, right renal artery, left renal artery, right iliac artery, left iliac artery. From the sections of each structure, I picked randomly the section to click within the structure. Then, I selected the structure with one mouse click, and activated IVSeg's region-grouping procedure. Table 6.1 describes the structures that I used for the sensitivity analysis.

Table 6.1 Structures used in sensitivity analysis.

Case	Structure	No. of sections	Ordinal section clicked (actual section number)
3	abdominal aorta	79	58 (58)
6	left renal artery	20	11 (30)
10	superior mesenteric artery	24	3 (14)
11	left renal artery	14	5 (26)
15	right iliac artery	110	105 (182)
18	right renal artery	15	10 (25)
20	abdominal aorta	69	24 (24)
22	left iliac artery	67	43 (134)
26	right iliac artery	75	18 (116)
31	superior mesenteric artery	22	9 (7)
32	left renal artery	9	6 (59)
33	aorta	52	14 (114)
39	left iliac artery	67	30 (121)
41	right iliac artery	91	42 (104)
42	superior mesenteric artery	62	55 (73)
50	right renal artery	8	2 (74)

The ordinal numbers reflect a count from the first section that represents the structure. Note that the number of instances that I selected for a given structure was roughly proportional to the number of sections that the structure spanned.

For each case, I activated the region-grouping procedure for the original parameter settings; then, I activated the procedure for 16 combinations of the four parameters at two levels each, where, for each parameter, each level constituted a 5-percent change, either above or below the original setting. I considered the results that IVSeg produced for the original parameter setting as the **baseline sequence**, and those for the 16 parameter-level combinations as the **treatment sequences.** I compared each treatment sequence to the baseline sequence. I measured the degree of mismatch for every pair of corresponding mask images, and I performed an analysis of variance for a four-way classification to determine whether any of the four parameters or parameter interactions explained differences in the mismatches between corresponding images in the treatment sequences and images in the baseline sequence. The null hypothesis was that *neither any one of the parameters (RTR, AEE, AARR, AOAD), nor any combination of these parameters, explained the differences in corresponding regions between images in the baseline sequence and images in the treatment sequences.*

6.3.2 Sensitivity-Analysis Measures

I looked at three measures for each pair of sequences that I compared. Two measures—the mean and standard deviation of NVM (see Section 5.3.1.2) for corresponding images in the baseline and treatment sequences—reflected the extent to which IVSeg's results in individual sections were similar for the two sequences. The third measure—the number of salient regions in one of the sequences that did not have an overlapping counterpart in the other sequence—reflected the extent to which IVSeg's results included different substructures for the two sequences. The third measure also reflected the number of mouse clicks that the user might have to make in the worst-case scenario to include the substructures that were not present in one of the sequences.

6.3.3 Sensitivity-Analysis Results

The results of the analysis of variance showed that there was insufficient evidence to reject the null hypothesis with respect to the three measures at $p < 0.05$, for all parameter combinations and for AEE, AARR, and AOAD alone, and that there was sufficient evidence to reject the null hypothesis for RTR alone. Table 6.2 shows the ***F* value** for each parameter or parameter combination, and the ***p* value**, or the observed significance level [78] (see also Section 5.3.1.3). The table shows the extent to which the parameter or parameter–level combinations explain the differences observed between the groups with respect to each dependent variable. From the table, we see that, if we reject the null hypothesis for each parameter or parameter combination, then we risk making errors at the rates specified in the third column. Because we do not have sufficient confidence to reject the null hypothesis with respect to the three measures, for all parameter combinations and for AEE, AARR, and AOAD, we accept it. We accept the null hypothesis with respect to the mean NVM for RTR. We reject the null hypothesis with respect to the standard deviation of NVM and the number of nonoverlapping regions for RTR.

Table 6.2 *F* values for parameters and parameter interactions.

Parameter (or interaction)	*F* value—mean NVM	Pr > *F*—mean NVM	*F* value—standard deviation of NVM	Pr > *F*—standard deviation of NVM	*F* value—number of non-overlapping regions	Pr > *F*—number of non-overlapping regions
AEE	0.78	0.3791	3.75	0.0540	0.15	0.7013
RTR	1.36	0.2448	23.21	0.0001	20.06	0.0001
AEE*RTR	1.57	0.2113	2.90	0.0900	1.49	0.2241
AARR	0.00	0.9872	0.00	0.9876	0.25	0.6192
AEE*AARR	0.05	0.8319	0.01	0.9368	0.02	0.8972
AARR*RTR	0.01	0.9420	0.00	0.9715	0.02	0.8879
AEE*AARR*RTR	0.04	0.8407	0.01	0.9127	0.02	0.9003
AOAD	0.00	0.9967	0.00	0.9976	0.08	0.7781
AEE*AOAD	0.00	0.9747	0.00	0.9931	0.04	0.8326
RTR*AOAD	0.00	0.9723	0.00	0.9921	0.05	0.8295
AEE*RTR*AOAD	0.00	0.9892	0.00	0.9976	0.08	0.7751
AARR*AOAD	0.00	0.9893	0.00	0.9985	0.09	0.7631
AEE*AARR*AOAD	0.00	0.9916	0.00	0.9970	0.07	0.7931
AARR*RTR*AOAD	0.00	0.9938	0.00	0.9972	0.04	0.8448
AEE*AARR*RTR*AOAD	0.00	0.9771	0.00	0.9926	0.06	0.8143

Parameter interactions are designated by ’*’. Pr>*F* denotes the area under the curve of the *F* distribution that is to the right of the *F* value; Pr>*F* is the probability of rejecting the null hypothesis when it is true. The unit of analysis is a case. The dependent variables are mean NVM, standard deviation of NVM, and number of regions with no overlapping counterpart.

To draw valid conclusions from the analysis of variance test about the sensitivity of IVSeg’s results to changes in program parameters, I needed to verify that the power $(1 - \beta)$ of the test was sufficient—that the test would have detected considerable ratios of the between-group variation and the within-group variation, if such ratios emerged from the data. The power calculation (see [79]) considered four factors (RTR, AEE, AARR and AOAD) at two levels each, and 16 observations for each factor-level combination. I assumed that the variance in the underlying populations was close to zero, because NVM for baseline images relative to themselves was zero, and because the number of regions with no overlapping counterparts was zero. The power calculations showed that, at $\alpha = 0.05$, $1 - \beta$ was greater than 80 percent for all three dependent variables. I concluded that the analysis of variance test had sufficient power to detect any existing significant effects due to 5-percent changes in IVSeg parameters.

6.3.4 Sensitivity-Analysis Conclusions

The results of the sensitivity analysis showed that IVSeg's region-grouping procedure was not sensitive to 5-percent changes in the parameters AEE, AARR, and AOAD, and that it was sensitive to RTR according to two out of the three measures. These findings suggest that, for CTA data, the initial settings of AEE, AARR, AOAD are likely to be adequate, and that a fine-tuning of RTR might be necessary. I noted that a 5-percent change in RTR affected the standard deviation of NVM and the number of nonoverlapping regions in opposite directions: an increase in RTR resulted in a smaller standard deviation of NVM, but in a greater number of regions with no overlapping counterparts. For the remaining experiments (see Section 7.2), I used the value of RTR that was a 5-percent increase over the centerpoint value—0.42—to target a smaller standard deviation of NVM. I assumed that missing regions could be included with additional mouse clicks.

6.4 Summary

In this chapter, I described the theory and methods that underly IVSeg's region-grouping procedure. I presented the region-grouping procedure as a search for elliptic regions that are related to one another. I explained how we can test whether a region is elliptic by approximating the contour with an ellipse. I claimed that we are guaranteed to find elliptic sections of a vessel structure in at least one of three orthogonal reformations of a volume.

To support this claim, I proved that right cylinders with circular bases that are intersected by arbitrary orthogonal planes have elliptic sections. I suggested that we can use overlap criteria to test whether elliptic regions in consecutive sections belong to the same structure, and I derived a formal expression for the overlap of consecutive sections through a right cylinder. Then, I described how the methods are implemented within IVSeg. I explained how IVSeg searches for elliptic regions within the entire volume, and how it dynamically derives overlap criteria by estimating cylinder parameters from salient regions in consecutive sections. Finally, I showed through a sensitivity analysis that the initial settings for three of the four parameters of the region-grouping procedure prevailed system robustness.

In Chapter 7, I describe an evaluation study that I conducted to compare editing of CTA data with IVSeg to editing via manual tracing.

Chapter 7

Comparison of Manual Editing to Editing with IVSeg

IVSeg is a unified system in which I implemented the methods that I described in Chapters 5 and 6. IVSeg allows users to extract vessels from medical-image volumes with little effort. To compare editing with IVSeg to traditional editing via manual tracing, I conducted a formal evaluation study. The goal of the study was to determine whether a user could edit CTA volumes with IVSeg and obtain results that were comparable to experts' manual tracing of these volumes, and whether editing with IVSeg saved time. I wished to identify the source of inadequate performance if it occurred, and therefore, where applicable, I examined the performance of IVSeg's procedure for identification of salient regions and the performance of IVSeg's region-grouping procedure, which used as input the results of the first procedure, separately.

Section 7.1 presents the user interaction with IVSeg. Section 7.2 describes the evaluation methods I used. Section 7.3 reports the results of the study. Section 7.4 comprises a discussion and conclusions. Section 7.5 is the chapter summary.

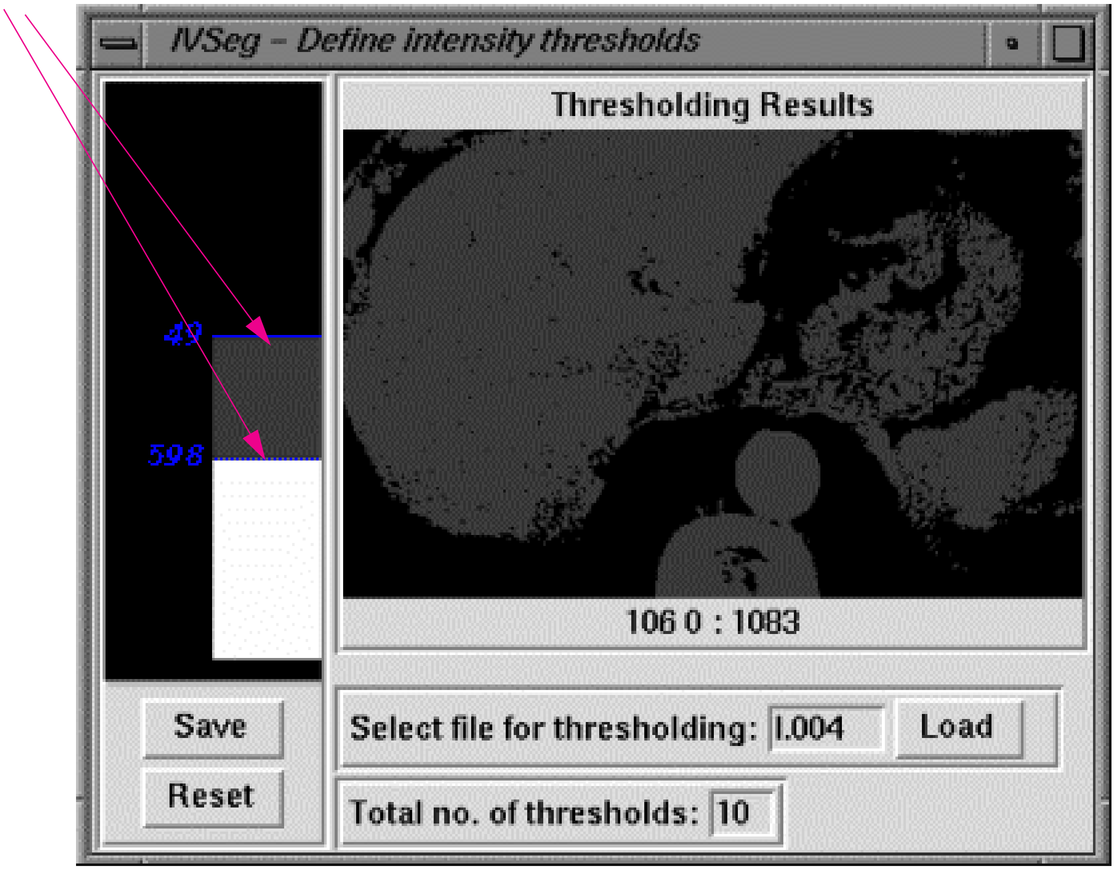
7.1 IVSeg: A System for Editing Medical Image Volumes

IVSeg allows the user to process a case in two stages. First, the user activates the procedure for identification of salient regions, which identifies in two dimensions major structures that are potentially of interest to her. Second, she indicates samples of the target structure, and then activates the region-grouping procedure to collect relevant salient regions from the entire sequence. In Sections 7.1.1 and 7.1.2, I describe the two procedures.

7.1.1 Step 1: Identification of Salient Regions

To activate the procedure for identification of salient regions, the user must specify three input parameters: the lower and upper intensities that guarantee inclusion of the target structure, and the number of thresholds that IVSeg should use for generating isolabel-contour maps (see Section 5.1.1). Figure 7.1 shows IVSeg's user interface for the user to enter the three parameters.

Intensity
specification bars



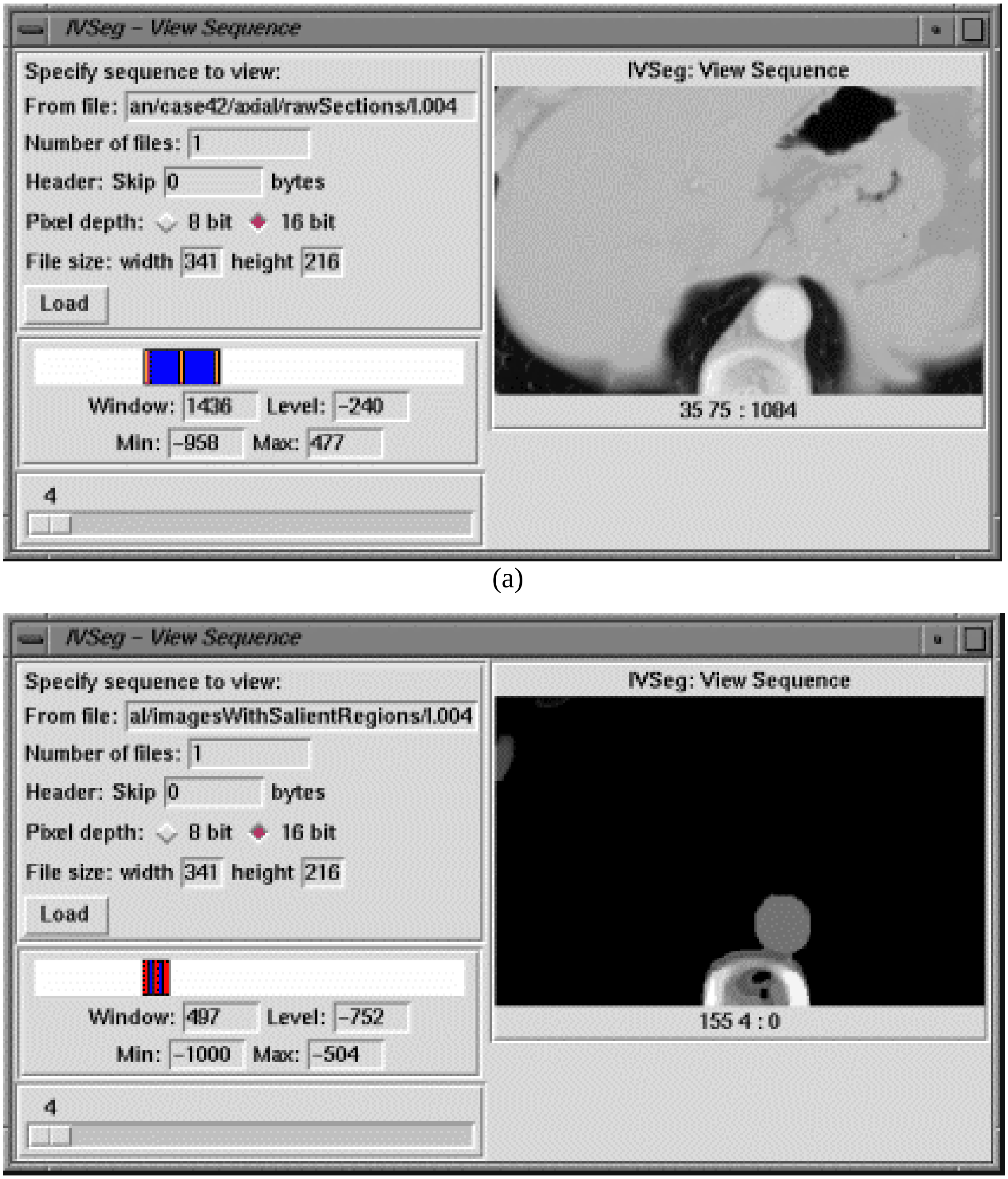
Select file for thresholding: I.004

Total no. of thresholds: 10

Figure 7.1 User interface for the user to specify threshold-partitioning parameters. She specifies the range of intensities by moving the intensity-specification bars on the left side of the window. As one of the bars moves, the image on the right reflects the results of thresholding at the specified intensities; thus, it provides feedback to the user. The numbers immediately below the image specify the image coordinates and intensity of the pixel at which the mouse points.

When the user selects the *edit* option, IVseg activates the procedure for identification of salient regions three times, once for the axial acquisition plane, and once for each of two orthogonal reformations of the data—the coronal and the sagittal reformations (see Chapter 5). For each orientation, IVSeg generates a sequence of images with salient regions.

Figure 7.2 shows one example of a raw image and the corresponding image with salient regions within IVSeg's sequence viewer.



IVSeg – View Sequence

Specify sequence to view:

From file: an/case42/axial/rawSections/I.004

Number of files: 1

Header: Skip 0 bytes

Pixel depth: ◇ 8 bit ◆ 16 bit

File size: width 341 height 216

Window: 1436 Level: -240

Min: -958 Max: 477

4

IVSeg: View Sequence

35 75 : 1084

(a)

IVSeg – View Sequence

Specify sequence to view:

From file: al/imagesWithSalientRegions/I.004

Number of files: 1

Header: Skip 0 bytes

Pixel depth: ◇ 8 bit ◆ 16 bit

File size: width 341 height 216

Window: 497 Level: -752

Min: -1000 Max: -504

4

IVSeg: View Sequence

155 4 : 0

(b)

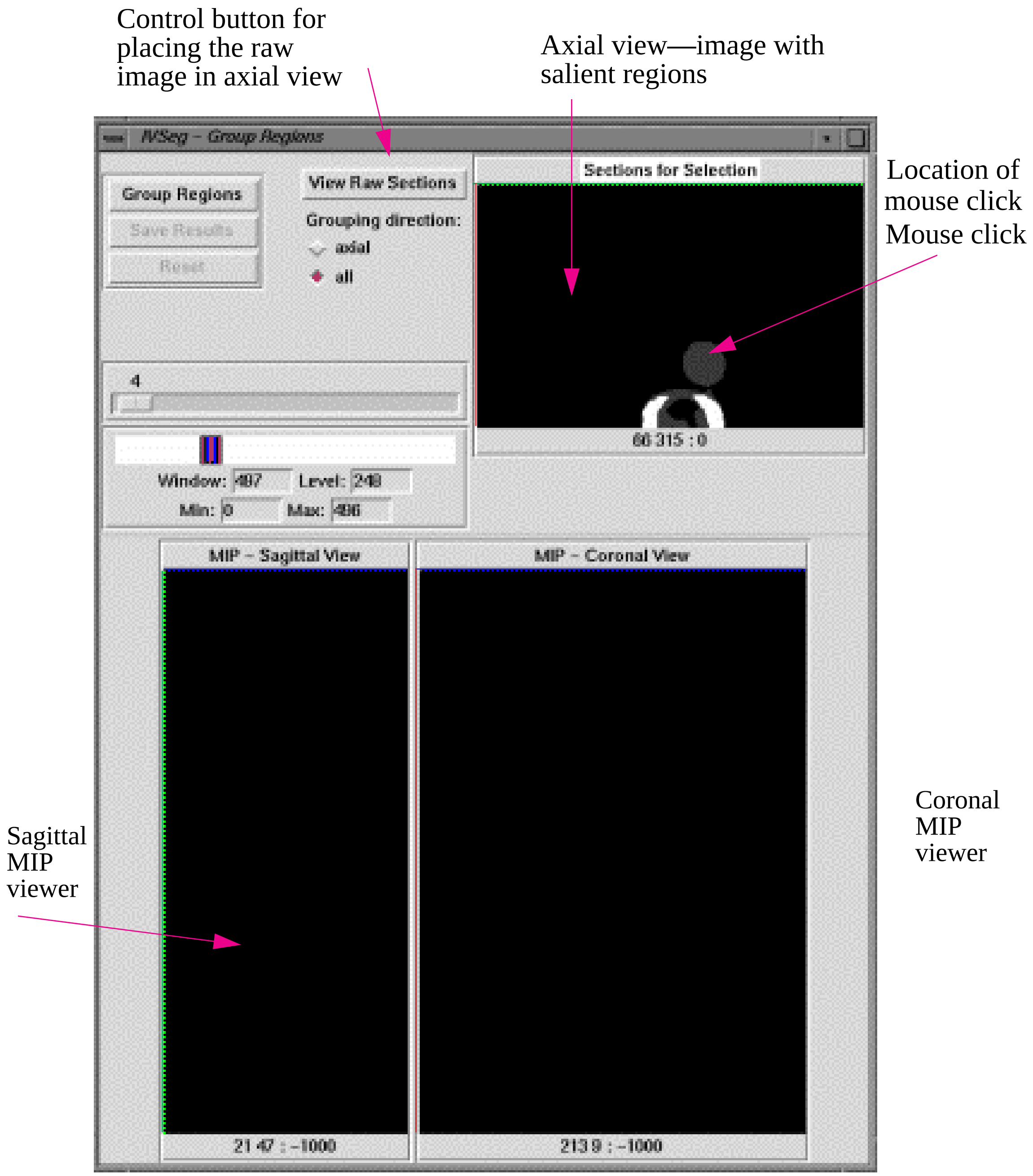
Figure 7.2 Images within IVSeg's sequence viewer. (a) A raw section. (b) The corresponding image with salient regions. The controls on the left side of the

sequence viewer allow the user to select an image and to adjust the window and level settings.

7.1.2 Step 2: Region Grouping

After IVSeg completes step 1, the user selects a target structure by clicking with the mouse on one or more regions throughout the sequence of raw sections, or throughout the sequence of images with salient regions (Figure 7.3). Then, she activates the region-grouping procedure (see Chapter 6). As IVSeg includes regions, it provides feedback to the user by displaying the included voxels within coronal and sagittal MIPs of the accumulated volume. After the first grouping cycle has terminated, the user can inspect the MIP views. If there are missing voxels on the MIP views, the user can locate them on the raw sections, or on the images with salient regions, by placing the crosshairs on the areas where voxels are missing in the sagittal and coronal views (Figure 7.4).

The user can determine the cause for missing structures in the following way. After she identifies the missing voxels on an axial raw section, she can view the image with salient regions that corresponds to that section. The presence of a salient region that encloses the missing voxels indicates that IVSeg failed in step 2. The user can include the missing voxels by clicking within the region and reactivating the region-grouping procedure for inclusion of any related regions that IVSeg did not include. The user typically edits a case incrementally, by performing several cycles of region selection with a mouse and activation of the region-grouping procedure. When the user is happy with the voxel-inclusion results as they are reflected within the MIP viewers, she can save them.



Grouping direction:

○ axial

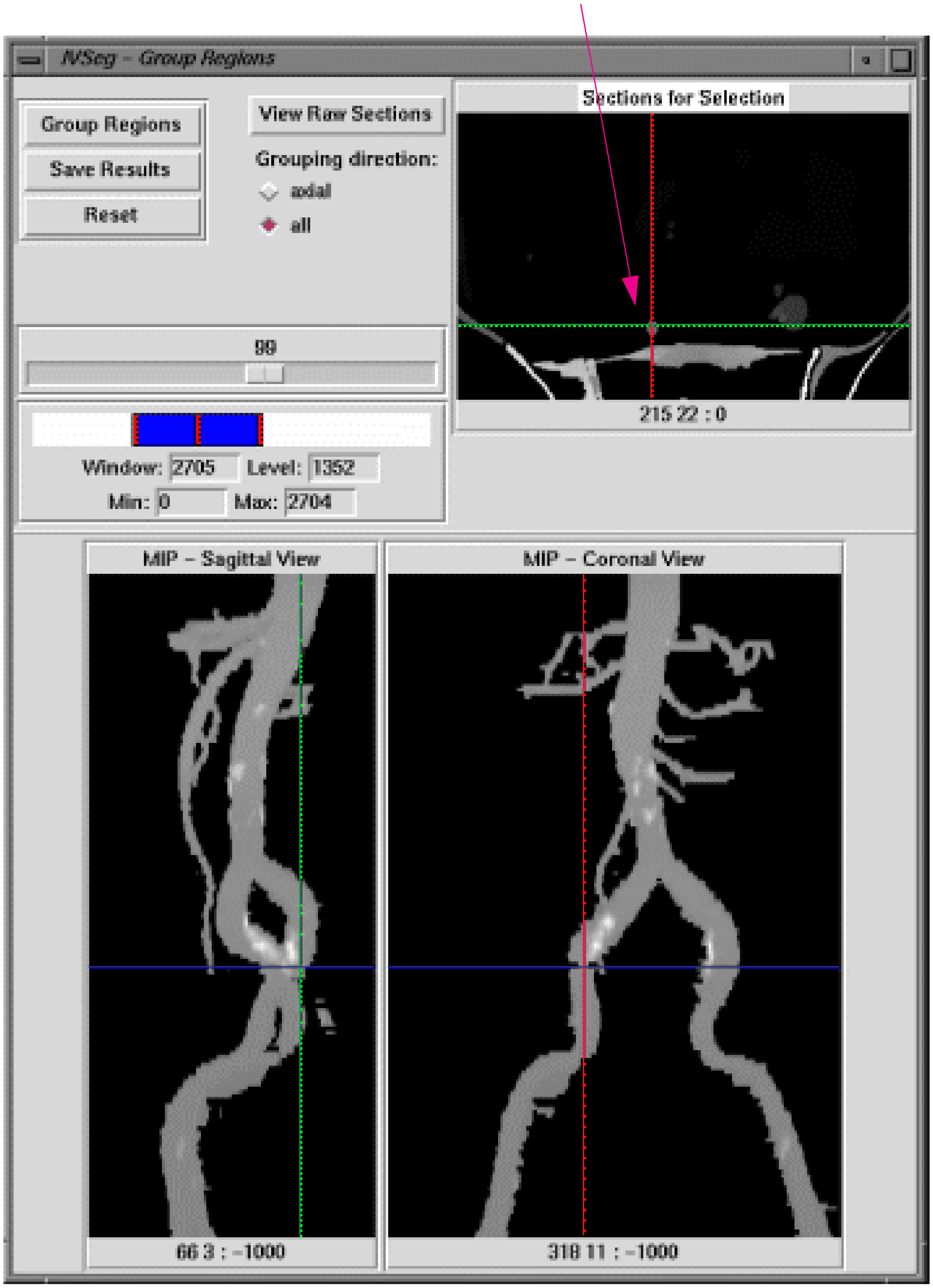
● all

Window: 497 Level: 248

Min: 0 Max: 496

Figure 7.3 User interface for the user to invoke the region-grouping procedure prior to activating it.

Point of missing voxels



Grouping direction:
○ axial
● all

Window: 2705 Level: 1352
Min: 0 Max: 2704

Figure 7.4 User interface for the region-grouping procedure following procedure activation. The user localizes the area of missing voxels by moving the crosshairs within the MIP viewers to this area. In response, IVSeg displays the appropriate image within the axial view and positions the crosshairs at the target point.

7.2 Evaluation Methods

The goal of the evaluation study was to compare editing with IVSeg to manual editing. The hypothesis of the study, which constituted part of the research hypothesis for my thesis work, was that IVSeg allows radiology experts to edit CTA volumes robustly in less than 10 seconds per section on average, with results comparable to those produced by experts using conventional editing tools. To verify the hypothesis, I conducted three experiments. In the first experiment, I compared the results of editing with IVSeg, to those of manual editing. In the second experiment, I compared editing time for IVSeg to that of manual editing. In the third experiment, I compared the differences between IVSeg's editing and manual-editing results to differences between manual- editing results that different experts generated.

Ideally, I could measure the accuracy of IVSeg for patient data and compare it to the accuracy of manual editing. However, measuring accuracy is impossible because we cannot obtain gold-standard measurements of patients' vasculature. The third experiment allowed me to compare results of IVSeg to results of manual editing without measuring accuracy. Differences between IVSeg's editing results and those of manual editing that were within the range of inter-expert variability for manual editing would indicate that the results of the two methods were comparable.

I recruited five radiology experts who had experience in editing medical image volumes using conventional editing tools. The editing tools—intensity thresholding, connectivity tracking, and manual tracing—were available within the General Electric, Inc., Advantage Windows Workstation. In the following discussion, I refer to the use of these editing tools as **manual editing**. One of the experts trained me, and then I served as the sixth expert. I selected randomly 16 abdomen–pelvis cases acquired at Stanford University Hospital in 1997. I asked the experts to edit these cases both via manual editing and with IVSeg.

The following sections explain the data collection and analysis for the three experiments. Section 7.2.1 describes a comparison of the results for the two methods. Sections 7.2.2 outlines a comparison of the user time that the two methods required. Section 7.2.3

describes a comparison of the differences between results of IVSeg and those of manual editing to differences of results that different experts generated via manual editing.

7.2.1 Experiment 1: Matching of IVSeg-Generated Regions and Manually Edited Regions

There were two goals for the first experiment: to determine whether the results of editing CTA cases with IVSeg were free from significant leaks into bone, and to test whether experts were able to use IVSeg to produce results that were similar in voxel inclusion to results of manual editing.

7.2.1.1 Data Collection for Experiment 1

Five experts edited the 16 cases manually. I assigned the cases to the experts randomly such that all cases would be edited once. Each expert edited one case per week. I instructed the experts to include only high-contrast arteries and calcifications, and to exclude all other structures. Because thrombi did not have characteristics that match IVSeg's prior assumptions, I instructed the experts to leave out thrombi as well. For each editing session, I recorded the editing time, because I would need those data for experiment 2.

Four experts edited the 16 CTA cases with IVSeg. The experts received the same instructions as those that I provided for the manual-editing sessions. IVSeg used program-parameter values that constituted the center point for the sensitivity-analysis experiments (see Sections 5.3.1 and 6.3), except for one parameter, the RTR. For RTR, IVSeg used the 5-percent–increment value, because the sensitivity analysis showed that the program's results were less sensitive to a 5-percent increase than they were to a 5-percent reduction.

Because computation time with the current implementation of IVSeg prohibited a truly interactive activation of the system, I needed to simplify the logistics of the editing sessions with the program. I limited the number of editing iterations to three. In the first iteration, the experts selected two regions out of each of six structures: aorta, celiac artery, right renal artery, left renal artery, right iliac artery, and left iliac artery. In the second iteration, the experts selected artery regions that were not included yet; they identified missing structures by manipulating the crosshairs on the MIP views, and clicked on up to 15 additional regions. In the final iteration, the experts selected artery regions that were not included yet by manipulating the crosshairs on the MIP views and by clicking on as many regions as necessary to obtain a completely edited volume. For each iteration, I recorded the image coordinates of mouse clicks and the duration of user input, including time allocated to browse through the sequence. For several cases, it was apparent from the MIP views that there were still regions missing after the third iteration. For these cases, after the experts had finished, I continued to select regions and to activate IVSeg, and I recorded the additional points that I selected. In addition, I recorded the total user-interaction time for editing with IVSeg; the user-interaction time was necessary for experiment 2.

For three cases, it was clear from the MIP views that the region-grouping procedure leaked into bone structures that obscured major vessels. I designated these cases as ones for which IVSeg failed (see discussion in Section 7.3.1), and I interrupted the program. I excluded these cases from further analysis. I also excluded from analysis the results of three sessions for which there were technical difficulties: For two manual-editing sessions, the saved data was corrupted, and for one IVSeg session, a change of FOV disrupted object connectivity. Thus, the total number of cases for which I compared the editing methods was 10. Table 7.1 provides information about the cases that the experts edited for the evaluation.

Table 7.1 Cases that the experts edited for the evaluation study.

Case	No. of sections	Special characteristics	Manual edit expert	IVSeg edit expert	IVSeg edit—total no. of clicks	IVSeg edit—clicks beyond three cycles
1	189	shape distorting thrombi	E	E	25	0
3	181	shape distorting thrombi	E	E	54	26
6	162	shape distorting thrombi, dilation	B	F	39	7
10	158	dilation	B	F	leak	NA
11	159		D	F	23	0
15	187	stent	A	A	37	0
18	157	shape distorting thrombi, dilation	C	C	81	16
20	156	streaks	E	E	41	0
22	158	shape distorting thrombi	E	E	leak	NA
26	173		C	C	55	9
31	175	dilation	C	C	leak	NA
32	198	dissection	D	F	124	86
39	158	shape distorting thrombi, dilation	B*	F	53	7
41	152	change of FOV	B	F*	34	5
42	167		A*	F	69	16
50	178		D	F	34	0

The case numbers are nonsequential because I did not use all the cases which I had originally selected from the archives of Stanford University Hospital. For three cases, editing with IVSeg resulted in a severe leak; consequently I interrupted the editing session for these cases. I, designated here as expert F, continued the IVSeg editing sessions beyond three cycles. An asterisk next to an expert's letter indicates that there was a technical problem in the editing session.

7.2.1.2 Data Analysis for Experiment 1

In the first part of experiment 1, I determined whether editing with IVSeg resulted in leaks into bone. For cases where IVSeg did not fail editing by leaking severely into bone, I examined the edited sections one by one. I noted whether there were any leaks into bone;

where there were leaks, I computed the ratio of the number of voxels within each leak relative to the number of voxels that experts included in the manually edited volume.

In the second part of this experiment, I determined whether editing results for the two methods were similar by comparing corresponding cases that experts edited using the two methods. Before comparing the results, I adjusted the edited cases for viewing, as experts do for clinical cases. For each case, I dilated each included region three times with a structuring element of size 3×3. After dilation, I had an expert determine a good window-level setting for the manually edited volume, then applied these settings to the two edited volumes. To determine the match of editing results at the volume level, I computed the NVM (see Section 5.3.1.2) for the entire volume, where I used the manually edited sequence as the baseline. To determine the similarity at the region level, I computed the NVM for corresponding regions. Because several manually edited cases included small spurious regions, I restricted the comparison to pairs of regions that included 50 or more voxels in area per region.

7.2.2 Experiment 2: Comparison of User Time Required for Editing

The goal of the second experiment was to determine the user time required for editing CTA cases with IVSeg, and to compare it to the user time required for manual editing of the same cases. Recall that I recorded user editing times for editing sessions with the two methods (see Section 7.2.1.1). I calculated user editing times per section, and I used a paired *t*-test to compare the times for corresponding sections of the 10 cases.

7.2.3 Experiment 3: Comparison of Difference Between IVSeg and Manual Results to Inter-Expert Variability for Manual Editing

The goal of the third experiment was to test whether IVSeg's results compared to those of manual editing as well as manual-editing results compared to one another. There were two parts to this experiment. First, I compared IVSeg's salient regions to corresponding manually edited regions. Because salient regions constituted IVSeg's intermediate results, from which the program would ultimately select the final results, it was important to verify that the intermediate results were similar in voxel inclusion to results of manual editing. Second, I compared regions that IVSeg ultimately selected to corresponding manually edited regions.

7.2.3.1 Data Collection for Experiment 3

To determine expert variability for manual editing, I selected randomly a set of 75 sections from the original 16 CTA cases. Experts A, B, C, D, and E independently edited each of the 75 sections. To reduce the effect of the order of editing on the results, I divided the sections into six subsets and asked the experts to edit the sections by subsets in random order. To reduce the effect of the timing and the order of subset editing for a given day on the results, the experts edited at most one subset per day.

Because I was interested in the experts' delineation of boundaries of regions rather than their judgment of which regions should be included, I generated master edits that indicated which structures to include. For each section, I used as a template the first edit that I obtained from any one of the experts, and I marked on a print of the raw section a circle around each region that the template included. I deliberately inserted a wide margin around the region, so that the experts would not be influenced by my delineation of the boundary.

To obtain IVSeg's data, I used as templates edited sections that I selected from the 16 volumes that experts edited manually, and that corresponded to the 75 sample sections. I wrote a routine that selected, for each manually edited region *a*, a region *b* in the corresponding image with salient regions that IVSeg produced, such that *b* overlapped with *a* and had the smallest mismatch with *a* compared to all other regions in the corresponding image that overlapped with *a*. The closest matching region was the best potential result for IVSeg with respect to the manually generated template; therefore, poor results in this experiment would imply poor results for the entire system.

7.2.3.2 Data Analysis for Experiment 3

In the first part of experiment 3, I compared salient regions that IVSeg generated to manually edited sections. I generated two sets of observations. Set 1 included NVM figures for pairs of manually edited results of the same section by different experts; these figures measured inter-expert variability. Set 2 comprised NVM figures for pairs of editing results for the same section, where an expert generated one result via manual editing, and I generated the other result with IVSeg. To compare the *difference* between manual-editing results and IVSeg results to inter-expert variability, I tested the hypothesis that NVM measurements for sets 1 and 2 differed significantly by using a paired *t*-test that compared mean NVM values for 75 corresponding sections.

In the second part of experiment 3, I performed a similar comparison for the regions that IVSeg ultimately selected via the region-grouping procedure. The manually generated data for this study constituted a subset of the 75-section sample. The subset included the 43 sections that belonged to the 10 cases for which I had results of both manual editing and editing with IVSeg. I generated two sets of observations from edits of the 43 sections in a way that was similar to what I described for the first part of this experiment. Set 1 was a subset of the observations in the first part. The observations in set 2, however, pertained to IVSeg's final result, as opposed to pertaining to the best possible result with respect to a specific expert's edits. As I did for the first part, I tested the hypothesis that the NVM measurements for sets 1 and 2 differed significantly by using a paired *t*-test that compared mean NVM values, but this time the comparison was for 43 corresponding images.

Table 7.2 summarizes the cases that I used in the three experiments.

Table 7.2 Cases that I used in each experiment.

Case	Experiment 1	Experiment 1	Experiment 2	Experiment 3	Experiment 3
	part 1	part 2		part 1	part 2
1	*	*	*	*	*
3	*	*	*	*	*
6	*	*	*	*	*
10	*			*	
11	*	*	*	*	*
15	*	*	*	*	*
18	*	*	*	*	*
20	*	*	*	*	*
22	*			*	
26	*	*	*	*	*
31	*			*	
32	*	*	*	*	*
39	*			*	
41	*			*	
42	*				
50	*	*	*	*	*

7.3 Evaluation Results

In Sections 7.3.1 to 7.3.3, I present the results of the three experiments.

7.3.1 Results for Experiment 1

Of the 16 cases that users edited with IVSeg, three had significant leaks into the bone (see Table 7.1). In all three cases, IVSeg's processing resulted in false-positive regions from the pelvic bone. Most likely, the reason for the error was that small branches from the iliac arteries touched elliptic salient regions that originated from the pelvic bone, such that all the criteria for inclusion were met (see Section 6.2). In two other cases (cases 1 and 50), there was a leak into a small segment of the pelvic bone. In both of these cases, the spurious volume was about 1 percent of the entire volume, and was easily removable with a single volume crop.

7.3.1.1 Assessment at the Volume Level

Figure 7.5 shows the distribution of NVM for the 10 cases that users edited with both methods. For these cases, the mean NVM for the entire volumes was 15.4 percent, and the standard deviation was 15.3 percent. Note that the relative mismatch was high for cases 11 and 26, in which the experts included kidney regions when they edited manually but not when they edited with IVSeg. When I excluded cases 11 and 26 from consideration, the mean NVM was 8.3 percent, and the standard deviation was 3.5 percent. Thus, the two editing methods produced results that were mostly in agreement with respect to inclusion of voxels.

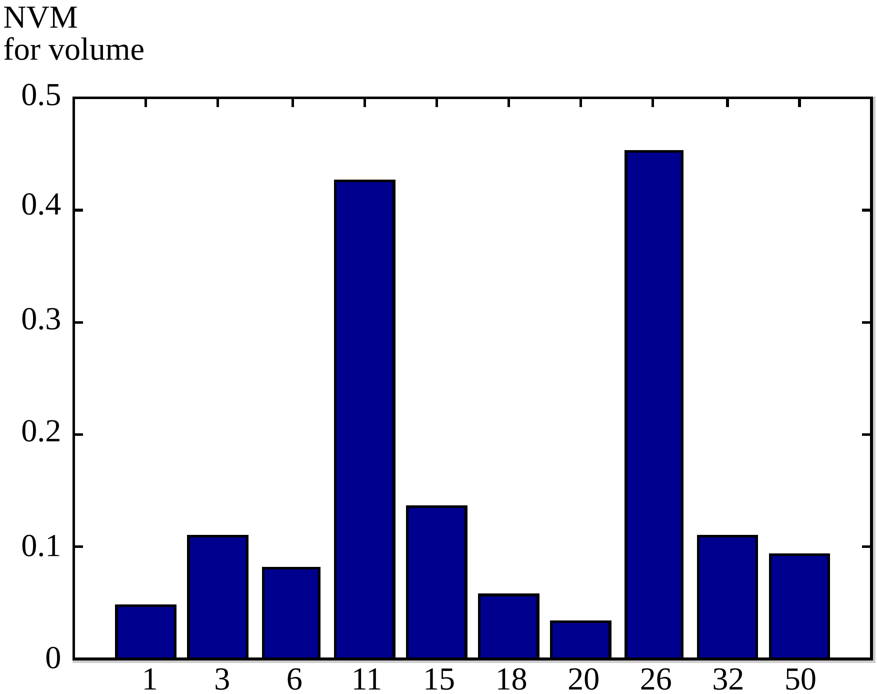


Figure 7.5 NVM for entire volumes that experts produced via manual editing and via editing with IVSeg.

Figure 7.6 illustrates the similarity of the results of manual editing and of editing with IVSeg by the same expert for case 20. Note that several small branches are missing from the results that the expert produced with IVSeg. The ratio of mismatching voxels to voxels that experts included in the manually edited volume was 0.034. Because IVSeg identified salient regions for many of the missing branches, I surmise that, with additional mouse clicks, the expert could have obtained a smaller mismatch ratio.

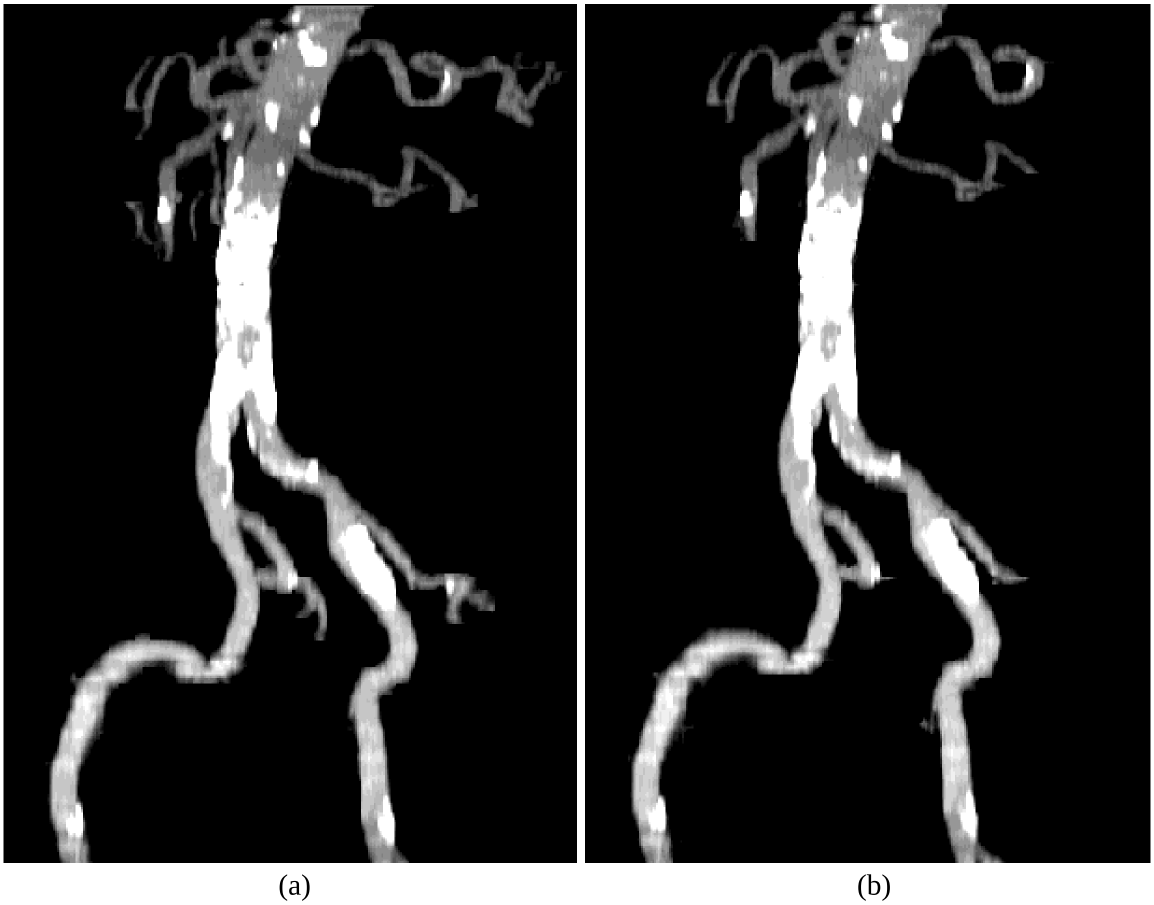


Figure 7.6 A MIP view of an edited case 20 (156 sections). (a) An edited sequence that an expert produced manually. The duration of the manual-editing session was 99 minutes. (b) An edited sequence that an expert produced with IVSeg. To edit the volume, the expert clicked with a mouse within 41 regions for 17.5 minutes. The window setting for this view was 257 HU; the level setting was -694 HU.

7.3.1.2 Assessment at the Region Level

In the analysis of NVM at the region level, I looked at the entire set of observations for which both regions comprised 50 voxels or more, as well as two subsets of the observations: observations for which $\text{NVM} < 1$ and those for which $\text{NVM} \geq 1$. Recall that the NVM is the number of voxel mismatches divided by the number of voxels in the baseline—here, the manually traced region. A value of $\text{NVM} \geq 1$ indicated that the region that IVSeg generated was at least twice as large as the corresponding manually traced region. Figure 7.7 shows the NVM for the 59 pairs of regions for which $\text{NVM} \geq 1$; these pairs constituted 1.6 percent of the 3773 observations. Note that most of the larger NVM values occurred for regions in cases 11 and 26, where the experts included kidney tissue when they edited manually but not when they used IVSeg.

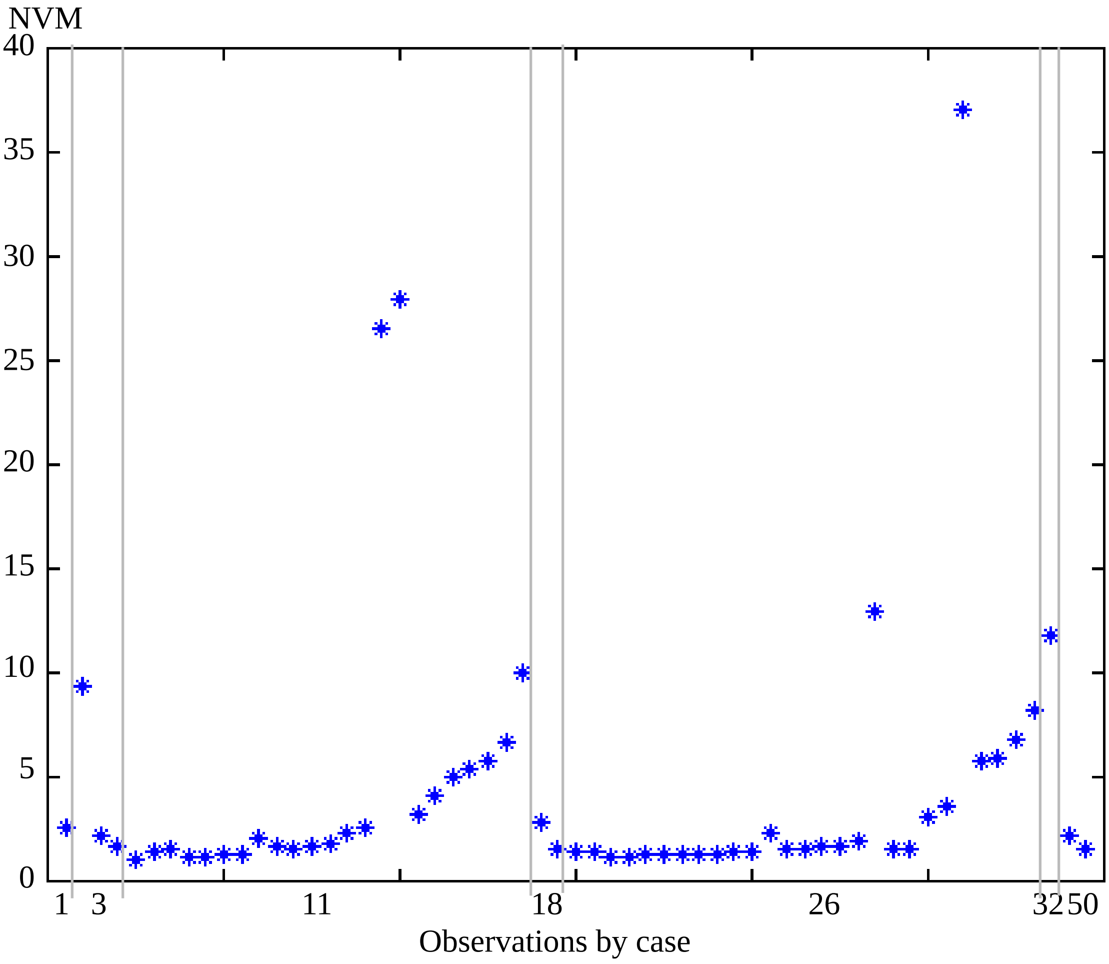
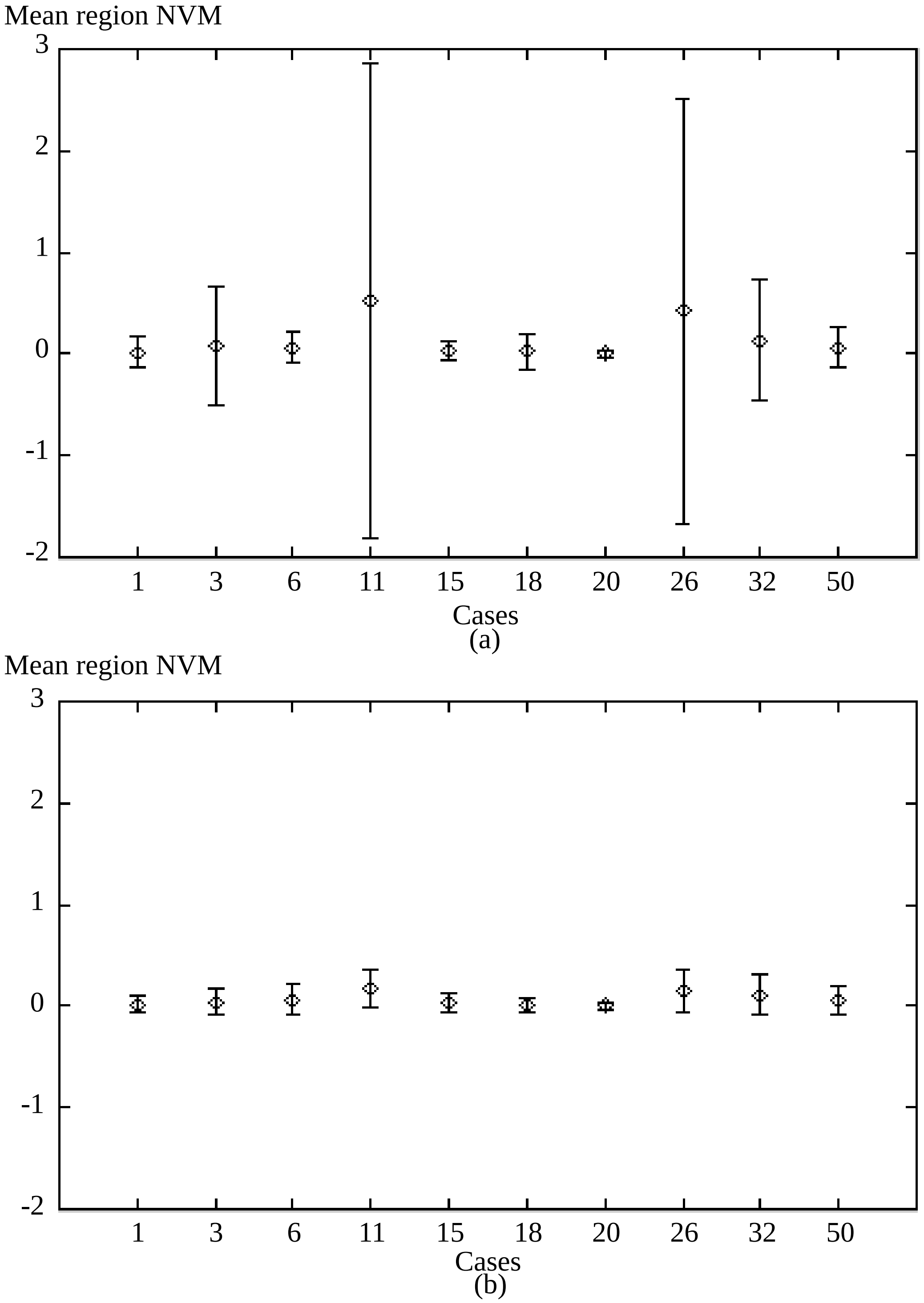


Figure 7.7 Observations for which $\text{NVM} \geq 1$ displayed by case.

Figure 7.8 shows a plot of the mean and standard deviation NVM for corresponding regions in the 10 cases. Figure 7.8 presents the results for all observations, where the mean NVM was 13.5 percent and the standard deviation was 18.3 percent. Figure 7.8 b shows the results for the repeated analysis for only observations with $\text{NVM} < 1$, where the mean NVM was 6.2 percent, and the standard deviation was 5.8 percent. Because the number of observations excluded in the repeated analysis was relatively small, and because some of these observations were outliers, I believe that the results in the repeated analysis represent adequately the performance of IVSeg.



(a)

(b)

Figure 7.8 Mean NVM for corresponding regions in 10 cases that experts handled with manual editing and with IVSeg. (a) Results for all 3773 observations; the mean NVM was 13.5 percent and the standard deviation was 18.3 percent. (b)

Results for the 3718 observations for which $\text{NVM} < 1$; the mean NVM was 6.2 percent, and the standard deviation was 5.8 percent.

7.3.2 Results for Experiment 2

The paired *t*-test showed that there was a significant difference between editing times with the two methods, with p>0.001 (Figure 7.9). Table 7.3 shows the times that I measured for manual editing and for editing with IVSeg. The mean and standard-deviation editing times per section for manual editing of the 10 cases were 0.53 and 0.17 minutes, respectively; those for editing the 10 cases with IVSeg were 0.10 and 0.03 minutes, respectively. The average number of user-specified points was 52. Note that more mouse clicks were necessary for editing cases for which the flow-channel shape was distorted—for example, because of protruding thrombi, or because the vessel was dissected (see Table 7.1).

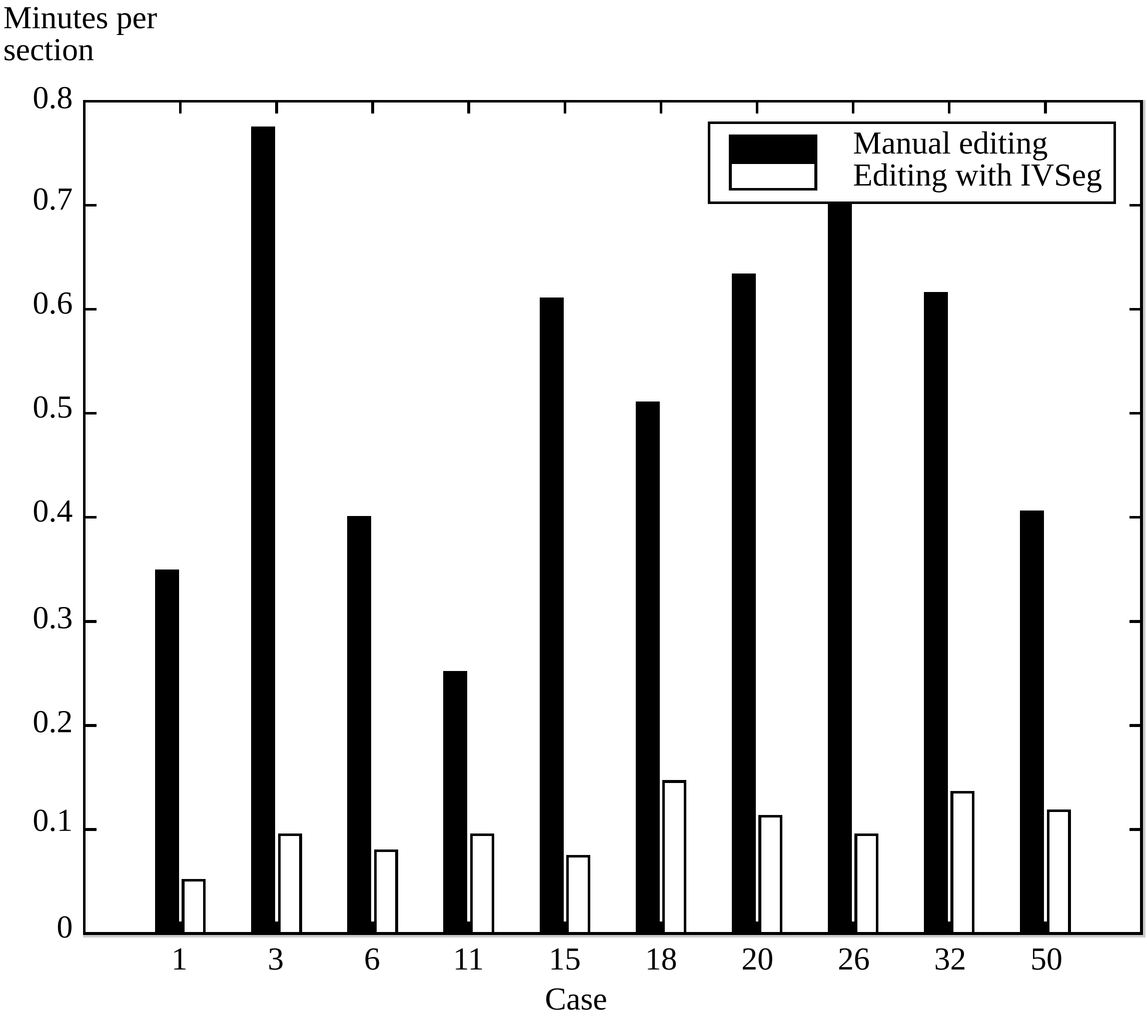


Figure 7.9 Bar chart of user time per section for the 10 cases, for manual editing and editing with IVSeg.

Table 7.3 Duration of manual editing and of editing with IVSeg (in minutes).

Case No.	No. of Sections	Manual editing—total time	Manual editing—time per section	IVSeg editing—total time	IVSeg editing—time per section	IVSeg editing—no. of mouse clicks	Ratio of mismatched voxels to total volume
1	189	66	0.3492	9.5	0.0503	25	0.0485
3	181	140	0.7735	17.0	0.0939	54	0.1092
6	162	65	0.4012	13.0	0.0802	39	0.0799
11	159	40	0.2516	15.0	0.0943	23	0.4250
15	187	144	0.6096	14.0	0.0749	37	0.1355
18	157	80	0.5100	23.0	0.1465	81	0.0568
20	156	99	0.6346	17.5	0.1122	41	0.0338
26	173	123	0.7110	16.5	0.0954	55	0.4516
32	198	122	0.6162	27.0	0.1364	124	0.1099
50	178	72	0.4045	15.0	0.1172	34	0.0919

Editing time and number of mouse clicks for editing with IVSeg reflect a typical activation of the system with the current, suboptimal, computer-response time of several hours per case. Assuming a truly interactive response time, the user could edit a case in an incremental fashion, and place only mouse clicks that were necessary based on the MIP feedback.

7.3.3 Results for Experiment 3

Recall that in the third experiment I performed a paired *t*-test to determine whether edits that experts generated with IVSeg compared to edits that the experts generated via manual tracing as the latter edits compared to one another. I compared two sets of observations: set 1 included NVM figures for pairs of edited results of the same section by different experts, and set 2 included NVM figures for edited results of the same section that experts generated with the two methods. The *t*-test showed that there was no significant difference between the sets of observations 1 and 2. Recall that I performed the *t*-test twice, once for salient regions that IVSeg generated by analyzing isolabel contour maps, and once for the final results that IVSeg generated with the region-grouping procedure. For the first test, the total number of sections was 75 and p>0.114. To calculate the power of the test, I assumed that the standard deviation of the underlying population was the standard deviation of mean image NVM in set 1, 0.23. The difference of mean image NVM in sets 1 and 2 was 0.22. With a sample size of 75, and for $\alpha = 0.05$, the power of the test was 0.99. These results suggested that the salient regions that IVSeg generated were comparable to those that experts generated via manual editing, and that therefore the former regions could be used as input to subsequent processing for selection of target regions.

For the second test, the total number of sections was 43 and p>0.3. To calculate the power of the paired *t*-test, I assumed that the standard deviation of the underlying population was the standard deviation of mean image NVM in set 1, 0.24. The difference of mean image NVM in sets 1 and 2 was 0.12. With a sample size of 43, and for $\alpha = 0.05$, the power of the test was 0.86. Based on these results, there was not sufficient evidence to reject the null hypothesis, that mean NVM for sets 1 and 2 did not differ significantly. Therefore, I accepted the null hypothesis.

The absence of a salient region that encloses the missing voxels indicates that IVSeg failed in step 1. While I was experimenting with IVSeg, I observed that it failed in step 1 for only several cases and for only a few small regions. To allow the user to add regions that step 1 failed to identify, we could built into IVSeg manual-tracing capabilities.

7.4 Evaluation Conclusions

The three experiments that I conducted to evaluate IVSeg with respect to the common practice of manual editing showed that IVSeg allowed experts to edit CTA volumes with significantly less effort, as measured by user-input time, without compromising the adequacy of the results. From the first experiment, I concluded that, in general, IVSeg's results compared favorably to results of manual editing. The leaks that occurred for editing of three cases with IVSeg suggested that, to make IVSeg a clinically useful tool, we need to improve the region-grouping procedure as to avoid false positives for situations where bone segments that touch artery branches have elliptic shape. From the second experiment, I concluded that user time required for editing CTA volumes with IVSeg was reduced by a factor of 5, on average, compared to user time required for manual editing.

From the third experiment, I concluded that we cannot distinguish between editing with IVSeg and manual editing based on differences between editing results for the two methods. That editing with IVSeg was successful for extracting flow channels from cases with a variety of pathologies—for example, adjacent thrombi, dilation, and dissection—suggests that IVSeg is robust with respect to these pathologies for editing flow channels in CTA cases.

I demonstrated the validity of the hypothesis that *IVSeg allows radiology experts to edit volumes of medical images robustly in less than 10 seconds per section on average, with results comparable to those produced by experts using conventional editing tools*. Due to limited resources, I was not able to compare inter- and intra-expert variability when experts use IVSeg to inter- and intra-expert variability of manual editing. Because the current implementation prohibited fully interactive use of the system, and because of limited availability of the participating radiology experts, in several cases I had to complete the experts' editing tasks by introducing additional mouse clicks where the MIP views designated missing regions. I envision that my participation did not introduce bias to the study because I merely activated the system in a way that a radiology expert could do easily. This evaluation study also did not look at learning-curve effects. I expect that there is such an effect for using IVSeg, and that the user time that is required to edit a case with IVSeg reduces as a user becomes more fluent in using the system, and when the implementation allows the system to be truly interactive.

7.5 Summary

I presented IVSeg and demonstrated its use for editing CTA cases. I described a formal evaluation of the system, which I conducted to verify the research hypothesis for this dissertation. In the evaluation, I measured the mismatch between results of editing with IVSeg and results of manual editing, I compared user-input time for editing with both methods, and I compared the difference between results of the two methods to inter-expert variability of manual editing. The evaluation results supported the research hypothesis, that IVSeg allowed experts to edit CTA cases in less than 10 seconds per section, with results that were similar to results of manual editing.

In Chapter 8, I present conclusions from my research.

Chapter 8

Conclusion

On-going technology innovations increase the ease with which we can obtain high-quality large volumetric image data sets to support medical care. We need accurate and efficient image-processing tools to realize the potential benefits embodied within image information. The research that I have described in this dissertation demonstrated promising methods and a system that could help radiology experts to edit large volumetric datasets. I now provide three conclusive discussions. Section 8.1 summarizes the contributions of my work, Section 8.2 discusses limitations of the work, and Section 8.3 suggest directions for further research. Section 8.4 is the chapter summary.

8.1 Summary of Contributions

The contributions of this work all relate to extraction of visual information, as necessary to support diagnosis, quantification, and therapy planning. The segmentation methods that I presented are different from current methods in that they are robust and yet require little prior knowledge about the contents of the input. Within IVSeg, these methods allow radiologists to edit image volumes more easily than with conventional editing tools. In the following sections, I discuss specific contributions. Section 8.1.1 discusses contributions to 2D image segmentation. Section 8.1.2 describes contributions to composition of vessel trees through classification of regions, and to ellipse fitting. Section 8.1.3 outlines contributions to editing of image volumes. These contributions are also summarized in related papers [80-86].

8.1.1 2D Segmentations Using Analysis of Isolabel-Contour Maps

The 2D segmentation method that underlies IVSeg's procedure for identification of salient regions takes advantage of PVE, which is inherent in image acquisition, independent of imaging modality. Because of PVE, imposing a fine partitioning of the intensity range results in an isolabel-contour map in which contours that reside along the boundaries of objects carry the shape of the object and thus are similar to each other. The segmentation method interprets consecutive isolabel contours that have similar shape as an indication for the boundary of an object with that shape. With this intrinsic-shape criterion, the method can meet the challenge of distinguishing between objects of similar appearance that are close to each other. To the best of my knowledge, comparing the shape of isolabel contours to detect transitions from within an object into the background is a novel idea. The work of Zheng and colleagues [87] is related to mine in that it analyzed features of regions that resulted from thresholding to determine the region that best represented the target object. Unlike me, however, Zheng and colleagues used three coarsely spaced thresholds and general features such as size growth and central-position shift.

Because my method uses shape information that it extracts from the images, it does not rely on prior shape models to find object boundaries. Consequently, my method's applicability is not restricted to objects of particular shape. My method requires only a fairly lax specification of the intensity range for target objects; in the sensitivity analysis (see Section 5.3.1), I showed that segmentation results were insensitive to 5-percent changes in the upper and lower intensities that the user specified for the target objects. For each object in a 2D section, the method finds an isolabel contour that approximates the boundary of the object. From my evaluation study (see Section 7.3.3) I concluded that the approximated boundaries that IVSeg found differed from boundaries that expert delineated using conventional editing tools as much as the latter boundaries differed from one another.

8.1.2 Composition of 3D Vessel Trees Through Classification of Regions

IVSeg's region-grouping procedure composes 3D tree structures of tortuous vessels from a sequence of presegmented regions. The input regions are images of objects whose intensities lie within a predefined range. Through application of dynamic ellipse-overlap criteria, the procedure meets the challenge of distinguishing between input regions that belong to previously included vessels and those that are not related to these vessels. I derived a formal expression for the overlap of elliptic sections through a cylinder, and I validated the expression by showing that, at the limits, the expression value agrees with our intuition.

IVSeg models vessel structures as compositions of cylindrical components, whose sections with arbitrary planes that cut through their axes at only one point result in ellipses. Because the overlap criteria are applicable to only elliptic regions, and because, for certain vessel substructures, intersections with the acquisition planes result in nonelliptic sections, IVSeg searches for ellipses in orthogonal reformations of the data. I proved that intersections of cylindrical components by three orthogonal planes result in an elliptic section in at least one of the intersecting planes. This proof provides a guarantee that, by searching for ellipses in three orthogonal reformations of the data, IVSeg considers all vessel substructures for inclusion.

To determine whether a region is elliptic, I used a known Fourier series approximation of contours in a novel way. Kuhl and Giardina [69] showed that the Fourier approximation to a contour can be viewed as the addition of rotating phasors with elliptic loci. I proved that the projections of the first harmonic along the x and y axes form an **optimal** elliptic approximation to the contour. Thus, to find the optimal ellipse fit to a region, we need to compute a one-harmonic Fourier series, a task that we can do in linear time. This ellipse-fitting procedure is more efficient than are conventional optimum-seeking registration techniques, which require at least cubic time.

8.1.3 IVSeg: A Robust Volume Editor That Requires Little User Input

I incorporated the methods that I developed into IVSeg, a system that radiology experts can use to edit image volumes by selecting samples of target objects with a mouse. Typically, the system requires about 6 seconds of user time per section, as opposed to 30 seconds of user time per section that conventional editing tools require. A sensitivity analysis and a comparison to conventional thresholding showed that the system is robust (see sections 5.3.1, and 6.3). A comparison of editing with IVSeg to manual editing showed that results of editing with IVSeg compared to those of manual editing as the manual-editing results compared to one another (see Section 7.3.3).

8.2 Limitations

I discuss next the limitations of the methods that I developed. In Section 8.2.1, I describe assumptions that limit the applicability of the method for identification of salient regions. In Section 8.2.2, I outline assumptions that limit the applicability of the region-grouping method. In Section 8.2.3, I explain how computation time prohibits a truly interactive use of the current implementation of IVSeg.

8.2.1 Assumptions That Limit the Applicability of the Method for Identification of Salient Regions

The method for identification of salient regions is based on two assumptions. These assumptions characterize target datasets for which the method is likely to succeed.

8.2.1.1 Assumption 1: There Exist Intensity Thresholds That Separate Between Target Objects and Their Background

The first assumption is that there exist intensity thresholds that separate well between target objects and structures that surround them. This assumption is less likely to hold as the ratio of noise standard deviation to the contrast between the true object and its true background increases. Although noise-removal methods—for example, the anisotropic-diffusion filter that I used for this work [76]—could improve the contrast-to-noise ratio, it is likely that a reduced level of noise will remain. For CTA images, IVSeg was able to separate the aorta flow channel from surrounding structures—for example, the vena ceva; the typical contrast in these images was about 200 HU, and the standard deviation of the noise in the arteries was about 25 HU. However, IVSeg was not able to separate thrombi from the vena ceva; the contrast was only about 20 HU, and the standard deviation of the noise within the thrombi was similar to that of the aorta.

8.2.1.2 Assumption 2: Sections of Target Objects Are Larger Than 10 Pixels

The second assumption is that sections of target objects are larger than about 10 pixels in area. For small objects, whose shape is distorted due to digitization, the intensity fall that is associated with PVE is not as smooth as it is for large objects, and thus the shape of isolabel contours along the boundary of the object image does not reflect well the shape of the underlying structure. Consequently, the contour-matching procedure is likely to detect large shape changes for contours that enclose voxels that belong to the same object. This false detection either would lead to elimination of a salient region for the object—in the event that IVSeg failed to detect two consecutive contours with similar shape—or might compromise the coherence of IVSeg's depiction of the salient region.

8.2.2 Assumptions That Limit the Applicability of the Region-Grouping Method

The method for grouping regions is based on two assumptions. These assumptions characterize target structures for which the method is likely to succeed.

8.2.2.1 Assumption 1: We Can Represent Structures of Interest as Compositions of Right Cylinders

The first assumption is that we can represent adequately the target structures as compositions of right-cylinder components of various sizes and orientations. This assumption is reasonable for blood vessels, but it does not hold for structures such as, vertebra, the cortex of the brain, or breast-cancer lesions. In general, I expect that the region-grouping method will fail for structures that have many dense protrusions.

8.2.2.2 Assumption 2: We Can Use Overlap Criteria to Distinguish Between Regions That Belong to Different Structures

The second assumption is that coherent regions in consecutive sections that belong to different structures will not comply with the overlap criteria. The evaluation study pointed out three cases for which this assumption did not hold. For these cases, editing sessions with IVSeg resulted in considerable leaks from fine branches of the iliac arteries into the pelvic bone. The reason for the leaks was that small sections of a highlighted branch of the iliac artery and small sections of the pelvic bone both complied with IVSeg’s overlap criteria, and consequently IVSeg misclassified the bone sections as components of the target arterial tree. These leaks suggest that there might be worst-case anatomic arrangements for which the current region-grouping criteria would not separate the structures adequately.

8.2.3 Computation Time

The time that IVSeg takes to identify salient regions and to group regions based on user input is too long to support fully interactive use of the system. For example, IVSeg requires about four minutes to find salient regions in a 256×256 section on a 200-mHz Unix workstation, or about 30 hours per case with 150 sections. IVSeg takes about 5 hours to group regions for a sequence of 150 sections. Because the user is involved mostly in iterative activation of IVSeg’s region-grouping procedure, computation time for this procedure is the significant factor that prohibits IVSeg from being a truly interactive editor. I expect that computation time will become negligible with the introduction of new high-speed computers.

8.3 Directions for Future Research

The evaluation study and the limitations that it pointed out suggest several directions for further research. In Section 8.1.3, I discuss extensions to the current methods. In Section 8.3.2, I suggest improvements to the current implementation of IVSeg. In Section 8.3.3, I outline additional evaluation studies that are necessary before IVSeg can be put to clinical use.

8.3.1 Extensions to Current Methods

I propose five extensions to the current methods in IVSeg, as detailed in Sections 8.3.1.1 through 8.3.1.5.

8.3.1.1 Determination of SDT as a Function of Target Object and Imaging Parameters

I derived through experimentation the SDT value that I incorporated into IVSeg. Preliminary experiments with that SDT indicated that it was unsuitable for detection of objects whose sections smaller than about 10 pixels. The fact that the SDT was adequate for detection of large objects but not for detection of small objects might relate to the shape distortion of images of small objects, a distortion that results from digitization. I expect that IVSeg's performance could be improved for specific applications if a researcher derived an expression for SDT as a function of the size of the target object, of image acquisition parameters—for example, pixel size, section thickness, section spacing—and of noise.

8.3.1.2 Adjustment of the Boundary of Salient Regions According to Local Image Characteristics

Currently, IVSeg assumes that the final similar contour in a sequence of similar isolabel contours is the boundary of an object. However, because isolabel contours reflect the results of aggregation of voxels whose intensities are within a bin defined by the multilevel-thresholding partition that IVSeg uses, the true boundary is most likely to be between the final similar contour and the contour that follows. I suggest that programmers enhance IVSeg's procedure for identification of salient regions to include refining the boundary of salient regions according to characteristics that are local to the boundary. For example, programmers can apply an active contour model with the final similar contour as a promising initial condition, and they can activate forces that direct outwards the search for a better approximation to the true boundary.

8.3.1.3 Tracking of Object Features Beyond Two Consecutive Sections

In Section 8.2.2.2, I stated that IVSeg's overlap criteria do not guarantee correct classification of salient regions. I suggest that researchers enhance these criteria with predictions of plausible object features, based on the aggregate structure that IVSeg included at a given point in time, and thus to reduce the risk of including false-positive salient regions. For example, leaks from iliac arteries into the pelvic bone could be avoided if we kept track of the diameter of the vessel along a predefined path length and verified that we did not include regions that indicate an abrupt diameter change.

8.3.1.4 Extension of the Overlap Criteria to Include Organs of Arbitrary Form

The applicability of IVSeg's region-grouping procedure is limited to objects that could be represented adequately within the generalized-cylinder framework. Extending the expression of section overlap to include sections of arbitrary form will allow us to broaden the range of objects that users could extract with IVSeg. One possible approach to this enhancement would be to apply the following steps to a given pair of regions in consecutive sections. First, we could approximate the contour of one of the regions with a spline. Second, we could generate another spline by shifting the first spline to superimpose the other region. The shift would reflect the orientation of the imaged object, the object's diameter, and the image-acquisition parameters: section thickness and section spacing. Third, we could derive the expected overlap for the regions by computing the intersection between the two splines.

8.3.1.5 Matching of Isolabel Surfaces

We could eliminate the need for the region-grouping procedure by enhancing IVSeg's procedure for identification of salient regions to identify salient **blobs**, or subvolumes, by matching isolabel surfaces. Within this entirely 3D framework, the user would select salient 3D blobs instead of salient 2D regions. Surface matching would identify by default any spatial relations between related volumetric components, such as branches or protrusions. I expect that isolabel-surface matching would be computation intensive, but that it might lead to higher accuracy—for isotropic data—than the 2D approach, due to the use of 3D shape information.

8.3.2 Improvement of Implementation

Three main improvements are necessary before IVSeg could be put to clinical use. First, we need to speed up the running time of IVSeg's region-grouping procedure so that it can support a truly interactive mode of operation. I expect that designing data structures that transfer information about included regions from one iteration to the next will avoid the necessity of recomputing this information for every iteration. Second, we need to enhance IVSeg's user interface to provide more detailed feedback to the user. Because IVSeg's sagittal and coronal MIP views are merely projections of the included volume, they do not provide the best possible feedback to the user. We can provide more useful feedback by incorporating into IVSeg axial views that delineate the state of voxel inclusion at any given time. To help the user override IVSeg's inclusion decisions when necessary, we must allow her to modify editing results—for example, by moving contour points or by tracing manually. Third, we need to restructure the architecture of IVSeg to allow asynchronous activation of independent system components.

8.3.3 Additional Evaluation Studies

The experiments that I conducted to evaluate my work merely demonstrate the potential of IVSeg and its methods for editing medical image volumes. We could replicate the design of these experiments to quantify the characteristics that IVSeg assumes for the input data (e.g., contrast-to-noise ratio) to allow more realistic expectations about performance. IVSeg does not make assumptions about the modality of input images. Nevertheless, to determine the usefulness of IVSeg for specific editing situations, we need to evaluate IVSeg within these situations. We could learn more about the general applicability of IVSeg by extracting a variety of objects from CT, MR, ultrasound, and nuclear-medicine images. Although each modality has its unique characteristics, I expect that IVSeg will segment successfully from these modalities objects that have a regular shape, a fairly uniform intensity, and a notable contrast-to-noise ratio.

8.4 Summary

In this chapter, I described the contributions of my work to segmentation of 2D sections, to extraction of 3D objects form medical image volumes, and to editing of clinical cases. Then, I outlined the limitations of the current work; they relate to the assumptions that underlie the methods, to worst-case anatomy, and to the current implementation. Finally, I highlighted directions for future research that could address the limitations of the current work. I suggested additional evaluations that would be necessary to promote the use of IVSeg. It is my hope that subsequent research will evolve from this work that will result in the provision of volume-editing tools for clinical use.

Appendix 1

The Error Associated with a Spatial Transformation of an Optimal Ellipse

I prove that the error associated with a spatial transformation of an optimal ellipse is a sum of orthogonal exponential terms:

Let

$$M_{E_1} = \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t}, \tag{1.1}$$

be the optimal ellipse, and let

$$M_{E_2} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t}, \tag{1.2}$$

be an ellipse resulting from a spatial transformation of M_{E_1}.

Let

$$\left\| M_c - M_{E_2} \right\| = \left\| M_{E_1} - M_{E_2} + \begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \sum_{n=2}^{T-1} \begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} e^{i\theta n t} \right\| \tag{1.3}$$

be the approximation error associated with M_{E_2}.

If we combine the expression for the two ellipses we get

$$M_{E_1} - M_{E_2} = \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t} - \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta t} . \quad (1.4)$$

Rearranging terms from Equation 1.4 we get

$$M_{E_1} - M_{E_2} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} e^{i\theta 0 t} + \left(1 - \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \right) \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} e^{i\theta 1 t} . \quad (1.5)$$

From the definition of a Fourier series approximation to a discrete time finite duration signal we get

$$\begin{bmatrix} c_{x_n} \\ c_{y_n} \end{bmatrix} = \frac{1}{T} \sum_{l=0}^{T-1} \begin{bmatrix} x(l) \\ y(l) \end{bmatrix} e^{-i\theta n l} . \quad (1.6)$$

From Equations 1.5 and 1.6 we get,

$$H_1 = \left(1 - \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \right) \frac{1}{T} \sum_{l=0}^{T-1} \begin{bmatrix} x(l) \\ y(l) \end{bmatrix} e^{-i\theta 1 l} e^{i\theta 1 t} ,$$

$$H_2 = \begin{bmatrix} c_{x_0} \\ c_{y_0} \end{bmatrix} + \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} = \left(\frac{1}{T} \sum_{l=0}^{T-1} \begin{bmatrix} x(l) \\ y(l) \end{bmatrix} + \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} \right) e^{-i\theta 0 l} e^{i\theta 0 t}$$

$$.$$

We can rewrite Equation 1.3 without the norm signs as

$$M_c - M_{E_2} = \sum_{n=2}^{T-1} \frac{1}{T} \left(\sum_{l=0}^{T-1} \begin{bmatrix} x(l) \\ y(l) \end{bmatrix} e^{-i\theta n l} \right) e^{i\theta n t} + H_1 + H_2 ,$$

After changing the order of the summation terms,

$$M_c - M_{E_2} = \sum_{l=0}^{T-1} \begin{bmatrix} x(l) \\ y(l) \end{bmatrix} \frac{1}{T} \sum_{n=2}^{T-1} e^{i\theta n(t-l)} + H_1 + H_2,$$

and after combining terms we get,

$$M_c - M_{E_2} = \sum_{l=0}^{T-1} \alpha_l \frac{1}{T} \sum_{n=0}^{T-1} e^{i\theta n(t-l)}. \tag{1.7}$$

Because

$$\frac{1}{T} \sum_{n=0}^{T-1} e^{i\theta n(t-l)} = \delta_{t-l},$$

we concluded that the summation terms on the right hand side of Equation 1.7 are orthogonal.

Q.E.D.

Appendix 2

Derivations and Limits Analysis for Relative-Overlap Expression

In Section 6.1.3 I derived the expression for the relative overlap between consecutive cylinder sections, which are elliptical, as a function of the semimajor axis e of the ellipses and the overlap between semimajor axes of ellipses in consecutive sections o.

$$RO = 1 - \frac{2}{\pi}\left(\operatorname{asin}\left(1 - \frac{o}{2e}\right) + \left(1 - \frac{o}{2e}\right)\sqrt{\frac{o}{e}\left(1 - \frac{o}{4e}\right)}\right). \tag{2.1}$$

The following derivation leads to the definition of RO as a function of section thickness w, section spacing s, angle in which cylinder intersects the imaging plane α, and cylinder diameter d.

Recall that

$$o = k(a+b) = k\left(\frac{w-s}{|\tan\alpha|} + \frac{d}{|\sin\alpha|}\right) = k\left(\frac{(w-s)|\cos\alpha| + d}{|\sin\alpha|}\right), \tag{2.2}$$

and that

$$e \;=\; k\left(\frac{b+c}{2}\right) \;=\; k\left(\frac{d+w|\cos\alpha|}{2|\sin\alpha|}\right). \qquad (2.3)$$

From equations 2.2 and 2.3 we get

$$1-\frac{o}{2e} \;=\; 1-\frac{k\left(\frac{(w-s)\,|\cos\alpha|+d}{|\sin\alpha|}\right)}{2k\left(\frac{d+w|\cos\alpha|}{2|\sin\alpha|}\right)} \;=\; 1-\frac{(w-s)\,|\cos\alpha|+d}{d+w|\cos\alpha|},$$

$$1-\frac{o}{2e} \;=\; \frac{d+w|\cos\alpha|-w|\cos\alpha|+s|\cos\alpha|-d}{d+w|\cos\alpha|} \;=\; \frac{s|\cos\alpha|}{d+w|\cos\alpha|}.$$

From equations 2.2 and 2.3 we also get

$$\frac{o}{e}\left(1-\frac{o}{4e}\right) \;=\; \frac{2\,(\,(w-s)\,|\cos\alpha|+d)}{d+w|\cos\alpha|}\left(1-\frac{(w-s)\,|\cos\alpha|+d}{2\,(d+w|\cos\alpha|)}\right),$$

and from there on,

$$\frac{2\,(\,(w-s)\,|\cos\alpha|+d)}{d+w|\cos\alpha|}\left(\frac{2\,(d+w|\cos\alpha|)\;-\;(\,(w-s)\,|\cos\alpha|+d)}{2\,(d+w|\cos\alpha|)}\right),$$

$$\frac{2\,(w|\cos\alpha|-s|\cos\alpha|+d)}{d+w|\cos\alpha|}\left(\frac{2d+2w|\cos\alpha|-w|\cos\alpha|+s|\cos\alpha|-d}{2\,(d+w|\cos\alpha|)}\right),$$

$$\frac{(w|\cos\alpha|-s|\cos\alpha|+d)}{d+w|\cos\alpha|}\left(\frac{d+w|\cos\alpha|+s|\cos\alpha|}{(d+w|\cos\alpha|)}\right),$$

$$\frac{dw|\cos\alpha|-ds|\cos\alpha|+d^2+\;(w|\cos\alpha|)^2-w|\cos\alpha|s|\cos\alpha|+w|\cos\alpha|d+ws\,(|\cos\alpha|)^2}{(d+w|\cos\alpha|)^2}$$

$$con\dot{t} \qquad \frac{-s^2\,(|\cos\alpha|)^2+ds|\cos\alpha|}{(d+w|\cos\alpha|)^2},$$

and finally,

$$\frac{2dw|\cos\alpha|+d^2+\;(w|\cos\alpha|)^2-s^2\,(|\cos\alpha|)^2}{(d+w|\cos\alpha|)^2} \;=\; \frac{(d+w|\cos\alpha|)^2-s^2\,(|\cos\alpha|)^2}{(d+w|\cos\alpha|)^2} \;=\; \text{""}$$

$$=\left(1-\frac{s^2\,(|\cos\alpha|)^2}{(d+w|\cos\alpha|)^2}\right).$$

Thus,

$$1 - \frac{o}{2e} = \frac{s|\cos\alpha|}{d + w|\cos\alpha|}\,, \tag{2.4}$$

$$\frac{o}{e}\left(1 - \frac{o}{4e}\right) = 1 - \frac{s^2\,(|\cos\alpha|)^2}{(d + w|\cos\alpha|)^2}\,. \tag{2.5}$$

From Equations 2.1 and 2.4 and 2.5 we get,

$$RO = 1 - \frac{2}{\pi}\left(\operatorname{asin}\left(\frac{s|\cos\alpha|}{d + w|\cos\alpha|}\right) + \left(\frac{s|\cos\alpha|}{(d + w|\cos\alpha|)}\right)\sqrt{1 - \left(\frac{s|\cos\alpha|}{d + w|\cos\alpha|}\right)^2}\right).$$

An analysis of limits yields the following results.

For a finite object diameter, as $\frac{s}{w}$ approaches zero

$$\lim_{\frac{s}{w} \to 0} RO = \lim_{\frac{s}{w} \to 0}\left(1 - -\frac{2}{\pi}(\operatorname{asin}(0) + 0)\right) = 1\,.$$

As α approaches $\frac{\pi}{2}$,

$$\lim_{\alpha \to \frac{\pi}{2}} RO = \lim_{\alpha \to \frac{\pi}{2}}\left(1 - -\frac{2}{\pi}(\operatorname{asin}(0) + 0)\right) = 1\,.$$

As d approaches infinity

$$\lim_{d \to \infty} RO = \lim_{\alpha \to \infty}\left(1 - \frac{2}{\pi}(\operatorname{asin}(0) + 0)\right) = 1\,.$$

RO is undefined for $s > \frac{d}{|\cos\alpha|} + w$, because the square root yields a complex number.

References

[1] S. Napel, “Principles and techniques of 3D spiral CT angiography,” in *Spiral CT: Principles Techniques and Clinical Applications*, E. Fishman and R. Jeffrey, Eds. New York: Raven Press, 1995, pp. 167-182.

[2] H. Cline, C. Dumoulin, W. Lorensen, S. Souza, and W. Adams, “Volume rendering and connectivity algorithms for MR angiograhy,” *Magnetic Resonance in Medicine*, vol. 18, pp. 384-394, 1991.

[3] E. Chaney and S. Pizer, “Defining anatomical structures from medical images,” *Seminars in Radiation Oncology*, vol. 2, pp. 215-225, 1992.

[4] G. Tracton, E. Chaney, J. Rosenman, and S. Pizer, “Mask: combining 2D and 3D segmentation methods to enhance functionality,” in *Proceedings of the SPIE Conference on Medical Imaging*, vol. 2299. Bellingham, Washington: The International Society for Optical Engineering, 1994, pp. 98-109.

[5] J. F. Brinkley, “A flexible, generic model for anatomic shape: application to interactive two-dimensional medical image segmentation and matching,” *Computers and Biomedical Research*, vol. 26, no. 2, pp. 121-142, 1993.

[6] W. Lorensen, A. Ferenc, and R. Kikinis, “The exploration of cross-sectional data with a virtual endoscope,” in *Interactive Technology and the New Paradigm for Health Care*,

R. Mattheus and J. Christensen, Eds. Amsterdam, Tokyo: IOS Press, Ohmsha, 1995, pp. 221-230.

[7] S. C. Orphanoudakis, C. Chronaki, and S. Kostomanolakis, “I/sup 2/C: a system for the indexing, storage, and retrieval of medical images by content,” *Medical Informatics*, vol. 19, pp. 109-122, 1994.

[8] G. Rubin, P. Walker, M. Dake, S. Napel, R. Jeffrey, C. McDonnell, R. Mitchell, and D. Miller, “Three-dimensional spiral computed tomographic angiography: An alternative imaging modality for the abdominal aorta and its branches,” *Journal of Vascular Surgery*, vol. 18, pp. 656-664, 1993.

[9] S. Napel, M. Marks, G. Rubin, M. Dake, C. McDonnell, S. Song, D. Enzmann, and R. Jeffrey, “CT angiography with spiral computed tomographic angiography: an alternative imaging modality for the abdominal aorta and its branches,” *Radiology*, vol. 185, pp. 607-610, 1992.

[10] R. B. Schwartz, K. M. Jones, D. M. Chernoff, S. K. Mukherji, R. Khorasani, H. M. Tice, R. Kikinis, S. M. Hooton, P. E. Stieg, and J. F. Polak, “Common carotid artery bifurcation: evaluation with spiral CT. Work in progress,” *Radiology*, vol. 185, pp. 513-519, 1992.

[11] M. Katz, E. Konen, J. Rozenman, A. Szeinberg, and Y. Itzchak, “Spiral CT and 3D image reconstruction of vascular rings and associated tracheobronchial anomalies,” *Journal of Computer Assisted Tomography*, vol. 19, no. 4, pp. 564-568, 1995.

[12] L. Zatz, “Basic principles of computed tomography scanning,” in *Radiology of the Skull and Brain: Technical Aspects of Computed Tomography*, vol. 5, T. Newton and D. Potts, Eds. Saint Louis: Mosby, 1981, pp. 3853-3876.

[13] P. Joseph, “Artifacts in computed tomography,” in *Radiology of the Skull and Brain: Technical Aspects of Computed Tomography*, vol. 5, T. Newton and D. Potts, Eds. Saint Louis: Mosby, 1981, pp. 3956–3991.

[14] D. Hentschel, J. Ezrielev, R. Fisher, C. Flanders, A. Bani-Hashemi, C.-C. Liang, S.-P. Liou, S. Samaddar, A. Singh, and D. Ney, “Techniques for editing and visualizing CT-angiographic data,” in *Proceedings of the Conference on Visualization in Biomedical Computing*, vol. 2359, R. A. Robb, Ed.: SPIE, 1994, pp. 307-318.

[15] P. Sahoo, S. Soltani, A. Wong, and Y. Chen, “A survey of thresholding techniques,” *Computer Vision, Graphics, and Image Processing*, vol. 41, pp. 233-260, 1988.

[16] H. Cline, W. Lorensen, R. Kikinis, and F. Jolesz, “Three-dimensional segmentation of MR images of the head using probability and connectivity,” *Journal of Computer Assisted Tomography*, vol. 14, pp. 1037-1045, 1990.

[17] R. Gonzalez and R. Woods, *Digital Image Processing*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1993.

[18] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practices*. Reading, Massachusetts: Addison-Wesley, 1990.

[19] E. Fishman, C. Liang, B. Kuszyk, S. Davi, D. Heath, D. Hentschel, S. Duffy, and A. Gupta, “Automated bone editing algorithm for CT angiography: preliminary results,” *American Journal of Roentgenology*, vol. 166, pp. 669-672, 1996.

[20] J. Duncan, L. Staib, T. Birkholzer, R. Owen, P. Anandan, and I. Bozma, “Medical image analysis using model-based optimization,” in *Proceedings of the First Conference on Visualization in Biomedical Computing*. Atlanta, Georgia: IEEE Comput. Soc. Press, 1990, pp. 370-377.

[21] L. Clarke, R. Velthuizen, M. Camacho, J. Heine, M. Vaidyanathan, L. Hall, R. Thatcher, and M. Silbiger, “MRI segmentation: methods and applications,” *Magnetic Resonance Imaging*, vol. 13, pp. 343-368, 1995.

[22] J. C. Bezdek, L. O. Hall, and L. P. Clarke, “Review of MR image segmentation techniques using pattern recognition,” *Medical Physics*, vol. 20, pp. 1033-1048, 1993.

[23] K. Fu and J. Mui, “A survey on image segmentation,” *Pattern Recognition*, vol. 13, pp. 3-16, 1981.

[24] N. R. Pal and S. K. Pal, “A review on image segmentation techniques,” *Pattern Recognition*, vol. 26, pp. 1277-1294, 1993.

[25] W. Pratt, *Digital Image Processing*, Second ed. New York: John Wiley &Sons, Inc., 1991.

[26] S. C. Zhu, T. S. Lee, and A. L. Yuille, “Region competition: unifying snakes, region growing, energy/Bayes/MDL for multi-band image segmentation,” in *Proceedings of the Fifth International Conference on Computer Vision*. Los Alamitos, CA, USA: IEEE Comput. Soc. Press, 1995, pp. 416-423.

[27] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, “Scale and the differential structure of images,” *Image and Vision Computing*, vol. 10, pp. 376-388, 1992.

[28] B. M. ter Haar Romeny, L. M. J. Florack, A. H. Salden, and M. A. Viergever, “Higher order differential structure of images,” *Image and Vision Computing*, vol. 12, pp. 317-325, 1994.

[29] R. Szeliski, D. Tonnesen, and D. Terzopoulos, “Modeling surfaces of arbitrary topology with dynamic particles,” in *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Comput. Soc. Press, 1993, pp. 82-87.

[30] W. C. Lin, E. C. K. Tsao, and C. T. Chen, “Constraint satisfaction neural networks for image segmentation,” *Pattern Recognition*, vol. 25, pp. 679-693, 1992.

[31] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.

[32] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: active contour models,” *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1987.

[33] J. Gauch, H. Pien, and J. Shah, “Hybrid boundary-based and region-based deformable models for biomedical image segmentation,” in *Proceedings of the SPIE Conference on Medical Imaging*, vol. 2299, 1994, pp. 72-83.

[34] G. Szekely, A. Kelemen, C. Brechbuhler, and G. Gerig, “Segmentation of 3D objects from MRI volume data using constrained elastic deformations of flexible Fourier surface models,” in *Proceedings of the Conference on Computer Vision, Virtual Reality and Robotics in Medicine*. Nice, France: Germany: Springer-Verlag, 1995, pp. 495-505.

[35] T. McInerney and D. Terzopoulos, “Medical image segmentation using topologically adaptable snakes,” in *Proceedings of the Conference on Computer Vision, Virtual Reality and Robotics in Medicine*. Nice, France: Germany: Springer-Verlag, 1995, pp. 92–101.

[36] G.-H. Chuang and C.-C. Kuo, “Wavelet descriptor of planar curves: theory and applications,” *IEEE Transactions on Image Processing*, vol. 5, pp. 56-70, 1996.

[37] G. Coleman and H. Andrews, “Image segmentation by clustering,” *Proceedings of the IEEE*, vol. 67, pp. 773-785, 1979.

[38] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Reading, Massachusetts: Addison-Wesley, 1994.

[39] K. Strasters and J. Gerbrands, “Three-dimensional image segmentation using a split, merge and group approach,” *Pattern Recognition Letters*, vol. 12, pp. 307-325, 1991.

[40] R. Dubes and A. Jain, “Random field models in image analysis,” *Journal of Applied Statistics*, vol. 16, pp. 131-164, 1989.

[41] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, 1984.

[42] A. Chakraborty and J. Duncan, "Integration of boundary finding and region based segmentation using game theory," in *Proceedings of the International Conference on Information Processing in Medical Imaging*, Y. Bizais, Ed. Nice, France: Kluwer Academic Publishers, the Netherlands, 1995, pp. 189-200.

[43] L. H. Staib and J. S. Duncan, "Model-based deformable surface finding for medical images," *IEEE Transactions on Medical Imaging*, vol. 15, no. 5, pp. 720-731, 1996.

[44] H. S. Stiehl, *On spatial image sequence understanding*: Technische Universitat Berlin, 1987.

[45] T. Binford, "Visual perception by computer," presented at IEEE Conference on Systems Science and Cybernetics, Miami, Florida, 1971.

[46] B. I. Soroka, "Understanding objects from slices: extracting generalized cylinder descriptions from serial sections," in *Moore School of Electrical Engineering*: University of Pennsylvania, 1979.

[47] U. Shani, "Three-dimensional geometric modeling for anatomical structures," *Proceedings of the Fifth Annual Symposium on Computer Applications in Medical Care*, pp. 571-575, 1981.

[48] K. Kitamura, J. M. Tobis, and J. Sklansky, "Estimating the 3D skeletons and transverse areas of coronary arteries from biplane angiograms," *IEEE Transactions on Medical Imaging*, vol. 7, no. 3, pp. 173-187, 1988.

[49] J. A. Fessler and A. Macovski, "Object-based 3-D reconstruction of arterial trees from magnetic resonance angiograms," *IEEE Transactions on Medical Imaging*, vol. 10, no. 1, pp. 25-39, 1991.

[50] J. Weszka and A. Rosenfeld, “Threshold evaluation techniques,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pp. 622-629, 1978.

[51] I. Abdou and W. Pratt, “Quantitative design and evaluation of enhancement/thresholding edge detectors,” *Proceedings of the IEEE*, vol. 67, pp. 753–763, 1979.

[52] S. Lee, S. Chung, and R. Park, “A comparative performance study of several global thresholding techniques for segmentation,” *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 171-190, 1990.

[53] V. Chalana and Y. Kim, “A methodology for evaluation of boundary detection algorithms on medical images,” *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 642-645, 1997.

[54] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Reading, Massachusetts: Addison-Wesley, 1979.

[55] W. Yasnoff, J. Mui, and J. Bacus, “Error measures for scene segmentation,” *Pattern Recognition*, vol. 9, pp. 217-231, 1977.

[56] C. Friedman and J. Wyatt, *Evaluation in Medical Informatics*. New York, NY: Springer, 1995.

[57] D. Bryant and D. Bouldin, “Evaluation of edge operators using relative and absolute grading,” in *Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*. Chicago, IL, 1979, pp. 138-145.

[58] J. Mitchell, S. Karlik, D. Lee, M. Eliasziw, G. Rice, and A. Fenster, “Evaluating the impact on operator performance of quantification algorithms,” in *Proceedings of the SPIE Conference on Medical Imaging*, vol. 2710, M. Loew, Ed. Newport Beach, Claifrnia: The International Society for Optical Engineering, 1996.

[59] J. Mitchell, S. Karlik, D. Lee, M. Eliasziw, G. Rice, and A. Fenster, “The variability of manual and computer assisted quantification of multiple sclerosis lesion volumes,” *Medical Physics*, vol. 23, pp. 85-97, 1996.

[60] S. Blumenfeld and G. Glover, “Spatial resolution in computed tomography,” in *Radiology of the Skull and Brain: Technical Aspects of Computed Tomography*, vol. 5, T. Newton and D. Potts, Eds. Saint Louis: Mosby, 1981, pp. 3918-3940.

[61] R. Greenes and J. Brinkley, “Radiology Systems,” in *Medical Informatics: Computer Applications in Health Care*, E. Shortliffe, L. Perreault, G. Wiederhold, and L. Fagan, Eds. Reading, Massachusetts: Addison-Wesley, 1990.

[62] F. Judy, “Evaluating computed tomography image quality,” in *Medical CT and Ultrasound: Current Technology and Application*, L. Goldman and J. Fowlkes, Eds. Madison, Wisconsin: Advanced Medical Publishing, 1995, pp. 359-377.

[63] K. Hanson, “Noise and contrast discrimination in computed tomography,” in *Radiology of the Skull and Brain: Technical Aspects of Computed Tomography*, vol. 5, T. Newton and D. Potts, Eds. Saint Louis: Mosby, 1981, pp. 3941-3955.

[64] J.-Y. Wang and F. Cohen, “Part II: 3-D object recognition and shape estimation from image contours using B-splines, shape invariant matching, and neural network,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 13-23, 1994.

[65] B. Scassellati, S. Alexopoulos, and M. Flickner, “Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgement,” IBM research division, San Jose, California RJ 9705 (84337), February 1994.

[66] F. Alt, “Digital pattern recognition by moments,” *Journal of the ACM*, vol. 9, pp. 240-58, 1962.

[67] A. Reeves, R. Prokop, S. Andrews, and F. Kuhl, “Three-dimensional shape analysis using moments and Fourier descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 937-943, 1988.

[68] D. Mumford, “Mathematical theories of shape: do they model perception?,” *Proceedings of the SPIE Conference on Geometric Methods in Computer Vision*, vol. 1570, The International Society for Optical Engineering, pp. 2-10, 1991.

[69] F. Kuhl and C. Giardina, “Elliptic Fourier features of a closed contour,” *Computer Graphics and Image Processing*, vol. 18, pp. 236–258, 1982.

[70] M. Unser, A. Aldroubi, and M. Eden, “A family of polynomial spline wavelet transforms,” *Signal Processing*, vol. 30, pp. 142-162, 1993.

[71] K. Zikan, “The Theory and Applications of Algebraic Metric Spaces,” in *Department of Operations Research*. Stanford: Stanford University, 1990.

[72] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, “An efficiently computable metric for comparing polygonal shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209-211, 1991.

[73] R. Fagin and L. Stockmeyer, “A Semi-Triangle-Inequality for Distances Between Shapes,” IBM Research Division, Almaden Research Center, San Jose, California 1996.

[74] D. Huttenlocher and W. Rucklidge, “A multiresolution technique for comparing images using the Hausdorff distance,” in *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1993.

[75] H. Alt and M. Godau, “Measuring the resemblance of polygonal Curves,” in *Proceedings of the 8th Annual Conference on Computational Geometry*. Berlin, Germany: ACM, 1992, pp. 102-108.

[76] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, 1990.

[77] G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni, “Trademark shapes description by sting-matching techniques,” *Pattern Recognition*, vol. 27, no. 8, pp. 1005-1018, 1994.

[78] W. Mendenhall and T. Sincich, *Statistics for Engineering and the Sciences*. New Jersey: Prentice-Hall, Inc., 1995.

[79] J. Neter, W. Wasserman, and M. H. Kutner, *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs*. Homewood, Ill: Irwin, 1990.

[80] S. Shiffman, G. Rubin, and S. Napel, “Semiautomated editing of computed tomography sections for visualization of vasculature,” in *Proceedings of the SPIE conference on Medical Imaging*, vol. 2707, Y. Kim, Ed. Newport Beach, California: The International Society for Optical Engineering, 1996.

[81] S. Shiffman, G. Rubin, and S. Napel, “Automated segmentation of blood vessels in CT angiograms via matching of isolabel contours,” in *Proceedings of the Scientific Assembly of the Radiological Society of North America*. Chicago, 1996.

[82] S. Shiffman, G. Rubin, and S. Napel, “Resolution of label ambiguity through matching of isolabel contours for segmentation of blood vessels,” Section on Medical Informatics, Stanford University, Stanford, CA SMI-96-0634, February, 1996.

[83] S. Shiffman, G. Rubin, and S. Napel, “Automated segmentation of blood vessels on CT angiograms via matching of isolabel contours,” in *Proceedings of the Scientific Assembly of the Radiological Society of North America*. Chicago, 1996.

[84] S. Shiffman, G. Rubin, and S. Napel, “Classifications of regions via an overlap criterion for segmentation of blood vessels in CT angiograms,” in *Proceedings of the Scientific Assembly of the Radiological Society of North America*. Chicago, 1997.

[85] S. Shiffman, R. Shifrin, C. Beaulieu, S. Heiss, G. Rubin, B. Daniel, and S. Napel, “Semiautomated blood-vessel segmentation from CT Angiograms: Comparison of human- to computer-generated results,” in *Proceedings of the Scientific Assembly of the Radiological Society of North America*. Chicago, 1998.

[86] S. Shiffman and S. Napel, “Object Segregation in Images,” Patent application, USA, January 1999.

[87] B. Zheng, Y. H. Chang, and D. Gur, “Computerized detection of masses in digitized mammograms using single image segmentation and a multilayer topographic feature analysis,” *Academic Radiology*, vol. 2, pp. 959-966, 1995.