

Washington University in St. Louis

## Washington University Open Scholarship

---

All Theses and Dissertations (ETDs)

---

8-21-2012

### Modeling Surfaces from Volume Data Using Nonparallel Contours

Ross Taylor Sowell

*Washington University in St. Louis*

Follow this and additional works at: <https://openscholarship.wustl.edu/etd>

---

#### Recommended Citation

Sowell, Ross Taylor, "Modeling Surfaces from Volume Data Using Nonparallel Contours" (2012). *All Theses and Dissertations (ETDs)*. 981.

<https://openscholarship.wustl.edu/etd/981>

This Dissertation is brought to you for free and open access by Washington University Open Scholarship. It has been accepted for inclusion in All Theses and Dissertations (ETDs) by an authorized administrator of Washington University Open Scholarship. For more information, please contact [digital@wumail.wustl.edu](mailto:digital@wumail.wustl.edu).

WASHINGTON UNIVERSITY IN ST. LOUIS  
School of Engineering and Applied Science  
Department of Computer Science and Engineering

Dissertation Examination Committee:  
Cindy Grimm, Chair  
Tao Ju  
Caitlin Kelleher  
Daniel A. Low  
Parag J. Parikh  
Weixiong Zhang

Modeling Surfaces from Volume Data Using Nonparallel Contours

by

Ross Taylor Sowell

A dissertation presented to the Graduate School of Arts and Sciences  
of Washington University in partial fulfillment of the  
requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2012  
Saint Louis, Missouri

copyright by  
Ross Taylor Sowell  
2012

## ABSTRACT OF THE DISSERTATION

Modeling Surfaces from Volume Data Using Nonparallel Contours

by

Ross Taylor Sowell

Doctor of Philosophy in Computer Science

Washington University in St. Louis, 2012

Research Advisor: Professor Cindy Grimm

Magnetic resonance imaging (MRI) and computed tomography (CT) scanners have long been used to produce three-dimensional (3D) samplings of anatomy elements for use in medical visualization and analysis. From such data sets, physicians often need to construct surfaces representing anatomical shapes in order to conduct treatment, such as irradiating a tumor. Traditionally, this is done through a time-consuming and error-prone process in which an experienced scientist or physician marks a series of parallel contours that outline the structures of interest. Recent advances in surface reconstruction algorithms have led to methods for reconstructing surfaces from nonparallel contours that could greatly reduce the manual component of this process. Despite these technological advances, the segmentation process has remained unchanged.

This dissertation takes the first steps toward bridging the gap between the new surface reconstruction technologies and bringing those methods to use in clinical practice. We develop VolumeViewer [68], a novel interface for modeling surfaces from volume data

by allowing the user to sketch contours on arbitrarily oriented cross-sections of the volume. We design the algorithms necessary to support nonparallel contouring, and we evaluate the system with medical professionals using actual patient data. In this way, we begin to understand how nonparallel contouring can aid the segmentation process and expose the challenges associated with a nonparallel contouring system in practice.

# Acknowledgments

First, I thank my advisor, Cindy Grimm, for her constant support and guidance throughout my tenure as a graduate student. During this time, she gave birth to two children, spent a year across the country on sabbatical, and yet always managed to be there when I needed input on my work. She continues to impress and inspire me as a scientist and a person.

I also express my gratitude to the other members of my committee: Tao Ju, Caitlin Kelleher, Daniel Low, Parag Parikh, and Weixiong Zhang. I thank them for challenging and even frustrating me. This document is better because of their careful critique.

I thank my collaborators at Washington University, the Washington University School of Medicine, and Cornell College: Lu Liu, Chris Trn, Michelle Vaughn, Christopher Abraham, Garima Gokhroo, and Kevin Johnson. Without their contributions, this work would not have been possible.

For six years I had the pleasure of going to work each day with some of the most interesting, intelligent, and friendly people that I have ever met. I am grateful to many past and present members of the Media and Machines Lab for making this so. I also thank my new colleagues at Cornell College, particularly Tony deLaubenfels and Leon Tabak, for welcoming me and creating a supportive environment that has enabled me to finish writing this document.

I owe a special token of gratitude to my undergraduate advisor, Lucia Dale, for plucking me out of an introductory programming course and getting me excited about her research in robotics. Without her inspiration and mentorship, I doubt that I would have ever found myself on this path. I can only hope that I will be able to return the favor to future generations of budding scientists.

Finally, and most importantly, I thank my family to whom I proudly dedicate this dissertation. It is their constant love and support that has allowed me to achieve anything. I thank my mother, Marilynn, my father, Joe Tom, and my sister, Chloe. I thank my son Benjamin for making me smile every day for the past sixteen months,

and I express my heartfelt thanks to my best friend and wife, Elizabeth, for going on this journey with me.

Ross Taylor Sowell

*Washington University in Saint Louis*  
*August 2012*

This work was partially supported by the National Science Foundation, under awards DEB-1053554, CCF-0702662, and CNS-0722328, and by the Washington University Imaging Sciences Pathway Fellowship.

To my family



# Contents

<b>Abstract</b> . . . . .	<b>ii</b>
<b>Acknowledgments</b> . . . . .	<b>iv</b>
<b>List of Tables</b> . . . . .	<b>ix</b>
<b>List of Figures</b> . . . . .	<b>x</b>
<b>List of Abbreviations</b> . . . . .	<b>x</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Motivation . . . . .	1
1.2 The segmentation problem . . . . .	2
1.3 Background on automated approaches . . . . .	4
1.4 Organization of the dissertation . . . . .	5
<b>2 Conceptual framework</b> . . . . .	<b>7</b>
2.1 Evaluation criteria . . . . .	8
2.1.1 Accuracy . . . . .	8
2.1.2 Consistency . . . . .	10
2.1.3 Efficiency . . . . .	10
2.2 Data set properties . . . . .	11
2.2.1 Boundary strength . . . . .	11
2.2.2 Structure size . . . . .	11
2.2.3 Structure shape and orientation . . . . .	12
2.3 Designing a contouring system . . . . .	13
2.3.1 Factors that influence accuracy and consistency . . . . .	13
2.3.2 Factors that influence efficiency . . . . .	15
2.3.3 Working hypotheses . . . . .	16
2.4 Analysis of parallel contouring . . . . .	17
2.5 Analysis of nonparallel contouring . . . . .	19
2.6 Chapter summary . . . . .	20
<b>3 VolumeViewer: An interactive tool for fitting surfaces to volume data</b> . . . . .	<b>21</b>
3.1 Introduction . . . . .	21
3.2 Related work . . . . .	22

3.2.1	Closed source systems . . . . .	22
3.2.2	Open source sytems . . . . .	23
3.3	User interaction . . . . .	24
3.4	Contouring protocols . . . . .	27
3.5	Visualization cues . . . . .	31
3.6	Implementation and distribution . . . . .	34
3.7	Chapter summary . . . . .	36
<b>4</b>	<b>Algorithms for nonparallel contouring . . . . .</b>	<b>37</b>
4.1	Introduction . . . . .	37
4.2	Related work . . . . .	37
4.3	Ensuring contour consistency . . . . .	38
4.3.1	Formal definition of consistency . . . . .	39
4.3.2	Contour validation and snapping . . . . .	40
4.3.3	Validate-as-you-go process . . . . .	45
4.4	Supporting partial contouring . . . . .	46
4.4.1	Extending Hermite radial basis function implicits . . . . .	47
4.5	Chapter summary . . . . .	49
<b>5</b>	<b>Evaluation . . . . .</b>	<b>50</b>
5.1	Introduction . . . . .	50
5.2	Related work . . . . .	51
5.3	User studies on the feasibility of nonparallel contouring . . . . .	52
5.3.1	Method . . . . .	52
5.3.2	Results . . . . .	57
5.3.3	Discussion . . . . .	61
5.4	System evaluation . . . . .	64
5.4.1	Method . . . . .	65
5.4.2	Results . . . . .	69
5.4.3	Discussion . . . . .	71
5.5	Chapter summary . . . . .	73
<b>6</b>	<b>Conclusion and future work . . . . .</b>	<b>74</b>
6.1	Future work . . . . .	75
6.1.1	Plane selection and protocol authoring . . . . .	75
6.1.2	Path to clinical adoption . . . . .	80
6.1.3	Error estimation for segmentation review . . . . .	81
6.1.4	Extension to other applications and citizen science . . . . .	81
	<b>References . . . . .</b>	<b>83</b>
	<b>Vita . . . . .</b>	<b>91</b>

# List of Tables

5.1	P-values from Wilcoxon signed ranks tests on the contour data . . . .	62
-----	---	----

# List of Figures

1.1	Parallel versus nonparallel contouring . . . . .	2
1.2	Impact of spatial resolution on image quality . . . . .	3
1.3	Low image contrast makes automatic segmentation difficult . . . . .	4
2.1	An example of differing boundary strengths . . . . .	12
3.1	The VolumeViewer interface . . . . .	25
3.2	An example protocol for contouring the prostate . . . . .	28
3.3	Contours and surface model generated by example prostate protocol . . . . .	29
3.4	Reference images . . . . .	29
3.5	Rendering contours as thin ribbons . . . . .	32
3.6	Coloring the image plane . . . . .	33
3.7	A screenshot of the VolumeViewer website . . . . .	35
4.1	An example of inconsistent contours . . . . .	39
4.2	Contour snapping . . . . .	43
4.3	Motivation for partial contouring . . . . .	46
4.4	Stage T4 lung tumors . . . . .	47
4.5	Surface reconstruction from partial contours using HRBF . . . . .	48
5.1	User study setup . . . . .	54
5.2	Inter-observer variability of experts . . . . .	58
5.3	Intra-observer variability of experts . . . . .	59
5.4	Inter-observer variability of novices with reference images . . . . .	60
5.5	Mean distances and percentage overlap between user drawn contours . . . . .	61
5.6	Example images from prostate and liver data sets . . . . .	66
5.7	Comparison of user contours for parallel and nonparallel liver cases . . . . .	67
5.8	Efficiency and consistency of user segmentations . . . . .	70

# List of Abbreviations

- **MRI:** Magnetic resonance imaging
- **CT:** Computed tomography
- **3D:** Three-dimensional
- **ST:** Slice thickness
- **FOV:** Field of view
- **EMD:** Earth mover's distance
- **IMRT:** Intensity modulated radiation therapy
- **MIPAV:** Medical Image Processing, Analysis, and Visualization
- **API:** Application programming interface
- **ITK:** Insight Segmentation and Registration Toolkit
- **VTK:** Visualization Toolkit
- **2D:** Two-dimensional
- **OpenGL:** Open Graphics Library
- **FLTK:** Fast Light Toolkit
- **GUI:** Graphical user interface
- **HRBF:** Hermite radial basis function
- **GTV:** Gross tumor volume
- **MPR:** Multiplanar reformation

# Chapter 1

## Introduction

### 1.1 Motivation

Cancer is the leading cause of death in economically developed countries. About 12.7 million cancer cases and 7.6 million cancer deaths are estimated to have occurred in 2008 [37]. One of the primary forms of treatment for cancer patients is radiation therapy, in which ionizing radiation is applied to shrink tumors and kill cancer cells. This process requires several steps to be performed prior to the start of treatment. First, the patient is imaged, usually by performing a CT scan of the area to be treated. From the CT scan, segmentation is performed, in which 3D models of the tumors as well as healthy organs and tissue are built. These models are then used to design a treatment plan that determines the size, shape, and intensity of the radiation beams to be used during treatment. Such a plan is reviewed by the radiation oncologist and physics staff, and only then can treatment begin.

This process typically takes a week or more from the time a patient is imaged to the time the first radiation dose is delivered. This is unfortunate because during that time organs can shift and tumors can change size and shape, causing treatment plans to become outdated. The segmentation process is one of the main bottlenecks in the pipeline. It is a time-consuming and manually intensive process that can itself be a source of error, as any inaccuracies in the segmentation will lead to inaccuracies in the treatment plan and dose distribution. In this dissertation, we aim to improve the overall accuracy and efficiency of the segmentation process, which can ultimately lead to reduced morbidity and improved patient care.

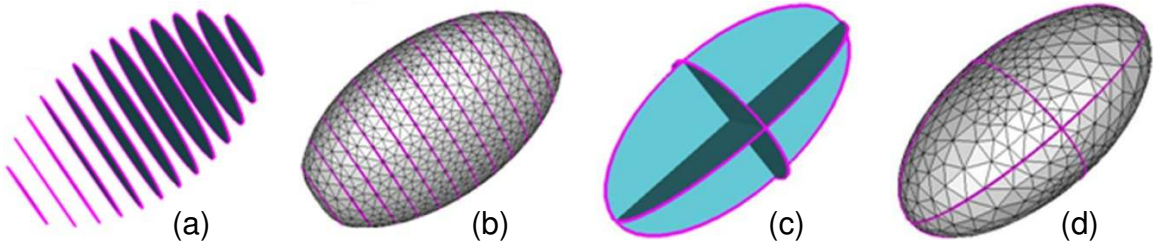


Figure 1.1: Parallel versus nonparallel contouring. The contours are defined from a mathematically exact ellipsoid. 13 parallel contours (a) are used to reconstruct a surface model of the ellipsoid (b). 3 nonparallel contours (c) are used to reconstruct a surface model of the ellipsoid (d). The surfaces look similar, but the ends of the ellipsoid are more accurately defined using the nonparallel contours. We anticipate that this will be indicative of many clinical structures, and that the accuracy and efficiency of the contouring process can be improved by using a small set of nonparallel contours rather than many parallel contours.

## 1.2 The segmentation problem

In radiation therapy, volumetric CT scans are used to determine the location and shape of tumors, target volumes, and normal organs. The CT data is a 3D scalar field, representing the photon attenuation coefficients of the patient and other materials in the imaging volume. The segmentation problem is to partition this image data into regions that are located inside and outside the structures of interest. The current state of the art segmentation process presents the treatment planner with a set of parallel CT slices and has the planner outline the structure boundaries, resulting in a series of parallel contours (see Figure 1.1a,b). This is a time-consuming and error-prone process [21, 29, 75]. But why is the user limited to marking contours on *parallel* cross sections?

Early CT scanners had limited spatial resolution along the sampling axis that was much coarser than the in-plane resolution, making it impractical to view images in nonparallel planes (see Figure 1.2). Thus, the parallel contouring process was arrived at not by intelligent design, but rather by historical consequence. Modern CT scanners now support isotropic or nearly isotropic sub-millimeter voxel resolution [8], making the quality of out-of-plane images comparable to those acquired along the sampling axis. Additionally, advances in surface reconstruction algorithms have led to methods [7, 44] for reconstructing surfaces from nonparallel contours that could

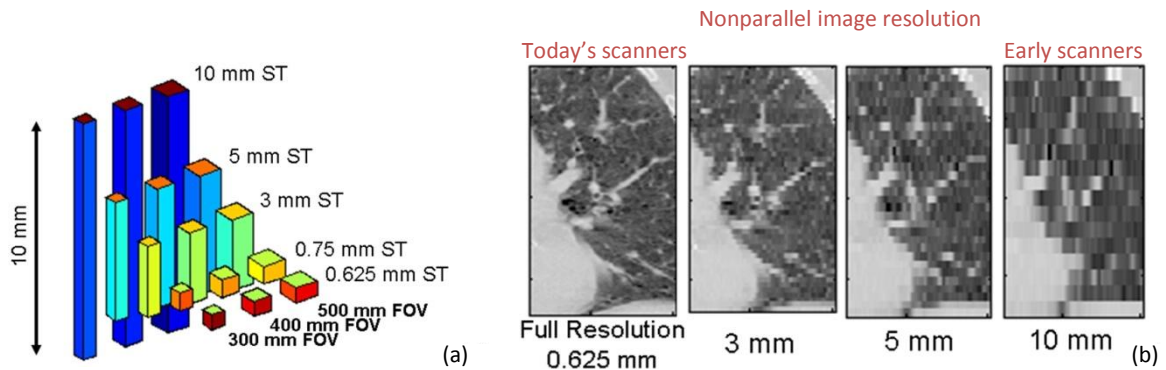


Figure 1.2: Impact of spatial resolution on image quality. (a) Example of the relative spatial resolution for five slice thicknesses (ST). The 0.625 mm and 0.75 mm thick slices are the intrinsic resolutions of the Phillips Big Bore and Brilliance 64, respectively and are representative of modern multislice CT scanners. The 3 mm, 5 mm, and 10 mm thicknesses are representative of reconstructions often used for treatment planning. The transverse resolution has 512 x 512 voxels and 300 mm, 400 mm, and 500 mm fields of view (FOV) are shown. b) Example of a coronal CT scan subset of the left lung acquired at 0.625 mm longitudinal resolution (0.98 mm in-plane resolution) and reconstructed at 3 mm, 5 mm, and 10 mm resolutions to show the degradation of the image content in the reduced resolution reconstructions.

greatly reduce the manual component of this process (see Figure 1.1c,d). Despite these technological advances, the segmentation process has remained unchanged.

The goal of this dissertation is to conduct the research necessary to begin to bring the new surface reconstruction technologies into routine clinical practice. This is a nontrivial task, as the off-the-shelf reconstruction algorithms are not designed to handle raw user input, and there are many human factors that must be considered in order to make this work in practice. We develop algorithms that make surface reconstruction from user-drawn nonparallel contours feasible, and we work with a group of radiation oncologists to develop VolumeViewer, an interactive nonparallel contouring system that is robust and easy to use. We evaluate the system with the assistance of medical professionals using patient data, comparing the consistency and efficiency of the segmentation process against traditional systems. In this way, we begin to understand how nonparallel contouring can aid the segmentation process and expose the challenges associated with a nonparallel contouring system in practice.



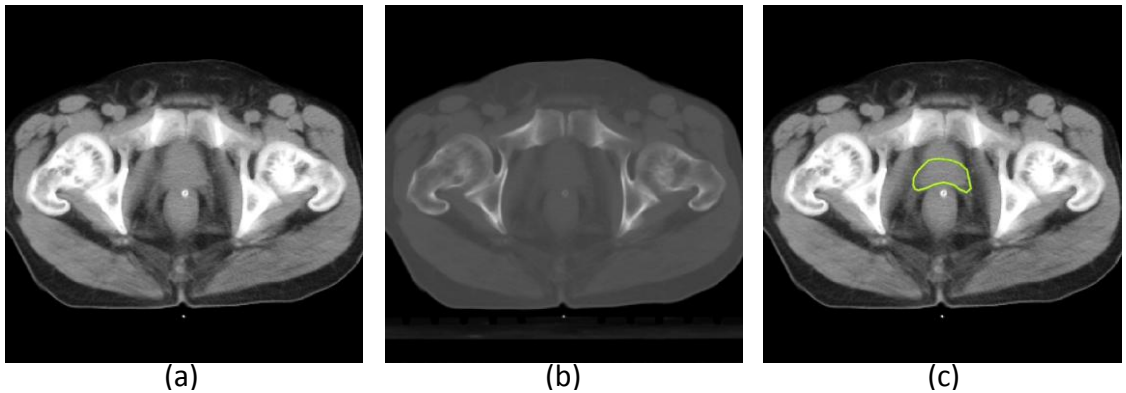


Figure 1.3: Low image contrast makes automatic segmentation difficult. (a) An image from a CT scan of a male pelvis. (b) An image corresponding to the magnitude of the gradient of the image in (a), highlighting the regions of sharp contrast. (c) The yellow contour outlines the location of the prostate in this image. Such weak boundary strength is common for many structures of interest in the CT scans used for radiotherapy, making automatic segmentation a difficult problem. Such segmentation tasks are better performed manually by a trained human observer.

### 1.3 Background on automated approaches

Why do treatment planners not use automatic processes? Many approaches have been proposed for automatic volume segmentation, including thresholding, k-means clustering, deformable models, watershed segmentation, graph cut algorithms, and level-set methods [56]. These methods typically depend on strong gradient information in the image data, and in some cases, a template model that can then be deformed to fit the image data. Unfortunately, there are still many cases where template models are not available, such as for irregularly shaped structures like tumors. The image data is often noisy and does not have sufficient contrast for automatic methods (see Figure 1.3). Even in cases where automated methods are suitable, physicians must still manually review the results before they are used to treat a patient.

Due to the limitations of auto-segmentation approaches, several interactive techniques have been developed to allow the user to specify input for semi-automatic segmentation. These methods allow the user to specify constraint points by roughly sketching foreground and background regions on a cross-sectional plane [73] or the volume-rendered image [82] or by sketching a contour of the region of interest on the volume

rendered image [53]. These constraint points are then given as input to a segmentation algorithm that returns a set of voxels inside the region of interest. A surface can then be generated by an isosurface algorithm such as Marching Cubes [45]. In more recent work, Poon et al. [59] describe a 3D Livewire algorithm that allows the user to provide a set of planar contours as input, which it uses to seed unvisited planes in order to generate a dense set of 3D contours.

These techniques allow the user to provide input that guides the auto-segmentation process, but they are still ultimately dependent on the underlying auto-segmentation algorithm and will therefore have difficulties in cases where the structure does not follow a clear isocontour. For this reason, manual segmentation on a series of parallel cross sections remains the standard practice for segmentation in radiation therapy. Furthermore, even as automated techniques improve, the resulting segmentations will still need to be reviewed manually before treatment is conducted, and we believe that the proposed techniques for improving segmentation can be applied to the review process as well. Therefore, our focus is on improving the manual segmentation process by allowing the user to contour on nonparallel planes.

## 1.4 Organization of the dissertation

The remainder of this document is organized into five chapters:

- **Chapter 2** provides a discussion of the system of concepts, assumptions, expectations, and theories that underpins our work. We describe the criteria that can be used to evaluate segmentation techniques and the key properties of our data sets that affect the segmentation task. From there, we identify factors that will influence the effectiveness of a contouring system and make several working hypotheses about how a contouring system should be designed. We then analyze parallel and nonparallel contouring with respect to these hypotheses. This discussion is used as a conceptual framework to guide and connect the different aspects of our research presented here.

- In **Chapter 3**, we present VolumeViewer, a novel sketch-based interface that allows the user to generate a surface from just a few contours drawn on arbitrarily oriented planes. The user can then review and edit the model globally and interactively, rather than marking many parallel contours on a slice-by-slice basis. We design structure-specific contouring protocols and a number of visualization cues that aid the user in building consistent surfaces efficiently.
- **Chapter 4** describes several algorithmic developments that are required for nonparallel contouring to be feasible in practice. In order to reconstruct a surface from nonparallel contours using existing techniques [7, 44], the input contours must be consistent. We describe a process for validating a set of user-drawn contours and develop tools to aid the user in detecting and fixing any contour inconsistencies. We also extend the work of [46] to provide support for partial contours. This allows the user to only provide input where it is necessary and where the user is confident in the input being provided, rather than requiring closed contours on each plane.
- In **Chapter 5**, we present the results of two evaluations. The first study shows that integrating a collection of reference images into the interface can reduce the user variation in nonparallel contouring. The second study evaluates our end-to-end system and offers insights into the situations in which it will provide the most utility.
- **Chapter 6** concludes with a summary of the dissertation and outlines several avenues of future research based on the presented work.

# Chapter 2

## Conceptual framework

We make the argument that the segmentation process can be improved through the use of nonparallel contours. This chapter sets the stage for this argument by answering the following fundamental questions:

- What are the criteria used to evaluate the effectiveness of a segmentation method?
- What are the essential properties of the data that affect the segmentation task, and how do they vary from one data set to the next?
- What are the factors that influence the effectiveness of a segmentation method, and what are the implications for the design of a contouring system?

The answers to these questions constitute a conceptual framework that informs our research. We conclude this chapter with an analysis of the advantages and disadvantages of both parallel and nonparallel contouring with respect to this framework.

It should be emphasized that this was not arrived at *a priori* as a result of surveying previous work (although some of it certainly was, and citations are included where that is the case). Instead, this is largely a product of our own investigation. It is the system of concepts, assumptions, expectations, and theories that we have developed by studying the contouring process. It includes a set of working hypotheses that we have constructed from our research, but that have yet to be individually tested. We will refer back to these hypotheses when we use them to make design decisions and when our experiments provide evidence to support them. Though really a result of

our research, we present this discussion here because it helps to put the rest of our work into context, and we believe it will prove valuable to anyone that is beginning work in this area.

## 2.1 Evaluation criteria

In order to assess the merit of any proposed changes to the segmentation process, we must first define how a segmentation method can be evaluated. Common criteria for evaluating segmentation methods are accuracy, consistency, and efficiency [52]. Here we define each of these terms and describe how they can be measured.

### 2.1.1 Accuracy

Accuracy is the degree to which the resulting segmentation matches the truth. If the ground truth is known, the resulting segmentation can be compared against the ground truth using the following measures:

- **Dice's coefficient** [19] can be used to measure the volume of the region shared by two segmentations as a percentage of the total volume occupied by both segmentations. It is calculated as twice the ratio of the number of voxels contained in the intersection of the two segmentations to the total number of voxels contained in each segmentation.
- **Mean distance** [14] measures the difference between two segmentations in terms of the distances between point samples on the surfaces of both segmentations. For each point on surface A, the distance to surface B is defined as the Euclidean distance to the nearest point on surface B. The mean distance from surface A to surface B is the average of these distances. Note that this is a one-sided distance in that the mean distance from surface A to surface B may be different from that of surface B to surface A. A two-sided mean distance can be found by taking the average of all the distances from surface A to surface B and surface B to surface A. If the maximum of all distances is taken, rather than the average, then this is known as the Hausdorff distance.

- The **Mallows distance** or **earth mover’s distance** (EMD) [47, 43] can be used as a measure of the cost to correct the segmentation. Intuitively, each voxel in the segmentation is thought of as a parcel of dirt. Two different segmentations are then considered as two different ways of piling up the dirt. The EMD is then the minimum cost of turning one pile into the other, which is the amount of dirt that needs to be moved times the distance it must be moved.
- The **Procrustes distance** [39] is a measure of the difference in shape. It is arrived at by first superimposing the two segmentations by finding an optimal translation, rotation, and scale from one segmentation to the other. The distance is then computed as the square root of the sum of the squared distances between corresponding points.

Unfortunately, we do not have ground truth for real patient data. We can simulate real patient data by generating synthetic image data (for which we know the ground truth) or by imaging a cadaver (for which ground truth can be determined by dissection). These options can give us a rough estimation of accuracy, but the validity of that estimation depends on how closely the synthetic or cadaver image data compares to real patient data.

In the absence of ground truth, we can gain some measure of the accuracy with real patient data in two ways. The first is to compare to a gold standard, which would be generated and reviewed by a panel of experts. The drawback of this approach is that the gold standard will vary depending on the composition of the panel of experts and the segmentation method used to generate the gold standard.

A second option is to conduct a subjective evaluation, in which experts would review and rank the results generated by other users. Because of its subjective nature, such an evaluation will again vary depending on the group of experts doing the ranking.

Ideally, we would compute an average, “consensus” segmentation and then measure the deviation from the consensus. However, computing the average and standard deviation of a set of contours or surfaces remains an open problem. One approach is the STAPLE algorithm [77] that considers a collection of segmentations and computes a probabilistic estimate of the true segmentation and a measure of the performance level represented by each segmentation. When prior information regarding the

expected anatomy is available, STAPLE provides estimates of performance parameters accounting for this external standard reference. When such information is not available, STAPLE provides estimates of performance parameters with respect to a weighted combination of the input segmentations and the appropriate weighting is computed automatically. Therefore, in situations where there is no ground truth and no reliable reference standard, consistency is commonly relied upon as a measure of segmentation quality.

### **2.1.2 Consistency**

Consistency, or precision, is the extent to which the same segmentation is produced during separate segmentation sessions by the same user or by different users. Consistency is measured in terms of inter-observer and intra-observer variability. For inter-observer variability, the same data set is segmented by several users and the resulting segmentations are compared. For intra-observer variability, the same data set is segmented several times by one user and the resulting segmentations are compared. Individual segmentations can be compared to one another using the same metrics listed above for comparing a segmentation to the ground truth, and we can take the mean difference between segmentations in a given population as a measure of consistency.

### **2.1.3 Efficiency**

Efficiency is the extent to which time and effort are well used during the segmentation process. One simple measure of efficiency is the total elapsed time from the beginning of the segmentation process until the final segmentation is produced. This is straightforward to evaluate, but it hides any wasted effort. What we really care about with efficiency is the ratio of user input to output. Therefore, a more meaningful measure is the time on task versus the accuracy of the segmentation generated from the input created thus far.

## 2.2 Data set properties

Now that we have defined the criteria used to evaluate a segmentation method, we are ready to consider how the effectiveness (in terms of accuracy, consistency, and efficiency) of a segmentation method is affected by the properties of different data sets. The effectiveness of a segmentation method is dependent on the data set to which it is applied. One segmentation method may be particularly effective for one type of data set, while being entirely useless for many others. In this section, we discuss the essential properties of the data sets that can affect the segmentation task. We describe the properties that are common in radiation treatment planning and how these properties can vary from one data set to the next.

### 2.2.1 Boundary strength

The degree to which the boundaries of the structure to be segmented are visible varies greatly from structure to structure and data set to data set. In the ideal scenario, the boundaries of the structure to be segmented would be crystal clear throughout the volume. In this case, the segmentation problem would be trivial. Automatic segmentation techniques would work satisfactorily every time.

In real-world data sets, however, the boundary strength of the structure to be segmented can span the spectrum from very strong to very weak. For some cases, the boundary is strong enough throughout the volume that automatic segmentation is a valid option. In other cases, portions of the boundary may be very strong while other portions are weak. In still other cases, the boundary is so weak that it is nearly nonexistent, and the user must rely on the visibility of other landmarks in the volume to delineate the structure (see Figure 2.1).

### 2.2.2 Structure size

The size of the structures to be segmented can vary greatly. For example, the average liver is 21 x 17 x 12 cm, while the average prostate is 4 x 2 x 3 cm. This translates into a large difference in the number of acquired slices in which the structure appears.



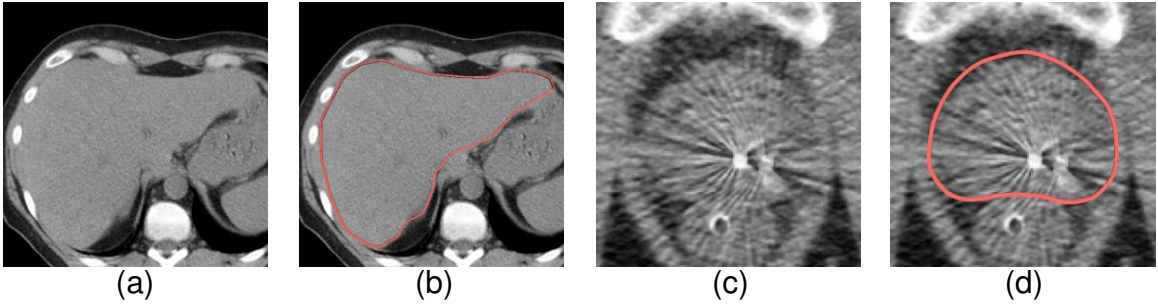


Figure 2.1: An example of differing boundary strengths. Example images are shown from a liver data set (a) together with its contour (b), as well as a prostate data set (c) together with its contour (d). We expect more user-variation in drawing the prostate contour, due to the weaker boundary.

Assuming a slice thickness of 1 mm, the average liver would appear in 120 slices, while the average prostate would appear in 30 slices. Therefore, there are many more parallel planes that need to be contoured for the liver than the prostate, and the length of the contour to be drawn on each plane will also be much larger for the liver than the prostate.

### 2.2.3 Structure shape and orientation

The shape of each anatomical structure also varies. We have spherical shapes, cylindrical shapes, tubular shapes, curved shapes, shapes with bumps, shapes with bulges, etc. Each of these 3D structures has its own orientation in the human body and, more importantly, an orientation with respect to the image plane. The combination of the shape of the 3D structure and the orientation of that structure with respect to the image plane determines what 2D shapes appear in the images that will be contoured. Intuitively, more complex shapes will be more difficult to contour than simpler ones.

We make a distinction between the local complexity of individual 2D contours and the global complexity of the 3D structure. A single slice through a tubular structure is simple, locally, whereas a slice through a structure like the pelvis, for example, could be much more complicated. However, the tubular structure may still be globally complex. The colon, for example, is more complex than the esophagus because it twists and turns in space.

## 2.3 Designing a contouring system

Having defined our evaluation criteria and the essential properties of the data sets, we would like to design a system for manual segmentation that will be as effective as possible. Here, we list the factors that will influence the effectiveness of a manual contouring system, and then, based on these factors, we make several hypotheses about how a contouring system should be designed.

### 2.3.1 Factors that influence accuracy and consistency

We group the factors that influence accuracy and consistency into three categories: differences in judgment, differences in performing the segmentation, and differences in the computational component of the tool. We describe these factors in terms of their impact on consistency, but it should be noted that the same set of factors apply to accuracy. In the discussion below, the differences between user segmentations could be substituted for the difference between a user segmentation and the ground truth.

**Differences in judgment:** Differences in judgment occur when one user considers the boundary of the structure to be located at a different position than another user. Factors that contribute to differences in judgment include:

- **Boundary strength** A weaker boundary is more difficult to delineate and therefore more subjective (see Figure 2.1). Weaker boundaries cause users to be more likely to disagree. In this case, users will often draw the same general shape, but the local variance along the contour will be higher.
- **A priori knowledge** The user relies on his or her prior knowledge and training to perform the segmentation. Intuitively, we expect experienced users to be more consistent with their segmentations than novices, but even amongst experienced users, their knowledge will vary. For example, when contouring a structure with an inner and outer wall, some users may contour the inner wall while others will contour the outer wall. In another example, an international proposal for the delineation of elective neck regions was developed and

published, but large variations persisted due to many users being unaware of the published guidelines [61]. In this case, users disagreed on which parts of the neck (high-, mid- and lower-jugular, and posterior) should be irradiated, as well as on the size and shape of the regions selected to treat similar lymph node regions [50]. A recent study on the segmentation of the female pelvic floor found that while users agreed on the location of organs, they demonstrated poor agreement at defining the structural boundaries [32].

**Differences in performing the segmentation:** Even when users agree on the location of the structure and are trying to mark the same boundary, errors may occur that are the result of differences in performing the segmentation. Factors that contribute to differences in performing the segmentation are:

- **Shape complexity** A more complex shape is more difficult to segment. The user must carefully control the mouse and use more mouse clicks in order to capture the fine details of a complex shape, increasing the chance of user error. For example, consider the case of marking a contour that is a perfect circle versus a bumpy circle. We expect more user-variation in the case of the bumpy circle due to the difficulty in correctly clicking on each of the bumps.
- **Simplification** Users may disagree on how much to simplify a contour. Consider again the bumpy circle. One user may choose to carefully outline the fine details of each bump, while another user may choose to ignore those details and simply draw a smooth circle that circumscribes the shape.
- **User fatigue** A tired user is more likely to be careless and make mistakes. The fatigued user is likely to use fewer clicks than is necessary to accurately mark the contour.
- **Mental integration** Differences in how users perceive the impact of their contours on the final shape will result in differences in the contours that they draw. For example, the naive user will treat each contour individually, while an expert may ensure that their contour is consistent with adjacent contours.

**Differences in the computational component of the tool:** Differences in the computational component of the tool are those differences that are unrelated to the user input. Instead, these differences occur due to how the software system uses the input contours to produce a 3D surface. The following factors should be considered:

- **Surface reconstruction algorithm** The choice of the algorithm for generating a surface from the input contours makes a difference. All algorithms should produce a surface that partitions space into inside and outside regions and interpolates the input contours, but how the contours get interpolated will vary from one algorithm to the next.
- **Algorithm parameters** Even if the same reconstruction algorithm is used, such algorithms often have input parameters. For example, in the algorithm by Liu et al. [44], input parameters control the degree of mesh fairing and refinement. Changing the parameters will alter the results.
- **Contour filtering** For sketch-based interfaces, filtering noise from the raw user input is a common operation [20, 16, 35, 80]. The algorithm used to filter the contours will have an effect on the input, and therefore the resulting segmentation as well.

### 2.3.2 Factors that influence efficiency

Factors that affect the efficiency of the segmentation process include:

- **Number of planes** There is an overhead associated with each additional plane that a user must contour. The user must move through the data set to the plane's location and then determine the location of the boundary on the new plane.
- **Contour length** Longer contours require more mouse clicks and therefore take more time to draw.
- **Shape complexity** This includes both the complexity of the 2D contours and of the 3D structure. More complex contours require more mouse clicks to capture the details of the boundary and therefore take more time to segment.

Additionally, for more complex 3D structures, additional planes will need to be contoured in order to capture the shape features, and it will take longer for the user to determine if the generated segmentation matches that 3D shape.

- **Boundary strength** Weaker boundaries require more time for the user to determine where the contour should be drawn.

### 2.3.3 Working hypotheses

Now that we have described the factors that influence the effectiveness of a manual contouring system, we are ready to make some hypotheses regarding the design of a contouring system. These are *working* hypotheses in the sense that, while not unfounded, they have not been individually and thoroughly tested. We will refer back to these when we use them to inform design decisions and when our experiments provide evidence that supports them. Following the list of hypotheses, we provide an analysis of the advantages and disadvantages of both parallel and nonparallel contouring with respect to these hypotheses.

**Hypothesis (H1):** In order to be as efficient as possible, each contour added should increase the accuracy of the segmentation as much as possible. This contour is likely to be the one that maximizes the area enclosed as well as passes through regions where the segmentation differs the most from the ground truth.

**Corollary:** For structures that have bumps or bulges, more contours will be required in those regions to accurately capture the feature. The contour that increases the accuracy of the feature as much as possible is likely to be the one that maximizes the area enclosed as well as passes through regions where the reconstructed feature differs the most from the true feature shape.

**Hypothesis (H2):** For structures that are cylindrical, tubular, or curved shapes with a simple cross section, planes that are perpendicular to the medial axis of the structure will result in the simplest 2D shapes to be contoured.

**Hypothesis (H3):** For data sets where the strength of the image gradient varies throughout the volume, planes that are approximately orthogonal to the gradient

field and pass through as many large magnitude gradients as possible will produce the strongest boundaries.

**Hypothesis (H4):** Users will be more accurate, consistent, and efficient when contouring on planes that are familiar to them based on their prior knowledge and training.

**Hypothesis (H5):** When transitioning from one plane to the next, moving the plane continuously through the volume data is more effective than jumping from one discrete viewpoint to the next.

**Hypothesis (H6):** When transitioning from one plane to the next, movements that keep the structure centered and the same size on the screen are more effective than those that do not.

**Hypothesis (H7):** Users produce segmentations that are more accurate and consistent when they can see the surface model and how it changes as they add each additional contour.

## 2.4 Analysis of parallel contouring

**Advantages of parallel contouring** The parallel contouring system offers numerous advantages with respect to these hypotheses, and perhaps none larger than the user familiarity with these planes (H4). Contouring on a series of parallel (transverse) planes is the standard of practice in radiation therapy. Users are trained to contour on these planes, and have more experience looking at the anatomy in this view than any other. Moreover, many structures in the human body are oriented in such a way that the long axis of the structure is perpendicular to the transverse image plane (H2).

Parallel contouring also enjoys advantages due to its simple navigation (H5,H6). Moving from one parallel slice to the next corresponds to moving one step forward and backward along the axis from the patient's head to feet. The size and position of a structure of interest on the screen changes little from one plane to the next. After

a set of contours have been drawn, paging back and forth quickly through the slices can give the user a rough approximation of what the surface will look like (H7).

**Disadvantages of parallel contouring** The largest room for improvement in the current parallel contouring system is in terms of its efficiency (H1). When adding an additional contour to a set of parallel contours, adding one more parallel contour on the next slice is unlikely to be the contour that increases the accuracy of the segmentation as much as possible. Consider the ellipsoid from Figure 1.1. Adding one more parallel contour between two other parallel contours will do little to improve the accuracy of the segmentation in comparison to the improvement that could be had by adding one nonparallel contour.

While many structures in the body (e.g., bones) are oriented in such a way that the long axis of the structure is perpendicular to the imaging plane, many other structures (e.g., cervix) are not axis-aligned or are curved (e.g., spinal cord). For these structures, parallel contouring will not result in the simplest 2D shapes to be contoured (H2). Similarly, for many data sets the standard parallel planes may not produce the strongest possible boundaries (H3). The boundaries for a portion of the structure may be clearer in another view, but the user is restricted to contouring on the parallel planes.

There is a growing body of literature on systems such as Teddy [34], ShapeShop [63], SmoothSketch [38], and FiberMesh [49] that allow users to create 3D models by sketching in 2D. None of these systems restrict the user to drawing sketches on parallel planes. To do so would make the task of the artist much more difficult and less efficient (H1). Now that we are able to view the image data in nonparallel planes without a loss of image resolution, there is no reason that the users of medical image segmentation software should be constrained to contouring on parallel planes.

## 2.5 Analysis of nonparallel contouring

In this dissertation, we address the disadvantages of parallel contouring by developing a nonparallel contouring system. We discuss the potential advantages and disadvantages of such a system in this section.

**Advantages of nonparallel contouring** The primary advantage of nonparallel contouring is that the user can contour on any plane. As discussed above, the contour that increases the accuracy of a segmentation as much as possible is often *not* a contour parallel to an existing one. Nonparallel contouring allows the user to potentially add the contour that does increase the accuracy of a segmentation as much as possible (H1). Similarly, in situations where the structure or the gradient field is not aligned to the transverse imaging axis, planes can be selected for contouring that result in the simplest 2D shapes (H2) and strongest boundaries (H3). Thus, nonparallel contouring has the potential to improve the segmentation process by directly addressing the disadvantages of a system that relies solely on parallel contouring.

**Disadvantages of nonparallel contouring** Nonparallel contouring is not without its disadvantages, which we view as the primary challenges to our goal. One major drawback of nonparallel contouring is the lack of user familiarity with these planes (H4). Users may be more familiar with some planes than others, based on their training in dissection of the anatomy for example, but users have not been trained to contour in these views in the same way as they have the parallel planes.

Another problem with nonparallel contouring is that moving from one nonparallel plane to the next does not correspond to a simple step forward or backward within the image data. It is a challenge to move smoothly from one plane to the next (H5) while helping the user maintain her point of reference (H6). Moreover, when given the choice to contour on any plane, the user may waste time in determining which plane to contour next.



## 2.6 Chapter summary

In this chapter we developed a conceptual framework for reasoning about the segmentation process that will be used throughout this dissertation. We discussed evaluation criteria for segmentation methods and the key properties of our data sets. We explored the factors that influence the effectiveness of a contouring system and made several working hypotheses about the design of such a system. We then analyzed the advantages and disadvantages of both parallel and nonparallel contouring with respect to these hypotheses. We believe that a nonparallel contouring system has the potential to make significant improvements to the accuracy and efficiency of the contouring process. However, in shifting to such a system, we face the challenge of losing many of the advantages inherent to parallel contouring. Our aim is to develop a system that supports nonparallel contouring while taking measures to address these challenges.

In the next chapter we describe the development of VolumeViewer, our nonparallel contouring system, in the context of the conceptual framework constructed in this chapter. Chapter 4 details the algorithmic developments that we made to support our system, and Chapter 5 presents results from our evaluation of the system.

# Chapter 3

## VolumeViewer: An interactive tool for fitting surfaces to volume data

### 3.1 Introduction

In this chapter we describe the development of VolumeViewer (see Figure 3.1), a novel sketch-based interface that allows the user to generate a surface from just a few contours drawn on arbitrarily oriented planes. The two primary challenges that need to be addressed in the design of such an interface are the lack of user familiarity with the nonparallel planes and the difficulty with navigating through the 3D volume data. Therefore, our goals are to help the user maintain a sense of orientation, so that the user always knows where she is and what she is looking at, and to keep the time spent on navigation to a minimum.

We begin with a review of related work that highlights the need for such a system. We then describe the user interface (see Section 3.3), the development of contouring protocols and reference images (see Section 3.4) — which extend the interface and further address the challenges above, and several visualization cues that can enhance the contouring process (see Section 3.5). Each of these sections ends with a discussion of the design rationale behind our decisions, referring back to the hypotheses (see Section 2.3) developed in the previous chapter where appropriate. VolumeViewer is free and open source software. We conclude this chapter with the details of its implementation and distribution (see Section 3.6).

## 3.2 Related work

A large number of systems exist for the interactive segmentation of medical images. Here we list some of the most commonly used systems. This is not intended to be an exhaustive list, but rather to highlight the most popular systems and those that are most closely related to our own. We divide the systems into two groups based on whether they are closed or open source.

### 3.2.1 Closed source systems

Pinnacle by Philips [28] and Eclipse by Varian [76] are two of the common commercial systems for radiation treatment planning. These systems support a rich set of features for the simulation and optimization of radiation treatments. This includes selecting the energy and arrangement of the radiation beams in intensity modulated radiation therapy (IMRT) as well as the seed positions and dwell times for brachytherapy. They also include support for the segmentation of medical images which must be performed prior to the planning phase. While arbitrarily oriented image planes may be viewed, contouring is restricted to the axial slices.

The MIPAV (Medical Image Processing, Analysis, and Visualization) system [51] is a Java application developed by the National Institutes of Health for general analysis and visualization of biomedical data. It is closed source, but freely available and extensible via plug-ins that make use of a public application programming interface (API). It provides a basic suite of segmentation tools, but visualization and segmentation are limited to slices along the three primary axes.

MeVisLab [3] is not an end-user application, but rather an application framework for medical image processing research and development. It provides an image processing library, tools for visualization and interaction, and modules for integration with open source libraries, including OpenInventor [27], ITK [33], and VTK [64].

Amira [70] is another application framework designed for visualizing and analyzing life science and biomedical data. It provides support for 3D and 4D visualization, processing, and analysis from a wide range of imaging sources and modalities.

TurtleSeg [59] is a tool designed for the interactive segmentation of 3D medical images. The key to the approach is a 3D Livewire algorithm that allows the user to provide a sparse set of planar contours as input, which it uses to seed unvisited planes in order to generate a dense set of 3D contours. This method is particularly relevant, as the planar contours may be defined on nonparallel slices. However, since this method relies on Livewire, it generally only works well in situations where there is strong gradient information in the image data.

### 3.2.2 Open source systems

The Insight Segmentation and Registration Toolkit (ITK) [33] is an application framework that provides a range of general algorithms for image processing, segmentation, and registration in two and three dimensions. The Visualization Toolkit (VTK) [64] offers support for isosurfacing, mesh smoothing, mesh decimation, volume rendering, and vector and tensor visualization. It also provides many high-level modeling algorithms and support for 3D interaction. These toolkits provide a foundation for many other systems.

MITK [81] is an application framework built on top of ITK and VTK that is specifically designed for the development of interactive medical imaging software. It offers multiple, consistent and synchronized views of the same data and a state machine based interaction concept that includes an undo stack. It provides a module for manual segmentation that includes basic tools for manual segmentation, interpolation of missing slices from neighboring slices, surface model creation and analysis. Only the three primary axis directions are supported, because as the authors state, “...on rotated planes, the pixel filling operations would cause problems.”

ITK-SNAP [83] is an application built on top of ITK that is designed to provide a user-friendly interface for manual and semi-automatic segmentation of 3D medical images. Manual labeling of voxels is restricted to the three orthogonal viewing directions.

Seg3D [15] is an application developed by the NIH Center for Integrative Biomedical Computing at the University of Utah that combines a manual segmentation interface with algorithms for image processing and automatic segmentation provided by ITK. Image volumes can be labeled using volume rendering and orthogonal slice view windows.

3D Slicer [10, 24, 57, 58] is a general-purpose medical imaging platform that began as a masters thesis project in 1998 [31]. It is built on top of VTK and now contains over a million lines of code. It provides a variety of functionality, including image registration, automatic image segmentation, processing of diffusion tractography, and tracking of devices for image-guided procedures. It does allow for the visualization of arbitrarily oriented planes and includes an Editor module for manual segmentation, but editing is restricted to slices along the three primary axes.

Note that, with the exception of TurtleSeg, none of these systems provide support for contouring on nonparallel planes. Rather than create our own system, we could have extended one of the open-source systems listed above. However, none of these systems were really designed to support nonparallel contouring. They were designed for other types of segmentation and have many additional features which might be distracting or increase the learning curve for our users. Also, by extending another system, we might be constrained by what we could offer in terms of the interface due to having to integrate into an existing framework. Our aim was squarely on developing an effective system for the interactive segmentation of volumetric images using nonparallel contours. VolumeViewer was created as an evaluation tool to pursue this research aim and allow us to easily make iterative refinements to the interface. Once an effective interface is stable and tested, then it could be incorporated into an existing toolkit.

### **3.3 User interaction**

The interface consists of one large main window and three smaller side windows. Sketching takes place in the main window where the view is always perpendicular to the image plane. The side windows provide fixed views along the three primary axes (transverse, sagittal, and coronal) to help the user maintain an awareness of the

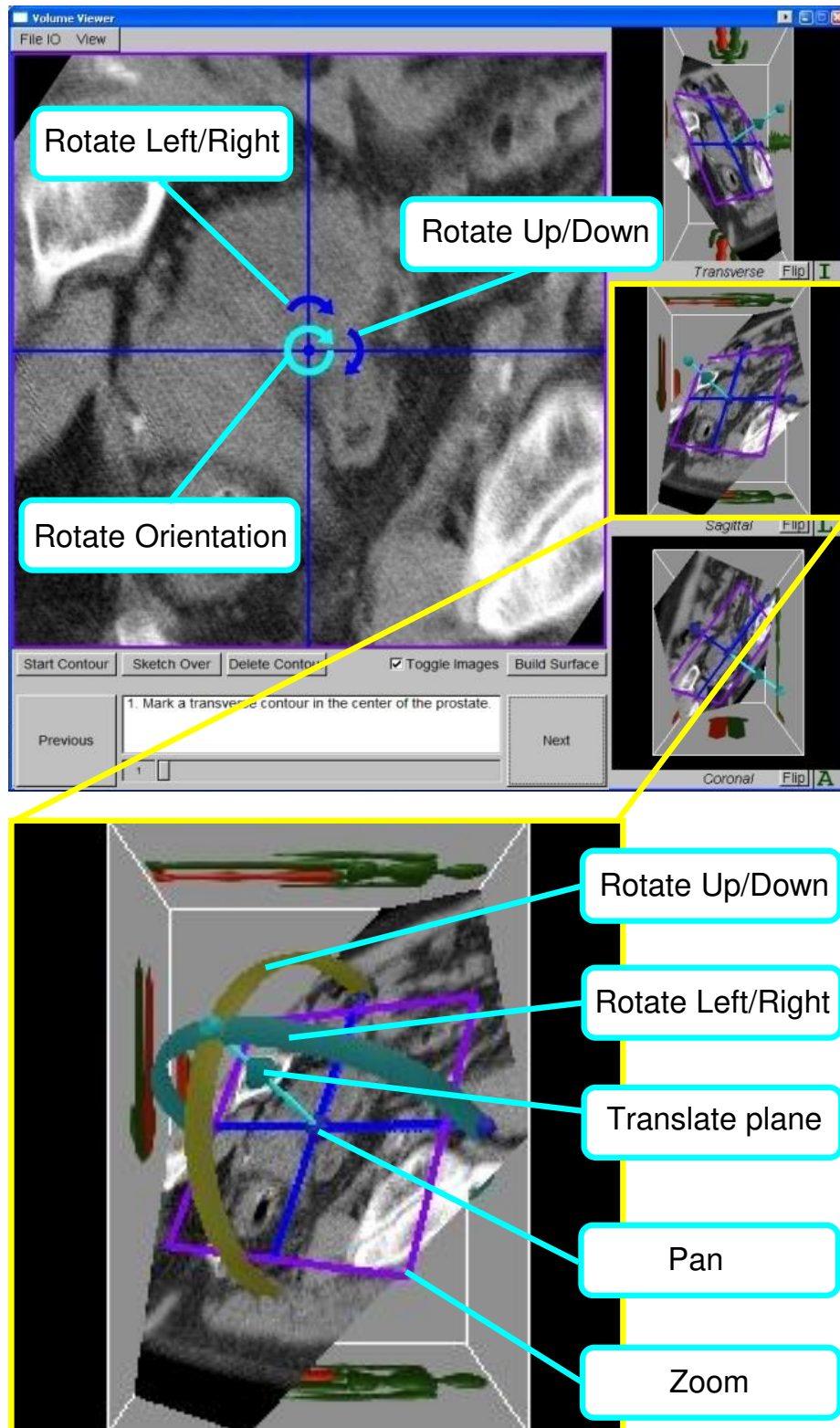


Figure 3.1: The VolumeViewer interface

current position and orientation of the image plane in the volume. The user’s mental model is that the view in the main window corresponds to a camera positioned on the tip of the blue arrow and looking at the purple box in the side windows. A bounding box of the volume is drawn in gray with images of a human model to indicate the patient’s orientation with respect to the volume data. This indicates to the user which end of the data is toward the head or feet of the patient, and the right leg of the model is colored red to help the user orient the slice.

From the main window, the user pans the current image plane by clicking and dragging the mouse anywhere on the image. The user rotates the plane up or down about its current axis or changes the viewing orientation by clicking and dragging the in-screen arrow controls, and translates the plane along its normal by scrolling the mouse. From the side windows, the user may also adjust the position and orientation of the plane by manipulating a simple transform widget (see Figure 3.1 (bottom)).

The user may sketch a contour on the current plane using one of three currently implemented drawing tools. “Paint” allows the user to contour using a traditional brush stroke. “Dots” allows the user to click a series of points that are linearly interpolated to form a contour, and an implementation of “LiveWire” [48] is also provided for use when the desired contour follows an image gradient. Contour points may be removed by pressing the delete key.

Once a set of contours has been drawn, a button is pressed to build a surface that interpolates the contours. After a surface has been reconstructed, it may be viewed simultaneously with the image data for evaluation. A clipping plane may be turned on to view the intersection of the image plane with the surface. The surface may then be edited by modifying existing contours or adding new contours and reconstructing the surface.

**Design rationale** One of the primary challenges to a nonparallel contouring system is the lack of user familiarity with the nonparallel planes (H4). Therefore, many of our design decisions were made to address this issue. For example, our initial prototype did not contain the fixed views in the side windows. When presented with only a nonparallel plane, users would often not understand at what angle and orientation they were viewing the structure. The fixed views together with the human model

drawn on the bounding box help the user determine the position and orientation of the slice within the greater volume.

The user can move or rotate the plane using either the in-screen controls or the transform widget in the side windows. In practice, we found that users tended to make use of the transform widget for large scale movements and the in-screen controls for fine-grained adjustments once the slice was already in the general region of interest within the volume. The important point here is that the movements are always *continuous* (H5). We start with a parallel plane that is familiar to the user and allow the user to adjust the position and orientation of the plane. As the user moves the plane, the image is updated continuously and the user can watch the planes position and orientation change in the side windows. This feedback helps the user maintain a sense of orientation while navigating the plane through the volume.

Another form of visual feedback that we provide to aid the user in performing the segmentation is the ability to build and view the surface, together with the image data, at any point during the contouring process (H7). The surface model serves as a 3D landmark for the user during navigation and allows the user to evaluate his or her progress and to better determine where the next contour should be added. It also makes refining the segmentation more efficient, because once a surface has been built, new contours no longer have to be drawn from scratch. Instead, the intersection of any plane with the surface defines a contour which can be edited by simply sketching over the portions of the existing contour that need adjusting.

### 3.4 Contouring protocols

The VolumeViewer interface as described in the previous section provides all the tools necessary to perform segmentation using nonparallel contours. However, it places the burden of determining on which planes to sketch and navigating to those planes entirely on the user. When we know the general shape of the structure that we are trying to reconstruct, we can aid the user by providing an initial set of planes on which to draw and automatically performing the 3D navigation from one plane to the next. For this, we provide contouring protocols that include step-by-step instructions for performing the segmentation and smooth, animated transitions from one image



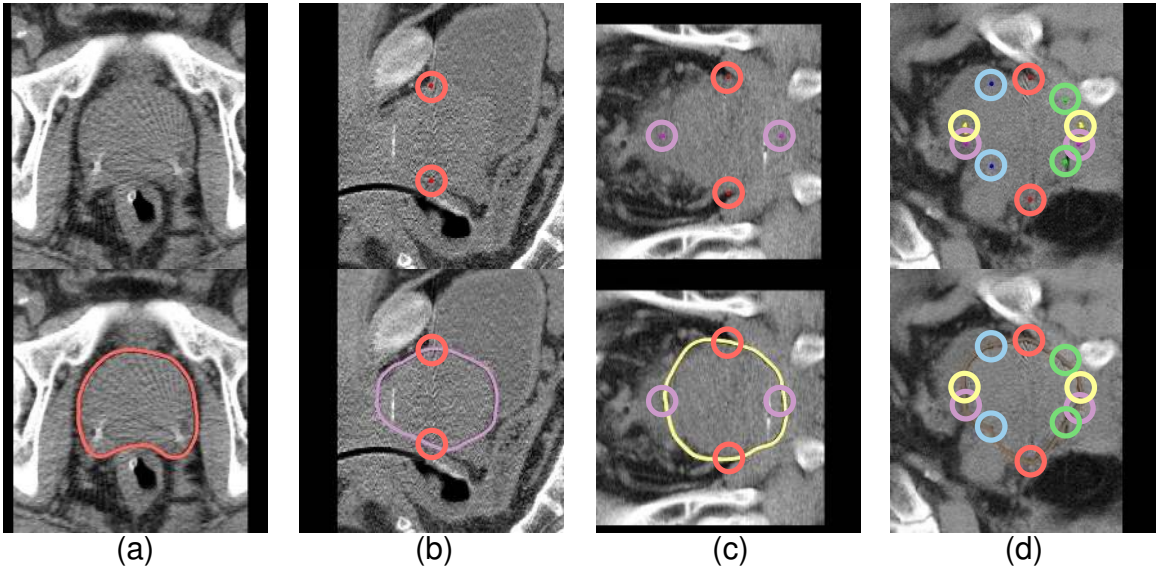


Figure 3.2: An example protocol for contouring the prostate. The top row shows what the user sees at the beginning of each step. The bottom row shows what the user draws. The user makes use of the intersection points of previously drawn contours as well as a set of reference images (see Figure 3.4) to aid in marking the contour at each step. (a) The user marks the first transverse contour in the middle of the prostate. (b) The user marks a second contour, perpendicular to the first. (c) The user marks a third contour, perpendicular to the first two. (d) The user marks the seventh and final contour. The resulting contours and surface are shown in Figure 3.3.

plane to the next. The use of these protocols with a set of corresponding reference images (see Figure 3.4) helps to ensure that users consistently produce surfaces of the highest quality.

We refer to this type of interaction as *spatio-temporal navigation*, as opposed to the free-form navigation described in the previous section. The idea is that the user moves through the volume in a set of pre-specified moves and views that are defined *relative* to one another, and are placed in a *temporal* order.

For example, we have experimentally determined a set of nonparallel planes that are generally sufficient at capturing the shape of the prostate [2] (see Figure 3.3). This consists of three transverse planes, one sagittal plane, one coronal plane, and two nonparallel sagittal-coronal planes. One possible contouring protocol for this structure using these planes is illustrated in Figure 3.2. To begin, the user is instructed to navigate to the transverse plane that lies in the middle of the prostate and mark a

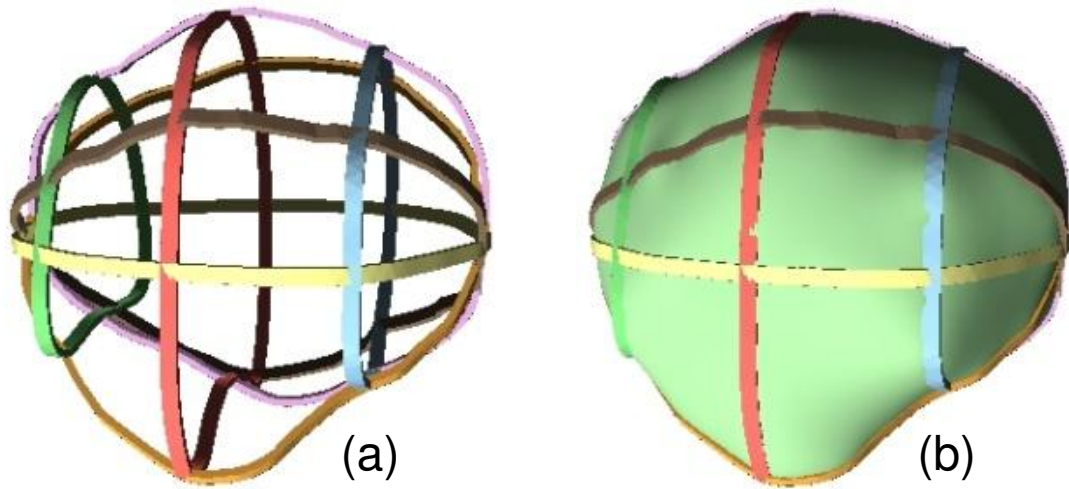


Figure 3.3: Contours and surface model generated by example prostate protocol. The contours (a) drawn by the user while following the protocol depicted in Figure 3.2 to segment a human prostate. The resulting surface model (b), reconstructed by our system.

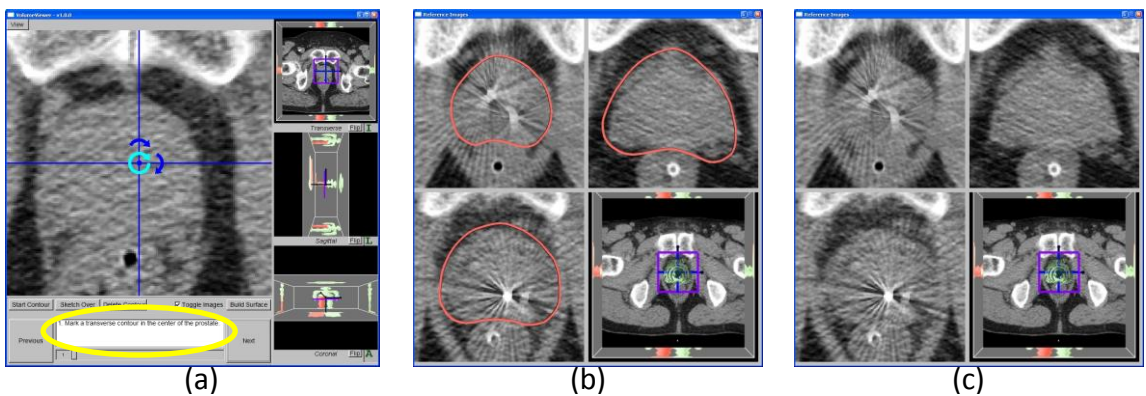


Figure 3.4: Reference images. The main window (a) displays the current image plane together with a protocol instruction (circled in yellow) that describes where the next contour should be marked. Reference images of similar contours in other patient data sets are provided as examples. The contours can be toggled on (b) or off (c) to train the user to look for cues in the image data.

contour (see Figure 3.2a). Once this contour is drawn, the user needs to perform no further navigation. The system automatically rotates the first plane ninety degrees about the center of the contour and instructs the user to mark the sagittal contour (see Figure 3.2b). From these two contours, the other five planes are easily generated and automatically navigated to by the system (see Figure 3.2c,d).

For each contour, the user is presented with a set of reference images (see Figure 3.4), and simple instructions to look for specific anatomic cues in the image in order to accurately delineate the contour. The reference images are presented to the user in a separate window of the interface. For each nonparallel plane, the window contains four reference images. Three of these images show example contours drawn on similar cross-sections of the anatomy of three different patients. These contours were made by intersecting the nonparallel planes with a surface model reconstructed from many physician-reviewed contours drawn on the parallel cross-sections of the data set. The contours can be toggled on and off, to give the user a sense of what they should be looking for in the image data. The fourth image illustrates which cross-section the user is currently observing, by showing its position as a slicing plane through a surface model of the anatomy.

We have worked with a team of radiation oncologists to develop contouring protocols for both the prostate and liver. Alternative protocols for these structures or new protocols for other structures can easily be created and used by VolumeViewer. This process simply involves writing a list of instructions and commands for transforming the image plane from one position to the next and capturing a set of corresponding reference images to be used in conjunction with the protocol.

**Design rationale** 3D navigation is a time-consuming and difficult task. In the absence of any contouring protocol, much of our users' time would be spent manipulating the image plane from one position and orientation to the next, as well as determining which contour should be marked next. We created the contouring protocols to remove this guesswork and make much of the required navigation happen automatically. The user still has the ability to manipulate the image plane as described in the previous section, but this is now generally used to make local adjustments to the plane. The automatic navigation will move the plane to the general vicinity of where the next

contour needs to be placed. The movement of the plane is animated smoothly and continuously (H5) to help the user maintain a sense of orientation, and the information provided by previously drawn contours is used to keep the region of interest centered and the same size on the screen (H6).

The other motivation behind our use of contouring protocols is that since we are instructing the user as to which plane to contour next, we are able to provide reference images from other patient data sets that correspond to that plane. This further helps to address the challenge posed by the users being unfamiliar with the nonparallel views (H4). The user can immediately determine the general shape and orientation of the structure in the current view by glancing at the set of reference images. Moreover, having users contour on the same set of planes at the outset helps to increase consistency across users.

### 3.5 Visualization cues

VolumeViewer allows the user to view the image data by moving a two-dimensional (2D) slicing plane through the volume. The user can mark contours on this plane, and once a surface is reconstructed, it can be viewed along with the image data. These are the basic visualization requirements for a nonparallel contouring system. However, the image data is often noisy, and the user must rely on subtle cues to delineate the contours. In this section, we explore several visual enhancements that can be made to the image data and the way that the contours and surface models are rendered in order to make the segmentation process easier for the user.

One such visualization enhancement is illustrated in Figure 3.5. As the image plane rotates from one position to the next, the user can see portions of the previously drawn contours that stick out above the image plane. This can be very useful in order to help the user maintain a point of reference and understand the overall shape of the structure that is being segmented. However, the spatial context of 3D curves floating in space can be difficult to infer (see Figure 3.5a). By adding a bit of geometry, so that the curves become thin ribbons, and using directed lighting and soft shadows where the curves intersect the plane, the spatial context becomes much more apparent (see Figure 3.5b).

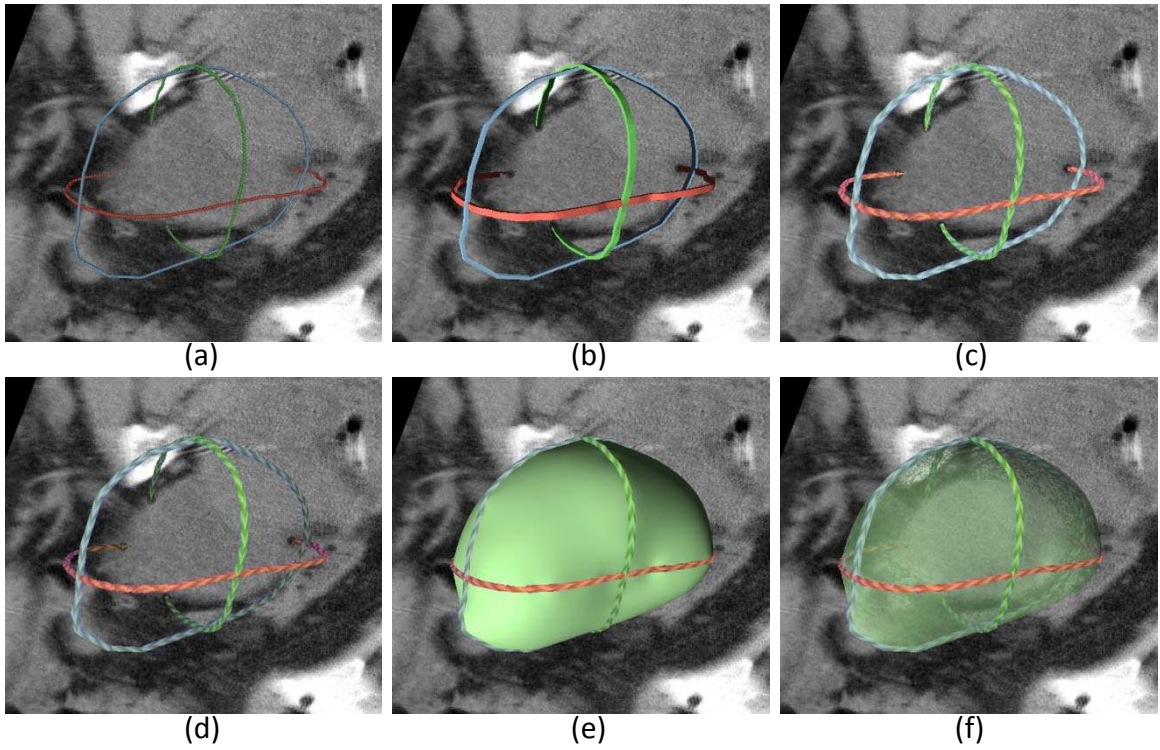


Figure 3.5: Rendering contours as thin ribbons. Visualizing the shape of an ongoing segmentation is challenging, as curves floating in space (a) are difficult to draw and understand. We can make the 3D structure of the curves more apparent by drawing the curves as thin ribbons, with directed lighting and shadows where the curves intersect the plane (b), adding a linear texture to the ribbons (c), and shading the contours so that portions of the contours that are further away appear darker (d). The underlying surface (e) can also be rendered translucently, to avoid obscuring the image data (f)

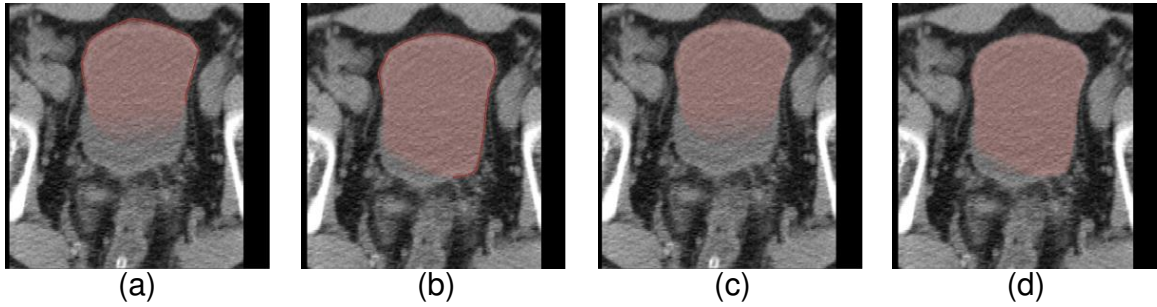


Figure 3.6: Coloring the image plane. A partial segmentation of the bladder is shown (a). The user-drawn contour is shown in solid red. The part of the image that is considered inside the surface is tinted with red hue. The user continues drawing the contour to further define the segmentation (b). The same images are shown without the contours (c, d). Note that when coloring the segmentation in this manner, there is no need to draw the actual contour. It can be omitted so as to not obscure the image data.

In addition, we can add a linear texture to the contour ribbon (see Figure 3.5c), and shade the contours based on their distance from the viewpoint which enhances the perceived depth (see Figure 3.5d). This all helps to better convey the 3D structure of the curves and the underlying shape that they define (see Figure 3.5e). The surface model itself can be drawn translucently, so that the underlying image data can still be seen (see Figure 3.5f).

While drawing the contours in this manner works well for visualizing the 3D shape of a set of contours, it is less than ideal when sketching a contour on a given plane. In this case, the solid geometry of the contour actually obscures the image data that the user is segmenting. A better approach is to color with hue the part of the image that is inside the surface for a given slice (see Figure 3.6). When the user begins to draw a contour, an initial estimated segmentation is found by extending the region enclosed by the contour a fixed distance. This area of the image data is colored with hue, appearing darker nearest the contour, where the segmentation is more certain, and fading out from there. As the user continues to draw the contour, the segmentation becomes more refined. The user always knows which portion of the plane is considered inside and outside the surface, and the entire image data remains visible.



**Design rationale** The user segments a 3D structure by marking a set of 2D contours. To do this effectively, it is important for the user to understand the impact that each 2D contour has on the generated 3D surface (H7). This surface model can serve as a landmark during navigation, and when viewed along with the image data, can help the user determine where more input is needed. All of the visualization enhancements described in this section were designed with the intention of conveying as much 3D information as possible to the user while obscuring the underlying image data as little as possible.

## 3.6 Implementation and distribution

The input to our system is a volumetric data set. The examples in Figure 3.3 were reconstructed from a 1024x1024x256 CT scan of a human prostate. The data is stored as a 3D texture and any desired smoothing or filtering can be performed in the fragment shader. The system linearly interpolates the data in each slice of the input data set in order to render arbitrarily oriented image planes. After a set of contours have been drawn using the tools described in Section 3.3, the system reconstructs a surface from the nonparallel contours using the technique of [44]. This algorithm requires that the contours be closed and that the intersection points along the common line between any two planes be located at the same position. For details on how we ensure this type of consistency between the contours, see Section 4.3. VolumeViewer is implemented entirely in the C++ programming language, using the Open Graphics Library (OpenGL) [26] for graphics and rendering, and the Fast Light Toolkit (FLTK) [69] for the graphical user interface (GUI).

For the general audience, VolumeViewer is freely available for use under the terms of the GNU General Public License. The VolumeViewer website [67], available via <http://volumeviewer.cse.wustl.edu>, is a comprehensive resource for all VolumeViewer users. Here we have all major and minor releases, cross-platform source code, sample data sets, a user guide, video tutorials, and a bug tracking system that allows users to submit errors or feature requests. To date, we have 56 registered users from 13 different countries who have downloaded VolumeViewer.

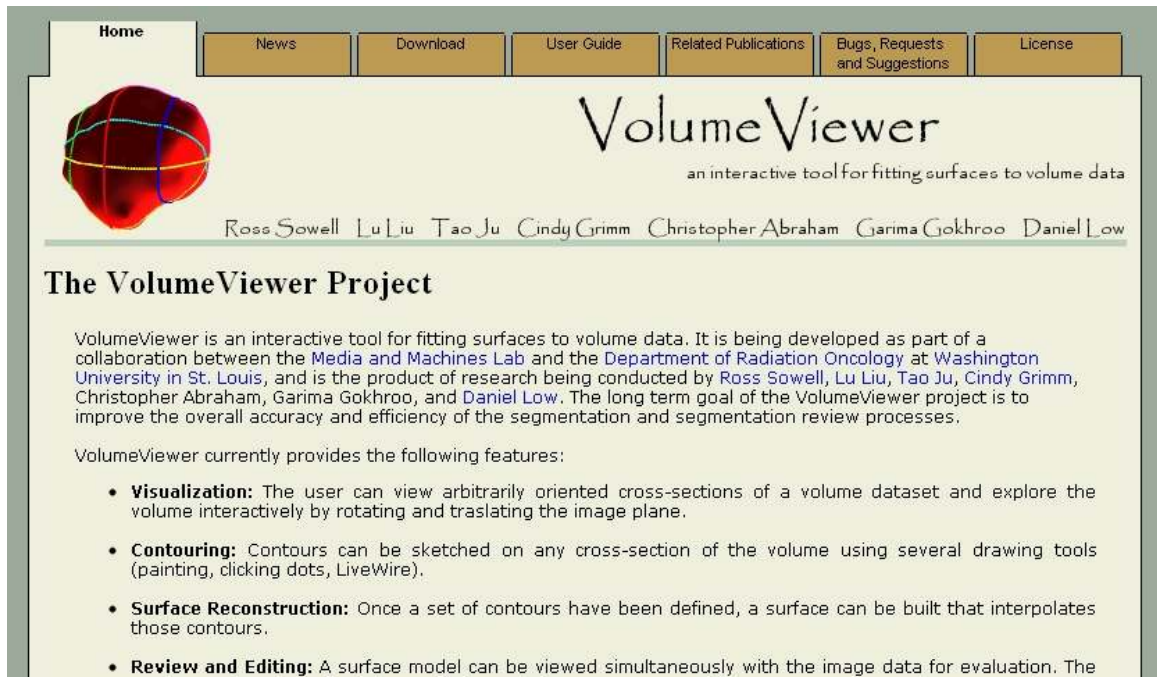


Figure 3.7: A screenshot of the VolumeViewer website

This system was implemented solely by the author of this dissertation with a few exceptions. VolumeViewer makes use of a free and open-source suite of tools for mesh processing and basic graphics routines [25]. The author worked closely with Liu et al. [44] to integrate their software for surface reconstruction into our system. The author served as a student mentor for two students that made valuable contributions to the project under the authors supervision. Michelle Vaughan, then a first-year graduate student, implemented the HRBF technique for partial contouring (see Section 4.4) and a few of the visualization cues, including textured ribbons, depth shading, and surface translucency. Chris Trn, an undergraduate student at the time, implemented the approach for coloring the image plane with hue during sketching. The rest of the software including the interface, visualization, user interaction, protocols, methods for ensuring contour consistency, and tools for editing were all implemented by the author.



## 3.7 Chapter summary

We have presented the VolumeViewer interface and described how it can be used to explore a volumetric data set, sketch contours on arbitrarily oriented cross-sections of the volume, construct a surface model that interpolates those contours, and review and edit the resulting segmentations. We defined contouring protocols that enable spatio-temporal navigation for segmenting known structures consistently and efficiently, and we described a number of visualization cues that can be used to provide the user with spatial context and make the segmentation process easier and more intuitive. We detailed the implementation of VolumeViewer and noted that it is freely available on our website. In the next chapter, we will develop the algorithms necessary to support our nonparallel contouring system.

# Chapter 4

## Algorithms for nonparallel contouring

### 4.1 Introduction

In the previous chapter we presented an interface that allows users to navigate through a volumetric data set and sketch a set of contours on arbitrary cross-sections of the data. In order to construct surface models from these user-drawn contours, several algorithmic developments were required that are the focus of this chapter. We begin with a discussion of contour consistency and how to ensure that the contours a user draws are consistent. We then describe an extension to an existing reconstruction algorithm that provides support for accepting partial (non-closed) contours as input.

### 4.2 Related work

The earliest approaches to surface reconstruction from contours [40, 23] focus on simple closed curves lying on parallel planes and are known as “contour stitching”. Many advancements have been made since then, but the basic approach remains the same: connect the vertices of adjacent contours to build a mesh that passes through all the input contours. For a complete review of methods of surface reconstruction from parallel contours, please see the literature review in [44].

More recently, several methods have been developed for reconstructing surfaces from curves on nonparallel planes [62, 54, 79, 17, 6, 7, 44]. A common strategy adopted in these methods is to consider a partitioning of space by all cross-sectional planes. Within each partitioned cell, a closed triangular mesh is extracted as the exterior of a 3D tetrahedral mesh that connects the cell boundary. These methods provide the inspiration for our work, as they make it possible to reconstruct surfaces using fewer nonparallel planes. However, current commercial treatment planning systems such as Philips’ Pinnacle [28] and Varian’s Eclipse [76] do not provide facilities for visualizing or sketching contours on non-transverse image planes. Even if they did provide such support for drawing nonparallel contours in the interface, there remains a missing link between allowing users to contour on nonparallel planes, and transforming that input into a *consistent* set of contours that can be accepted as input to a surface reconstruction algorithm. That missing link is the subject of the next section.

### 4.3 Ensuring contour consistency

Available algorithms for reconstructing a surface from a set of nonparallel contours [7, 44] require that the input contours be closed and consistent. By consistent, we mean that the intersection points lying along the common line between any two planes must be located at the same position. An example of a set of inconsistent contours is shown in Figure 4.1. It is trivial to force the user to sketch a closed contour, but ensuring the consistency of a set of user-drawn contours is more difficult.

We begin by defining formally what it means for a set of contours to be *consistent* (see Section 4.3.1). Then, in Section 4.3.2 we describe a semi-automated approach for taking a set of contours and making them consistent. In Section 4.3.3, we describe an alternative approach that involves modifying the user interface to require the user to always maintain a consistent set of contours throughout the segmentation process.

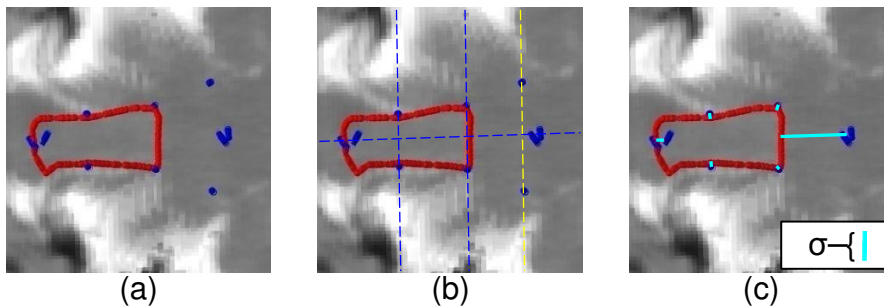


Figure 4.1: An example of inconsistent contours. This example is taken from a set of user-drawn contours in our first user study (see Section 5.3). (a) The red contour is inconsistent with the other contours, whose intersection points with the plane are shown in blue. (b) The red contour violates the topological validation condition, because it does not intersect the yellow common line twice. (c) If the contour defining the yellow common line had not been drawn, the red contour would still violate the geometric validation condition, because its distance (displayed in cyan) to one of the intersection points is greater than the allowable distance  $\sigma$ .

### 4.3.1 Formal definition of consistency

We begin with a set of  $n$  planar contours  $C = \{C_1, C_2, \dots, C_n\}$ , where each contour is a simple, closed curve defined by a sequence of vertices  $C_i = \{c_1, \dots, c_m | c_j \in \mathbf{R}^3\}$ . Consecutive vertices in the sequence are assumed to be connected to one another, and since the contour is closed,  $c_m$  is connected to  $c_1$ .

Each contour  $C_i$  lies on a plane  $P_i$  embedded in  $\mathbf{R}^3$  defined by any set of three noncollinear points in  $C_i$ . Let  $L$  be the set of all lines found by intersecting each pair of planes  $(P_i, P_j)$  defined by the contours  $C_i, C_j \in C$ . Note that each pair of planes will intersect in a single line, unless they are parallel (or are the same plane in the case of two contours lying on the same plane), and that multiple planes can share the same line of intersection. We refer to  $L$  as the set of common lines.

Each common line  $l \in L$  is shared by a subset of the contours in  $C$ . Let  $C_l \subseteq C$  such that for all  $C_i, C_j \in C_l$ , the intersection of  $P_i$  and  $P_j$  is  $l$ . Furthermore, each contour sharing a common line will intersect that line 0 or more times. Let  $I_{li}$  denote the set of intersection points between line  $l$  and contour  $C_i$ . That is,  $I_{li} = \{h \in \mathbf{R}^3 | l \text{ and } C_i \text{ intersect at } h, l \in L \text{ and } C_i \in C_l\}$ .

Note that we can order this set of intersection points according to their positions along the line  $l$ . This requires us to orient the line  $l$ , but the choice of orientation is arbitrary. Let  $h_k \in I_{l_i}$  be the  $k^{\text{th}}$  intersection point of contour  $C_i$  along the line  $l$ . This has the effect of identifying corresponding intersection points. For example, if the contours  $C_i$  and  $C_j$  share the common line  $l$ , then  $h_k \in I_{l_i}$  corresponds to  $h_k \in I_{l_j}$ .

In order for a set of contours  $C$  to be *consistent*, they must be both topologically consistent and geometrically consistent.

A set of contours  $C$  is *topologically consistent* if each contour that shares a common line intersects that line the same number of times. Formally,

$$\forall l \in L, \forall C_i, C_j \in C_l (|I_{l_i}| = |I_{l_j}|).$$

A set of contours  $C$  is *geometrically consistent* if the corresponding intersection points of each contour sharing a common line all lie at the same position. Formally,

$$\forall l \in L, \forall C_i, C_j \in C_l (h_k \in I_{l_i} = h_k \in I_{l_j}).$$

Surface reconstruction algorithms [7, 44] require that the input contours are consistent. Unfortunately, an arbitrary set of user drawn contours is unlikely to satisfy this property. We now describe a semi-automated approach for taking a set of contours and making them consistent.

### 4.3.2 Contour validation and snapping

We developed a procedure called MAKECONSISTENT, that takes an inconsistent set of contours as input and outputs true, along with a modified set of the input contours that is consistent, or outputs false, along with a list of the invalid intersection points, if the input contours cannot be made consistent automatically. Pseudocode for the procedure is given in Algorithm 1.

If this procedure returns true, then the contours will be consistent and ready to be used as input to a surface reconstruction routine. A return value of false indicates that the contours are either topologically or geometrically invalid. Figure 4.1 illustrates both of the possible validation failure cases. If the contour does not pass validation, then the user is notified. The intersection points that are causing the failure are

---

**Algorithm 1** Make consistent algorithm

---

```
1: procedure MAKECONSISTENT( $C, \sigma$ )
2:   Input. A set of contours  $C$  and a threshold distance  $\sigma$ .
3:   Output. True and a consistent set of contours  $C'$ , if the input contours are
   topologically and geometrically valid; false and a list of the invalid intersection
   points otherwise.
4:   if  $\neg$ ARETOPVALID( $C$ ) then return false.
5:   if  $\neg$ AREGEOMVALID( $C, \sigma$ ) then return false.
6:    $C' = \text{SNAP}(C, H, T)$ 
7:   return true
8: end procedure
```

---

highlighted, and the user can either edit the selected contour or drag the intersection points along the common line to modify intersecting contours. After corrections have been made, the MAKECONSISTENT procedure can be repeated.

The MAKECONSISTENT procedure relies on three subroutines, ARETOPVALID, AREGEOMVALID, and SNAP, which we now define in turn.

**Validation** Our goal is to take a set of inconsistent contours and make them consistent. In order to do that, we must first detect the inconsistencies and ensure that they are something we can repair automatically. This is the job of the ARETOPVALID and AREGEOMVALID procedures.

Recall that for a set of contours to be consistent, they must be both topologically and geometrically consistent. The conditions for a set of contours to be *valid* are similar, but with the geometrical consistency requirement relaxed.

There are two conditions that will cause a set of contours to fail validation, resulting in a return value of false from the MAKECONSISTENT procedure. The first is if the contours are topologically inconsistent. While topological inconsistencies are easy to detect, we do not have a method for automatically repairing them. A topological inconsistency is considered to be a major inconsistency that we prefer to have the user manually correct. Pseudocode to check for topological inconsistencies is provided in Algorithm 2.

---

**Algorithm 2** Topological validation algorithm

---

```
1: procedure ARETOPVALID( $C$ )
2:   Input. A set of contours  $C$ .
3:   Output. True if the input set of contours satisfy the topological validation
   condition; false and a list of the invalid intersection points otherwise.
4:   Compute the set of common lines  $L$  shared by the contours in  $C$ .
5:   For each  $l \in L$ , compute the set of intersection points,  $I_l = \{h \in \mathbf{R}^3 | l \text{ and } C_i$ 
   intersect at  $h, l \in L \text{ and } C_i \in C\}$ , of the contours that share it.
6:   For each  $l \in L$ , if  $\exists C_i, C_j \in C_l, |I_{l_i}| \neq |I_{l_j}|$  then return false.
7:   return true
8: end procedure
```

---

On the other hand, we can automatically repair a set of contours that is topologically consistent but geometrically inconsistent. Recall that a geometrical inconsistency implies that  $\exists l \in L, \exists C_i, C_j \in C_l (\exists h_k \in I_{l_i} \neq h_k \in I_{l_j})$ . For each such inconsistency we define a target point  $h'_k$  to be the average of all points  $h_k \in I_{l_i}, \forall C_i \in C_l$ . We then use the SNAP procedure, which we define later in this section, to deform each  $C_i \in C_l$  such that  $h_k \in I_{l_i} = h'_k$ .

Note, however, that while we can repair all geometrical inconsistencies in this manner, we do not choose to do so when the distance between  $h_k$  and  $h'_k$  is large. This is considered to be a major inconsistency that the user should correct manually, and automatic correction might result in creating contour self-intersections or new intersection points that did not previously exist.

To detect and prevent this problem, we introduce a user-defined threshold distance  $\sigma$ , specifying how close together the intersection points need to be in order to be considered valid. If  $\|h'_k - h_k\| > \sigma$  for any target-intersection pair, the contours fail validation. As long as  $\sigma < \delta$ , where  $\delta$  is the minimum distance between any  $h_k$  and any point of intersection of the common lines, we can guarantee that the deformed contours  $C'$  will not contain any self-intersections or intersection points that did not previously exist. By default we initialize  $\sigma$  to be a small fraction of the estimated size of the structure being segmented, and we update it as needed to ensure that  $\sigma < \delta$ .

Pseudocode for the AREGEOMVALID procedure is given in Algorithm 3.

---

**Algorithm 3** Geometric validation algorithm
 

---

- 1: **procedure** AREGEOMVALID( $C, \sigma$ )
  - 2:    *Input.* A set of contours  $C$  and a threshold distance  $\sigma$ .
  - 3:    *Output.* True if the input set of contours satisfy the geometric validation condition; false and a list of the invalid intersection points otherwise. As a side effect, each contour  $C_i \in C$  is updated to include its intersection points,  $H_i = \{h_k \in I_{li} | l \in L\}$ , with the common lines as vertices. Furthermore, for the set of intersection points on each contour, a corresponding set of target points is computed  $T_i = \{h'_k \in \mathbf{R}^3 | h'_k \text{ is equal to the average of all points } h_k \in I_{li}, \forall C_i \in C_i\}$ .
  - 4:    For each  $l \in L$ , sort the intersection points  $I_{li}$  into order along  $l$ .
  - 5:    For each  $l \in L$ , compute a target point  $h'_k$  for each intersection point  $h_k \in I_{li}$ ,  $C_i \in C_l$ . Doing this for all lines in  $L$  results in the set of target points  $T_i$  for each contour  $C_i$ .
  - 6:    For each contour  $C_i \in C$ , **if**  $\exists h_k \in H_i, h'_k \in T_i (\|h'_k - h_k\| > \sigma)$ , **then return** false.
  - 7:    For each contour  $C_i \in C$ , add the intersection points  $H_i$  as vertices.
  - 8:    **return** true
  - 9: **end procedure**
- 

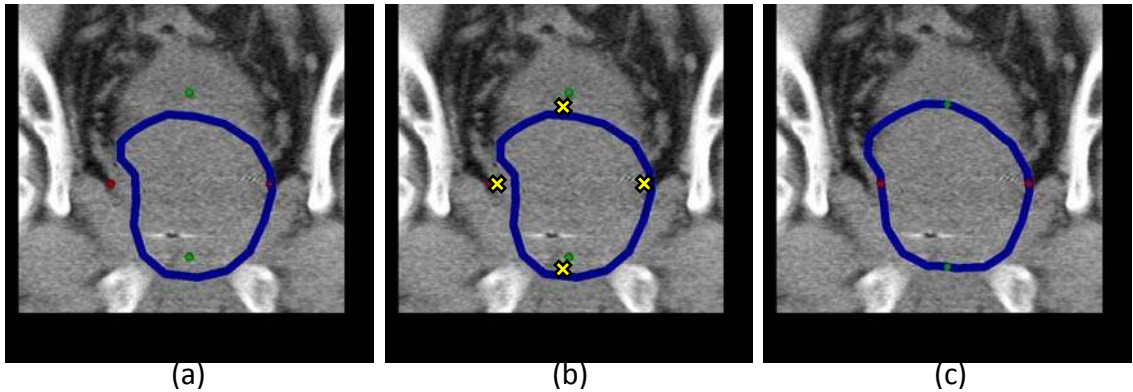


Figure 4.2: Contour snapping. (a) These contours pass validation, but the blue contour is inconsistent with the intersecting red and green contours. (b) Target points, shown as yellow crosses, are defined to be the midpoint between each pair of intersection points. (c) Each contour is deformed to pass through the target points using constrained Laplacian editing.



**Snapping** After all the contours have been successfully validated, they are then automatically snapped together before being used as input to the surface reconstruction algorithm. For this, we use a contour snapping process based on constrained Laplacian editing [66]. This technique is presented in the context of editing 3D surface meshes, but as the authors indicate, the technique can be applied to 2D meshes (curves) just as well. For completeness, we describe our implementation of this approach in the context of 2D contour deformation. We chose this technique because it allows us to deform the contours to pass through the target points while preserving the overall shape of the contour as much as possible.

---

**Algorithm 4** Snapping algorithm

---

```

1: procedure SNAP( $C, H, T$ )
2:   Input. A set of contours  $C$ . It is assumed that for each  $C_i \in C$ , there is a
   set of intersection points, or handles,  $H_i \subset C_i$ , and a corresponding set of target
   points  $T_i$ . In our algorithm, these are computed in the AREVALID routine above.
3:   Output. A set of contours  $C'$  such that for each  $C'_i \in C'$ ,  $h_k \in H'_i = h'_k \in T_i$ .
4:   for each  $C_i \in C$  do
5:     Compute the transformation matrix  $S$  that transforms  $C_i$  to the  $xy$  plane.
     This will allow us to work in 2D. Transform the contour vertices  $c_i \in C_i$  and
     target points  $t_i \in T_i$  by  $S$  and drop the  $z$ -coordinate, yielding a corresponding 2D
     contour  $V$  and set of 2D target points  $U$ .
6:     Compute the Laplacian of each vertex:  $L(v_j) = v_j - 0.5(v_{j-1} + v_{j+1})$ 
7:     Solve for the new set of vertices  $V'$  by minimizing the following error
     functional:  $E(V') = \sum_{v_j \in V} \|R_j(V')L(v_j) - L(v'_j)\|^2 + \sum_{u_j \in U} \|v'_j - u'_j\|^2$ , where

$$R_j = \begin{pmatrix} a & w & r_x \\ -w & a & r_y \\ 0 & 0 & 1 \end{pmatrix}$$

8:     Transform  $V'$  by  $S^{-1}$ , to yield the resulting contour  $C'_i$ .
9:   end for
10: end procedure

```

---

An example of this deformation process is shown in Figure 4.2. In this way, we are able to ensure consistency, while requiring minimal input from the user and modifying the shape of the user-drawn contours as little as possible.

### 4.3.3 Validate-as-you-go process

One major drawback to the approach described in the previous section is that if one or more contours fail validation, then the user must go back to previously drawn contours in order to make the necessary corrections. This can be time-consuming, particularly if there are multiple inconsistencies. Additionally, even if the contours all pass validation, the automated snapping process makes changes to the contours over which the user has no control. In some situations, this may be the only option, such as when the contours were generated in another system or by another process. However, since we have control over the interface for contouring in our system, we can make changes that address these issues by employing a validate-as-you-go process.

In this approach, when the user adds a new contour, we ensure contour consistency by requiring that the new contour passes through the intersection points of previously drawn contours. We make connecting the intersection points easy by highlighting the intersection dots and connecting the contour to an intersection point when the user clicks in its vicinity. If the completed contour does not pass through all of the intersection points, it will be automatically adjusted to do so by projecting the missed intersection points onto the contour to get handle vertices that are then moved to the corresponding intersection points using the same snapping procedure described in the previous section.

The user can always adjust the position of the intersection points by clicking and dragging them along the common line. Doing so will modify not only the current contour, but also any other previously drawn contours that pass through the intersection point. Again, this is done using the same snapping procedure as before, except that the handle point is the intersection point being dragged and the target point is the position to which it is dragged. By validating the contours as the user draws them, we are able to ensure that the set of drawn contours are always consistent, and that the user is in control of any adjustments that get made to the contours.

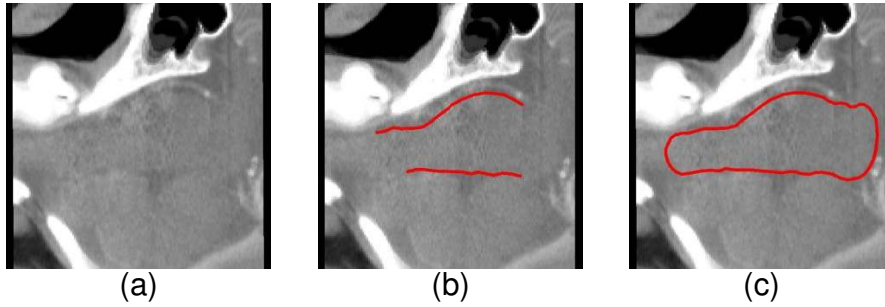


Figure 4.3: Motivation for partial contouring. (a) The input cross-sectional image is a sagittal view of the brainstem. (b) The partial contour that represents the portion of the boundary in which the user is confident. (c) The completed contour. We extend the surface reconstruction algorithm to support partial contours.

## 4.4 Supporting partial contouring

In addition to being consistent, current algorithms for reconstructing surfaces from nonparallel contours [7, 44] require that the input contours be closed. This restriction is unfortunate because in many cases the complete boundary of the structure is not able to be delineated in a given view (see Figure 4.3). In these cases, the user must make an educated guess in order to complete the contour. Later, when contouring another plane, the user may notice that this portion of the contour is inaccurate, and must then edit the previous contour. Moreover, future contouring decisions made by the user are influenced by these uncertain portions of the contours, because the interface makes no visual distinction between portions of the contours that the user is confident or not confident about.

In other cases, contouring the entire boundary of a structure is unnecessary because autosegmentation techniques can accurately segment the majority of the structure. An example is an attached lung tumor, where much of the tumor can be automatically segmented from the lungs, but the attached portion provides an ambiguous region that can only be segmented by an experienced user (see Figure 4.4). In such cases, the user should only have to contour the difficult regions and the system should automatically complete the segmentation. Thus, a reconstruction algorithm that accepts partial contours could improve the accuracy and efficiency of the segmentation process.

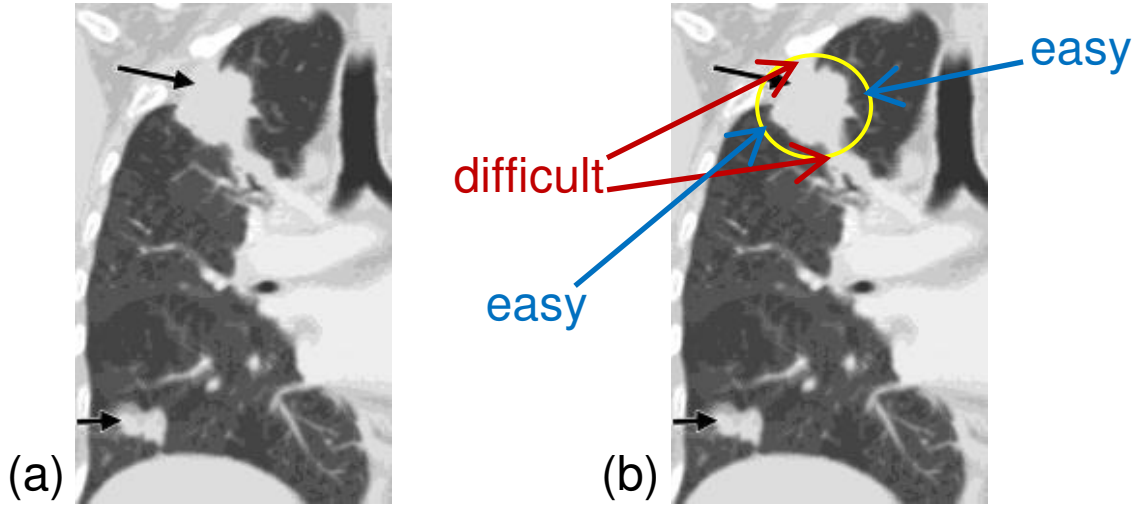


Figure 4.4: Stage T4 lung tumors. (a) Chest CT scan shows a primary lung tumor in the right upper lobe (long black arrow) with a smaller separate nodule in the right lower lobe (short black arrow). Original image courtesy of UyBico et al. [74]. (b) Some portions of the primary tumor (circled) are difficult to segment, while others are trivially easy. We propose a system that supports partial contouring, allowing the user to manually segment the difficult portions, while the rest is handled automatically.

#### 4.4.1 Extending Hermite radial basis function implicits

Recently introduced by Macedo et al. [46, 9], Hermite radial basis function (HRBF) implicits provide a powerful tool to reconstruct implicitly-defined surfaces from points and normals, rather than closed planar contours. Its intrinsic properties allow it to handle irregularly spaced points as well as operate in the presence of close sheets. We extend this method to support partial contouring (see Figure 4.5).

For completeness, we provide the relevant equations from Macedo et al. [46, 9]. The authors frame the problem of reconstructing an implicit surface from points  $\{\mathbf{x}^j\}_{j=1}^N \subset \mathbf{R}^3$  and normals  $\{\mathbf{n}^j\}_{j=1}^N \subset \mathbf{S}^2$  as a Hermite interpolation problem, in which a function  $f: \mathbf{R}^3 \rightarrow \mathbf{R}$  was sought such that  $f(\mathbf{x}^j) = 0$  and  $\nabla f(\mathbf{x}^j) = \mathbf{n}^j$ , for each  $j = 1, \dots, N$ . They show that this can be solved using the following concrete form:

$$f(\mathbf{x}) = \sum_{j=1}^N \{\alpha_j \|\mathbf{x} - \mathbf{x}^j\|^3 - 3 \langle B^j, \mathbf{x} - \mathbf{x}^j \rangle \|\mathbf{x} - \mathbf{x}^j\|\} + \langle \mathbf{a}, \mathbf{x} \rangle + b,$$

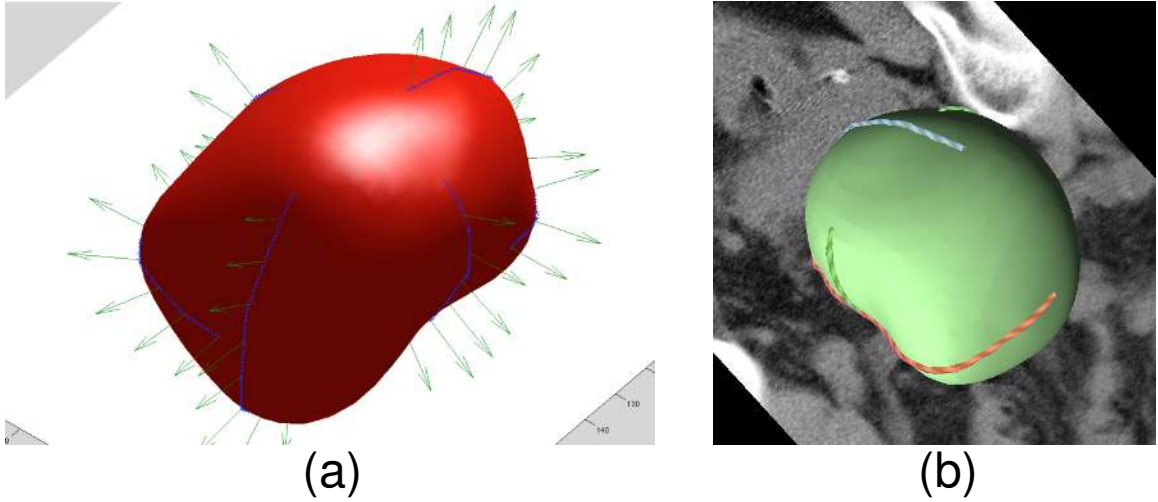


Figure 4.5: Surface reconstruction from partial contours using HRBF. (a) A visualization of the input point (blue dots) and normal (green vectors) constraints together with the resulting surface (red). (b) The three input partial contours and resulting surface in VolumeViewer.

with coefficients  $\alpha_1, \dots, \alpha_N, b \in \mathbf{R}$  and  $B^1, \dots, B^N, \mathbf{a} \in \mathbf{R}^3$ . These coefficients can be uniquely determined by enforcing the Hermite interpolation conditions above along with  $\sum_{j=1}^N \alpha_j = 0$  and  $\sum_{j=1}^N \{\alpha_j \mathbf{x}^j + B^j\} = 0$ .

In order to make use of this framework, we must provide as input a set of points and corresponding normal vectors which will be interpolated by the implicit surface. We obtain a set of points by regularly sampling the user-drawn contours, and we assign each sample point a normal by computing the cross-product between the normal to the drawing plane and the unit tangent vector at the given sample point on the contour. Intersection points between contours are handled as a special case. To get the normal at an intersection point, we consider the set of points that includes two contour points, the one immediately preceding and the one immediately following the intersection point, from each contour that shares the intersection point. We fit a plane to this set of points and assign the normal of the resulting plane to the intersection point.

We then fit a HRBF to the set of points and normals to obtain the final implicit surface. Note that the user-drawn contours need not be closed in this case, because we simply need a set of 3D points and normal vectors. The previous discussion

on contour consistency (see Section 4.3) is still relevant however, as providing an inconsistent set of points and normals would produce undesirable results.

In addition to allowing us to remove the restriction of closed contours, the use of HRBF implicits makes it easy for us to add data to guide the surface reconstruction in places where we do not have user input. For example, in areas where there is a strong image gradient, we can add additional normal constraints and replace the planar normals defined above with normal vectors in the direction perpendicular to the local gradient of the image data. This results in a surface that interpolates the input contours as well as follows the image gradient in the space between contours. For structures where the general shape of the structure is known, we can similarly add additional constraints based on the expected curvature. In this way, we are able to produce quality reconstructions from partial contours.

## 4.5 Chapter summary

In this chapter we have described the algorithms necessary to support nonparallel contouring. We looked at the problem of ensuring that a set of user-drawn contours is consistent, and we developed a validate-as-you-go contouring process that makes certain that the contours drawn by the user are always consistent. We developed tools for editing the position of intersection points and adjusting contours to meet the new locations naturally, using constrained Laplacian editing. We also presented a method of supporting partial contours through the use of HRBF implicits, and we demonstrated how to incorporate additional information such as image gradients and expected curvature into this model in order to guide the surface reconstruction through areas containing no explicit user input.

# Chapter 5

## Evaluation

### 5.1 Introduction

In the previous two chapters we described the design of an interface for nonparallel contouring and the development of the algorithms that support it. In this chapter we present user studies that help us understand how nonparallel contouring can aid the segmentation process and expose the challenges associated with a nonparallel contouring system in practice.

In Section 5.3, we describe a sequence of studies designed to answer the question: Are parallel-plane experienced users able to contour structures using nonparallel planes with similar consistency? The findings show that users are not naturally able to contour on the nonparallel planes with the same level of consistency, but that this problem can be effectively addressed by integrating reference images (see Section 3.4) into the interface. The results from this study heavily influenced the design of the VolumeViewer interface (see Chapter 3).

In Section 5.4, we describe a study designed to answer the question: Are parallel-plane experienced user able to segment structures consistently and more efficiently by using VolumeViewer to follow a nonparallel contouring protocol? This study provides evidence that nonparallel contouring can yield significant gains in efficiency, without a drop in consistency, over traditional parallel contouring techniques, but that care must be taken to choose an effective protocol for the given data set.

## 5.2 Related work

Early in the adoption of CT-based treatment planning segmentation errors were unknown but believed to be minor compared to the inaccuracy of localization and treatment delivery techniques of the day. Since then, the degree of segmentation error has been quantified for some segmentation tasks. Fiorino et al. [21] reported a study which examined inter-observer variability when contouring the prostate and seminal vesicles. They found that investigators differed by 1 cm in the cranio-caudal direction in four of six patients. Van de Steene et al. [18] examined the inter-observer variability in the definition of gross tumor volume (GTV) for lung cancer and found that the dimensions of the primary tumor varied by 4.2, 7.9, and 5.4 cm in the axial, cranio-caudal, and antero-posterior directions, respectively. Similar results have been found for nearly every anatomic site. Additionally, when compared to localization and treatment delivery inaccuracies, it is now known that segmentation error comprises a considerable portion of the treatment uncertainty [75].

Meanwhile, radiology capitalized on multislice CT and developed multiplanar reformation (MPR), a technique capable of generating 2D images of arbitrary image planes from a 3D image data set, to improve diagnosis and reduce inter-observer variability. Higashino et al. [30] investigated the utility of MPR for the assessment of non-small cell cancer and concluded that MPR improved diagnosis of chest wall lesions as well as decreased mean reading times. Sebastian et al. [65] found that investigators made additional findings and were more confident in their diagnosis when reviewing MPR images of the abdomen and pelvis than they were compared to axial images alone. They went on to state that MPR may become the preferred reading orientation due to its ability to improve efficiency and productivity. Other studies have shown that the use of MPR leads to improved diagnosis for multiple anatomic sites [41, 5]. To date there are no reports in the radiation oncology literature describing the impact of MPR on structure segmentation.

While observer variation in contouring structures on transverse image planes has been widely studied in the radiation oncology community [11, 22, 78], few studies have examined viewing or contouring on non-transverse planes. Steenbakkers et al. [71] found that users who do not reference coronal and sagittal planes have a higher variation in the superior and inferior directions when contouring, and thus have a



greater level of inter-observer variation. While no significant improvement in accuracy was measured, Petric et al. [55] found that contouring difficulty was decreased when using paratransverse (perpendicular to the long cervical axis) than transverse planes for cervical cancer segmentation. These two studies suggest that in addition to being more efficient, nonparallel contouring is easier and more accurate in some cases.

In this dissertation, we examine the ability of users, who have been trained to contour structures on transverse image planes, to contour structures on nonparallel image planes, and we evaluate the effectiveness of our VolumeViewer system at enabling users to segment structures using a nonparallel contouring protocol. In doing so, we aim to determine when and how nonparallel contouring can benefit the segmentation process.

### **5.3 User studies on the feasibility of nonparallel contouring**

To begin to examine the feasibility of nonparallel contouring, we sought to answer the question: Are parallel-plane experienced users able to contour structures using nonparallel planes with similar consistency? If users are able to do this while taking about the same time to draw a parallel contour as a nonparallel contour, then we should be able to improve the overall efficiency of the segmentation process in many cases by allowing the user to contour the structure with fewer nonparallel planes. We ran a series of three studies that we will refer to by what was being measured in each: “Study I: Inter-observer variability of experts”, “Study II: Intra-observer variability of experts”, and “Study III: Inter-observer variability of novices with reference images”.

#### **5.3.1 Method**

**Participants** Five experts and five novices were recruited to participate in the study. The experts consisted of four radiation oncology residents and one medical dosimetrist, all employed by the Washington University School of Medicine. Each had considerable experience segmenting the male pelvis and head-and-neck regions

with parallel contours. The novices were undergraduate or graduate students in the Computer Science Department at Washington University in St. Louis. None of the novices had any previous experience marking contours on medical images.

**Patient image data** Anonymized high-resolution CT scans from five pelvis and five head-and-neck cases were selected from a data archive of previously treated conformal radiation therapy patients. Each patient was scanned with a Philips Brilliance CT 64-slice scanner at the Washington University School of Medicine. The CT scans were reconstructed with 0.625 *mm* thick transverse slices, each having 512 x 512 pixels and corresponding in-plane pixel resolutions of 0.8 x 0.8 *mm*<sup>2</sup>.

**Procedure** We describe the procedure of Study I in detail, and then describe the other two in terms of how they differed from Study I.

**Study I: Inter-observer variability of experts** The expert users segmented one structure (prostate or brainstem) from each of the ten CT data sets using both parallel and nonparallel methods, for a total of twenty trials. Between eighteen and twenty-eight planes were used for the parallel trials, and four planes were used for the nonparallel trials. The parallel planes were chosen so as to match those currently used in clinical practice, and the nonparallel planes were chosen so that when combined with two or three of the parallel planes, they would capture the shape of the structure with as few planes as possible [2]. The order of the ten data sets was randomized, and the order of the parallel and nonparallel trials was alternated to compensate for any learning effect.

**Contouring tools** All the participants used identical workstations and contouring tools. The interface provided to the users was a much simplified version of the VolumeViewer interface (see Chapter 3). A picture of the setup for the user studies is provided in Figure 5.1. All surface reconstruction tools were disabled, as the focus was on 2D contouring. All navigation controls were disabled. Instead, the users were required to move back and forth between the planes using Next and Previous buttons. This is the same approach that is currently used in treatment planning systems to

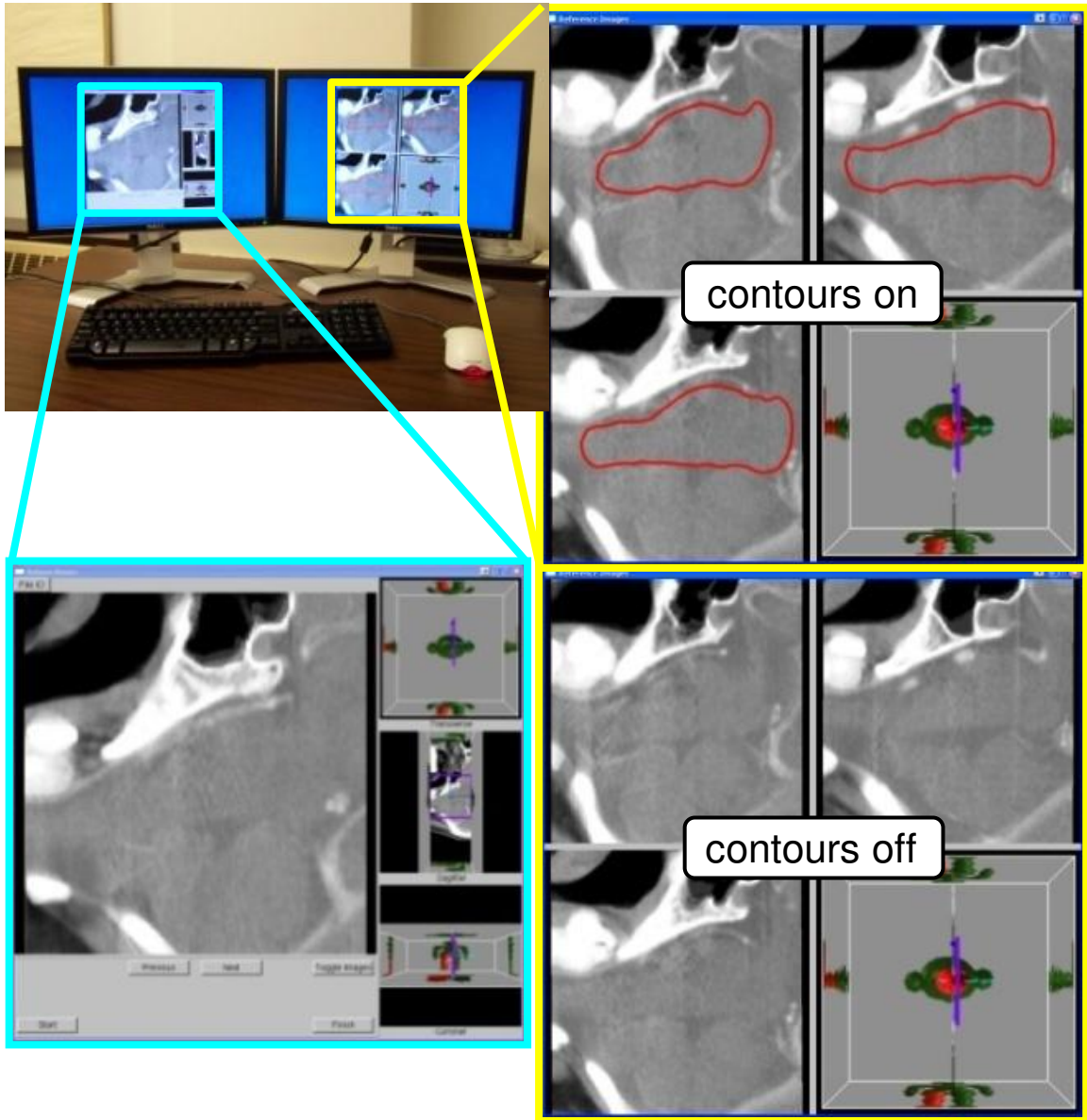


Figure 5.1: User study setup. The main window is highlighted in cyan. This is where the drawing takes place, and was the only window available in Study I and Study II. The window highlighted in yellow contains the reference images (see Section 3.4) used in Study III, and we show each of the two possible views. The contours can be toggled on (top) or off (bottom) to train the user to look for cues in the image data.

page back and forth through the slices in the stack. We also chose to only display the contour corresponding to the current image plane. For the nonparallel planes, we could have also displayed the intersection points of previously drawn contours with the current plane. However, this would have influenced what the user would have drawn and would have caused the order in which the planes were presented to affect the results. Instead, we wanted all users to draw on the same planes with the same information.

**Training and duration** The only training the users received was a brief set of presentation slides which gave an overview of the study and introduced the VolumeViewer interface. The users were allowed to read through the slides at their own pace. They were given an example data set, not included in the study, on which they could practice, and were encouraged to ask any questions they had about the operation of the interface at this time. No user spent more than fifteen minutes on the training. The total duration of the user study, including training together with trials, was capped at ninety minutes. Users were allowed to take short breaks in between trials if needed. No user failed to complete all trials within the allocated time.

**Study II: Intra-observer variability of experts** Approximately two weeks after Study I, we invited the same expert users back to repeat the study. Four of the five expert users participated. The study design was exactly the same as before, except that the users were asked to segment one structure from just four (randomly chosen) of the ten original data sets (two prostate, two brainstem). They again used both parallel and nonparallel methods for a total of eight trials.

**Study III: Inter-observer variability of novices with reference images** Although the data sets and design for this study were largely the same as those used for the intra-observer study, this study was entirely different because of the difference in users and the incorporation of reference images into the interface. These users were *novices*, having no prior experience whatsoever segmenting medical images. The only knowledge and training they received is what we provided to them in the interface.

In addition to the difference in users, we only had the users contour on the non-parallel planes (four trials total), and we added one new window to the interface that was not available in the previous studies. This new window is highlighted in yellow, with two possible views, in Figure 5.1. For each nonparallel plane, the window contained four reference images (see Section 3.4). Three of these images show example contours drawn on similar cross-sections of the anatomy of three different patients. These contours were made by intersecting the nonparallel planes with a surface model reconstructed from many physician-reviewed contours drawn on the parallel cross-sections of the data set. The contours can be toggled on and off, to give the user a sense of what they should be looking for in the image data. The fourth image illustrates which cross-section the user is currently observing, by showing its position as a slicing plane through a surface model of the anatomy.

**Measures** Our primary source of data is the contour data produced by our users. We ran a total of 152 trials (Study I: 5 users, 20 trials; Study II: 4 users, 8 trials; Study III: 5 users, 4 trials). Each trial had the user mark between 4 and 28 contours. We also recorded video screen captures of the users performing the segmentations, and we recorded the total time from start to finish for each trial.

We computed the mean distance and percentage overlap between each pair of user-drawn contours for each plane in each case. For the parallel cases, we chose a sample of four planes that were evenly spaced through the data set. For inter-observer variation, we compare the users against one another, and for intra-observer variation, we compare each users' contours with the contours that they drew in the previous study.

The mean distance [14] (see Section 2.1.1) was a two-sided mean distance computed as the average minimum distance between point samples on the first contour to the second contour and from point samples on the second contour to the first contour. The percentage overlap, or Jaccard index [36], was computed as the ratio of the number of pixels contained in the intersection of the two contours to the number of pixels contained in their union. The mean distance measure captures the average distance from one contour to another, while the overlap measure normalizes for the

size of the contour. We then averaged these numbers across all cases for the prostate and the brainstem. See Figure 5.5 and Table 5.1 for these results.

### 5.3.2 Results

For Study I, sample images of the results from the brainstem case A and of the prostate case B are shown in Figure 5.2. The contours of each of the five users are displayed in a different color. Colored dots on the nonparallel planes represent the intersection points of the plane with contours drawn on the other nonparallel planes. These images show that the variation among users was much larger in the nonparallel cases than in the parallel cases.

For Study II, sample images of the results from the brainstem case A and of the prostate case B are shown in Figure 5.3. The results illustrate that intra-observer variability was much larger for nonparallel contours than for parallel contours. In general, the users did not draw highly similar contours to what they had drawn before on the nonparallel planes. Also, there was no significant improvement in consistency between the user-drawn parallel or nonparallel contours from Study I to Study II.

For Study III, sample images of the results from the brainstem case A and of the prostate case B are shown in Figure 5.4. Given the reference images, the novice users were much more consistent at marking contours on the nonparallel planes than the experts.

The quantitative results verify what is observed in the images. The results are shown in Figure 5.5, and corresponding p-values are provided in Table 5.1. Comparisons between parallel, nonparallel, and nonparallel reference image trials were carried out using Friedman's two-way non-parametric analysis of variance followed, if significant at the 0.05 level, by Wilcoxon signed ranks tests. The Bonferroni correction was applied to correct for repeated measures, thus a p-value below 0.017 indicates that the two groups do differ significantly. This analysis further confirms that the variation among users was significantly larger in the nonparallel cases than in the parallel cases for Study I and Study II. With the benefit of the reference images, the novice users were much more consistent on the nonparallel planes in Study III. While the mean

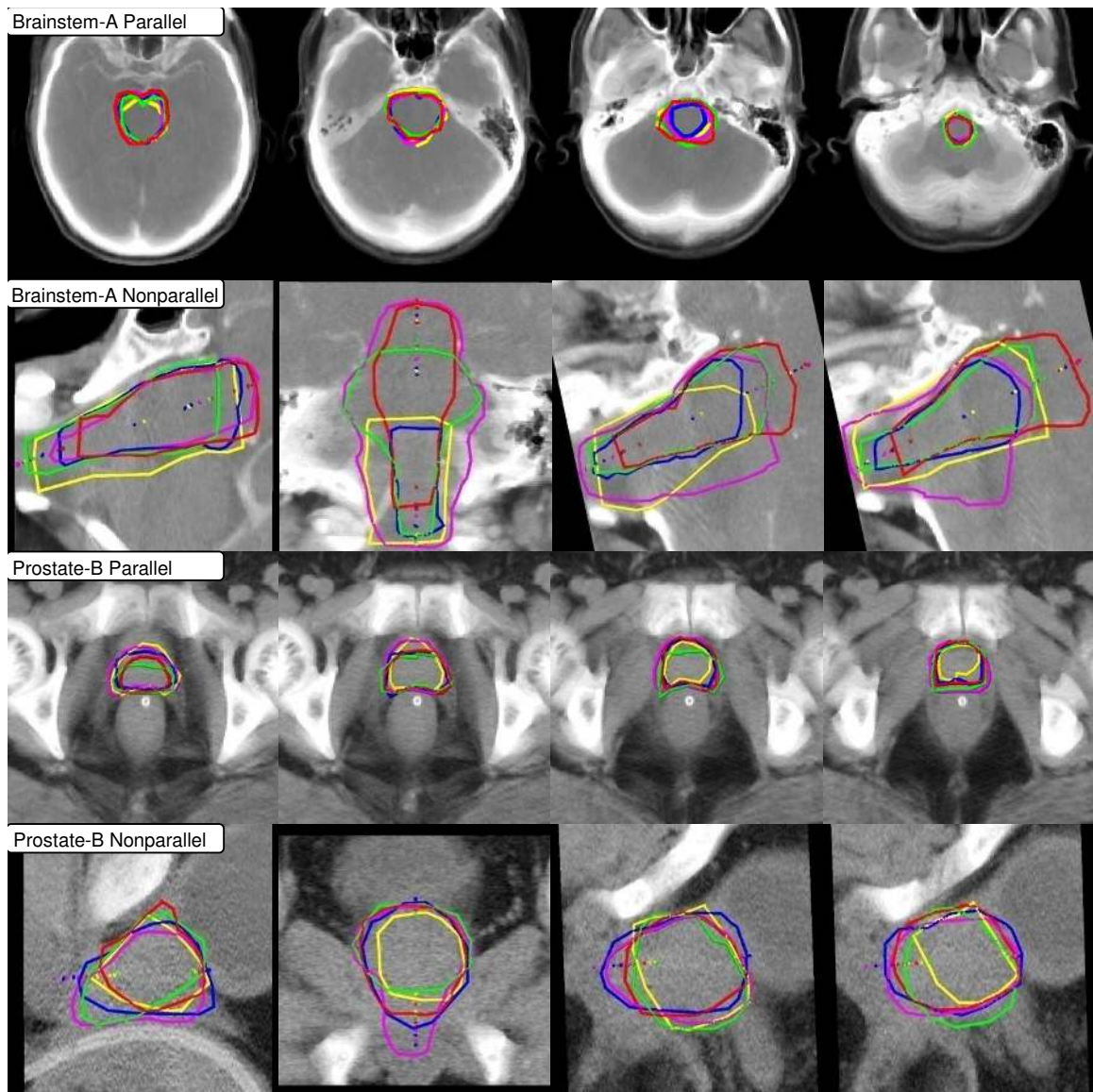


Figure 5.2: Inter-observer variability of experts. Each row contains four cross-sections from the following cases: (Top row) Brainstem-A Parallel, (Second row) Brainstem-A Nonparallel, (Third row) Prostate-B Parallel, (Bottom row) Prostate-B Nonparallel. The contours of each of the five users are displayed in a different color. Colored dots on a nonparallel plane represent the intersection points of the other contours with that plane. Note that the variation among users is larger in the nonparallel cases than in the parallel cases. The users were also largely inconsistent with the other contours that they drew in the nonparallel cases.



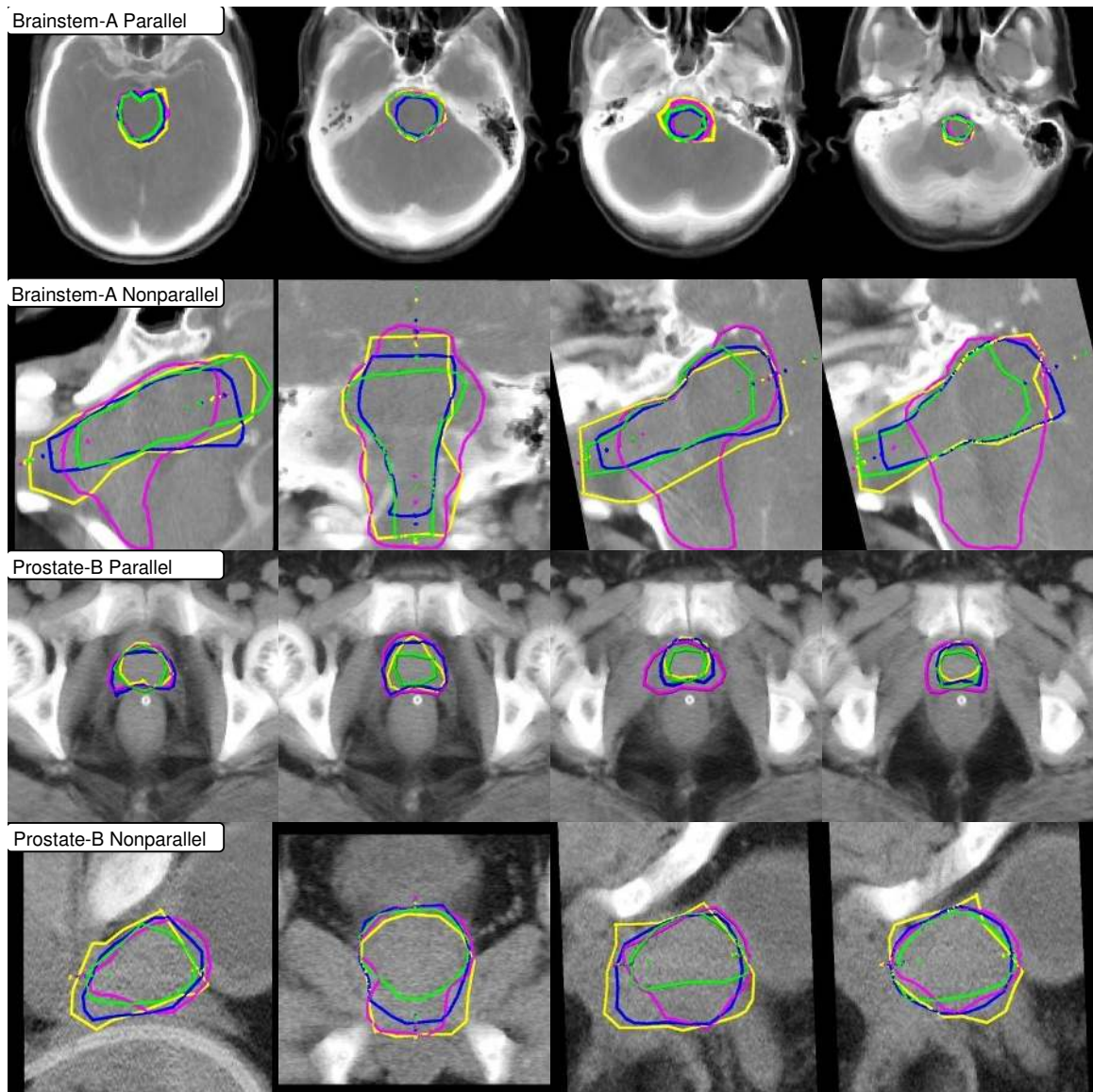


Figure 5.3: Intra-observer variability of experts. Each row contains four cross-sections from the following cases: (Top row) Brainstem-A Parallel, (Second row) Brainstem-A Nonparallel, (Third row) Prostate-B Parallel, (Bottom row) Prostate-B Nonparallel. The contours of each of the four users are displayed in a different color. The same color-scheme was used as in Figure 5.2. Colored dots on a nonparallel plane represent the intersection points of the other contours with that plane. Note that the variation among users was again larger in the nonparallel cases than in the parallel cases. The users were also largely inconsistent with what they had drawn in Study I (see Figure 5.2).



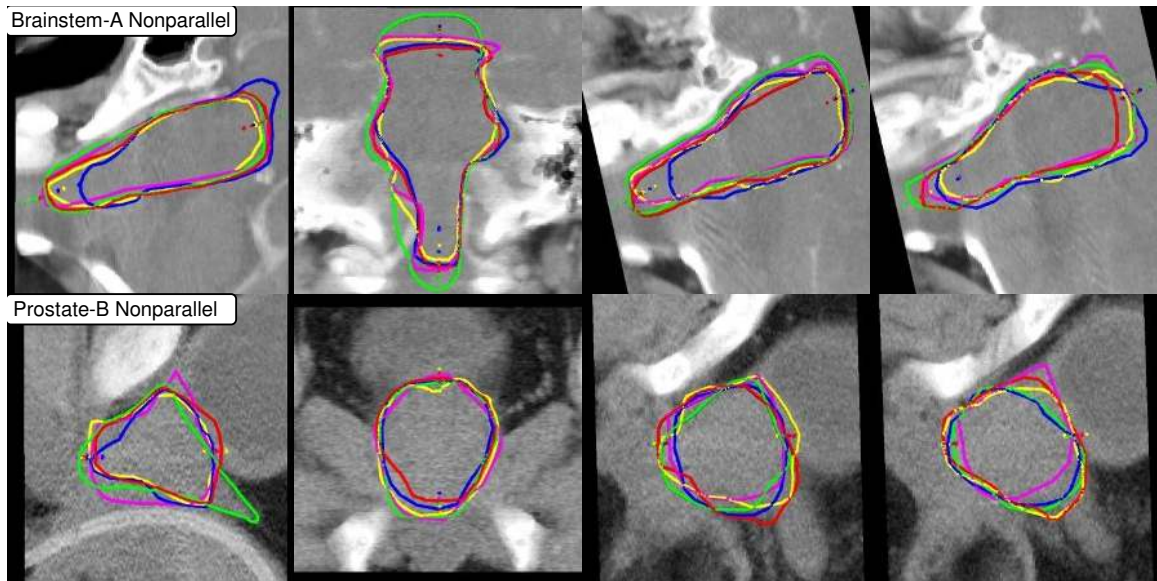


Figure 5.4: Inter-observer variability of novices with reference images. Each row contains four cross-sections from the following cases: (Top row) Brainstem-A Nonparallel, (Bottom row) Prostate-B Nonparallel. The contours of each of the five users are displayed in a different color. Colored dots represent the intersection points of the other contours with that plane. Note that the variation among users is smaller than in the previous studies (see Figures 5.2 and 5.3). The users were also more consistent with the other contours that they drew in the nonparallel planes.

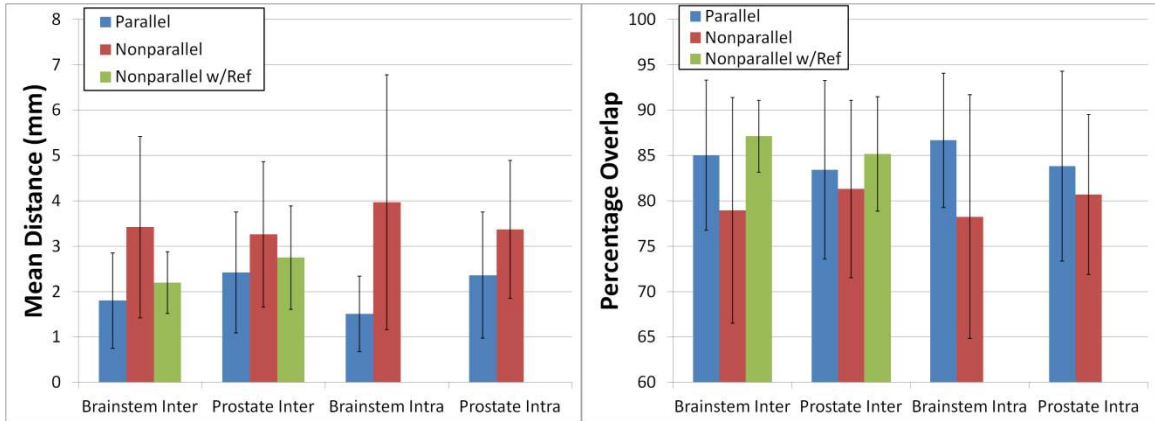


Figure 5.5: The average mean distances (left) and average percentage overlap (right) between user drawn contours. “Inter” and “Intra” refer to the results from the inter-observer (Study I and Study III) and intra-observer (Study II) studies, respectively. “Nonparallel w/Ref” refers to the results from novices drawing on nonparallel planes with reference images (Study III). Intra-observer variation was not measured for this case. Note that the average distance between user drawn contours was higher and the percentage overlap lower for the nonparallel cases than the parallel cases. The average distance between nonparallel contours drawn by novices using reference images was lower and the percentage overlap higher than for the nonparallel contours drawn by experts without reference images.

distances for the brainstem case were still larger than that of the expert-drawn parallel contours, the mean distances for the prostate case and the average percentage overlaps were not significantly different. By and large, the novice users working with reference images were able to contour the nonparallel planes with similar consistency to that of the expert users contouring parallel planes.

### 5.3.3 Discussion

Study I and Study II showed that the expert users were much more consistent at drawing parallel contours than nonparallel contours. We can use several of our working hypotheses (see Section 2.3.3) to explain this negative result. First, the users had considerable experience interpreting imaging data using the parallel planes (H4), as this is the current standard of practice in radiation oncology. Yet, the interpretation of nonparallel image planes was limited to a brief tutorial and two example cases.

Case	Comparison	Mean	Overlap
Brainstem Inter	Parallel v. Nonparallel	0.000	0.000
	Nonparallel v. w/Ref	0.000	0.000
	Parallel v. w/Ref	0.000	0.049
Prostate Inter	Parallel v. Nonparallel	0.000	0.533
	Nonparallel v. w/Ref	0.000	0.204
	Parallel v. w/Ref	0.111	0.091
Brainstem Intra	Parallel v. Nonparallel	0.000	0.009
Prostate Intra	Parallel v. Nonparallel	0.004	0.082

Table 5.1: P-values from Wilcoxon signed ranks tests on the data presented in Figure 5.5.  $\alpha = 0.017$  was used as the level of statistical significance.

Users were given no additional imaging or spatial location information to assist with nonparallel image planes.

In addition, for the brainstem and prostate, the parallel contours were generally smaller and more symmetric than the nonparallel contours (H2). The image data is noisy, making it very difficult to delineate the boundary, but if the user is able to delineate part of the boundary in the parallel cases, the rest can often be completed by symmetry. This is not the case for the irregular shaped nonparallel contours in our examples.

Finally, the Next and Previous buttons provide more information for the parallel cases than the nonparallel. For the parallel cases this corresponds to paging forward and backward continuously through the image stack (H5). One parallel contour can be used as a template for the next parallel contour in the stack, and paging back and forth quickly through the parallel slices can give the user a rough idea of what the surface would look like (H7). This is not the case for the nonparallel contours. The next nonparallel plane is at a completely different location and orientation from the one before it.

The results from Study I and Study II imply that some form of additional training for drawing contours on nonparallel image planes is required in order to achieve consistency between users. Study III investigated the use of reference images as one possible training mechanism. The results of Study III showed that novice users working with reference images were much more consistent drawing nonparallel contours than the

expert users were without reference images, and they were nearly as consistent drawing nonparallel contours as the expert users were drawing parallel contours.

**Limitations** Though we took measures to replicate the environment of the clinic as much as possible, it is impossible to exactly mimic clinical conditions. The stakes are different. We asked our users to treat this just like they would a real patient, but they knew that the contours would not be used for treatment. They also knew that their names would not be attached to the results in any way. We tailored our VolumeViewer interface to have the same basic functionality as those used in the clinic, but it is still not the same interface.

These results may not generalize to other data sets that may exhibit different properties. While we used real patient data sets and asked users to segment the prostate and brainstem, which are two of the most commonly segmented structures in radiation oncology, our results may not necessarily generalize to other structures or data sets that differ significantly in terms of their properties, such as boundary strength, size, or shape (see Section 2.2).

The participants for Study III were limited to novice users. We hypothesize that if we were to run this study again on the expert users with the use of reference images, that we would see similar results. However, we have chosen not to do this largely due to signs of user fatigue after the intra-observer study. Ideally, we would invite the novice users back to repeat the study in order to verify the consistency results that we observed and to measure the intra-observer variability of the novices.

It should also be noted that we will not have the ability to create reference images for every structure that might need to be segmented. For example, tumors are often irregular in shape and therefore difficult to characterize with a set of reference images. However, in this case, the difference between parallel and nonparallel contouring may not be so large, as users are unlikely to be any more familiar at looking at the irregular shape of the tumor in a parallel view than a nonparallel view.

**Design implications** The main implication of this study is that users with experience contouring medical images on parallel planes are not able to contour nonparallel

planes with the same level of consistency, at least not without some additional training or enhancements made to the interface to support nonparallel contouring. Our study suggests that the integration of reference images into the interface is one viable way to address this issue. Other ideas that we have implemented to further address this issue include: giving the user the ability to manipulate the image plane (see Section 3.3), displaying the surface reconstruction (see Section 3.3), animating the image plane (see Section 3.4), displaying contour intersection points (see Section 3.4), keeping the region of interest centered and the same size on the screen (see Section 3.4), and adding visualization cues that help convey 3D information (see Section 3.5). However, the efficacy of these ideas have not been tested in this study.

## 5.4 System evaluation

The results from the studies presented in the previous section highlighted the challenges associated with nonparallel contouring and informed the redesign of our VolumeViewer interface (see Chapter 3). We evaluate the effectiveness of our system by comparing the ability of radiation oncologists to reconstruct surfaces using the proposed nonparallel contouring system against traditional systems. The main question that we wanted to answer was as follows: Are parallel-plane experienced users able to segment structures consistently and more efficiently by using VolumeViewer to follow a nonparallel contouring protocol?

In contrast to our previous studies (see Section 5.3), users did not simply contour on a given set of planes, but used our system to sketch a set of consistent contours, build a surface model that interpolates those contours, review the surface model together with the image data, and make any necessary edits to produce a segmentation. We compare parallel and nonparallel segmentation techniques in terms of efficiency and the consistency of the segmentations produced.

### 5.4.1 Method

**Participants** We recruited seven radiation oncology residents, all employed by the Washington University School of Medicine, to participate in the study. Each had considerable experience segmenting the prostate and liver. None of these users had participated in Study I or Study II or had any previous experience with the VolumeViewer interface.

**Patient image data** Anonymized high-resolution CT scans from two prostate and one liver cases were selected from a data archive of previously treated conformal radiation therapy patients. The two prostate cases were the same as those used in our previous studies (see Section 5.3). Each patient was scanned with a Philips Brilliance CT 64-slice scanner at the Washington University School of Medicine. The CT scans were reconstructed with 0.625 *mm* thick transverse slices, each having 512 x 512 pixels and corresponding in-plane pixel resolutions of 0.8 x 0.8 *mm*<sup>2</sup>. Example images from both the prostate and liver cases are provided in Figure 5.6.

**Procedure** The users segmented the two prostate data sets using our nonparallel contouring system and the liver data set using both our nonparallel contouring system and traditional parallel contouring techniques. In order to avoid user fatigue and keep the length of the study under one hour for each user, we did not have the users segment the prostate data sets using traditional parallel contouring, but instead we compare the results to the parallel contouring data obtained in Study I (see Section 5.3).

For the nonparallel contouring trials, we asked the user to construct an initial surface by following a protocol (see Section 3.4) that consisted of seven planes for the prostate cases (see Figure 3.3a) and eight planes for liver cases (see Figure 5.7c). For the parallel contouring trials, the user was asked to construct an initial surface by following a traditional “parallel protocol”, marking a contour on every other parallel slice in the data set, as is commonly done in practice for the liver case (see Figure 5.7a). For both the parallel and nonparallel trials, after constructing an initial surface, the user was asked to review the surface along with the image data and optionally make any edits by adding additional contours or adjusting existing ones.

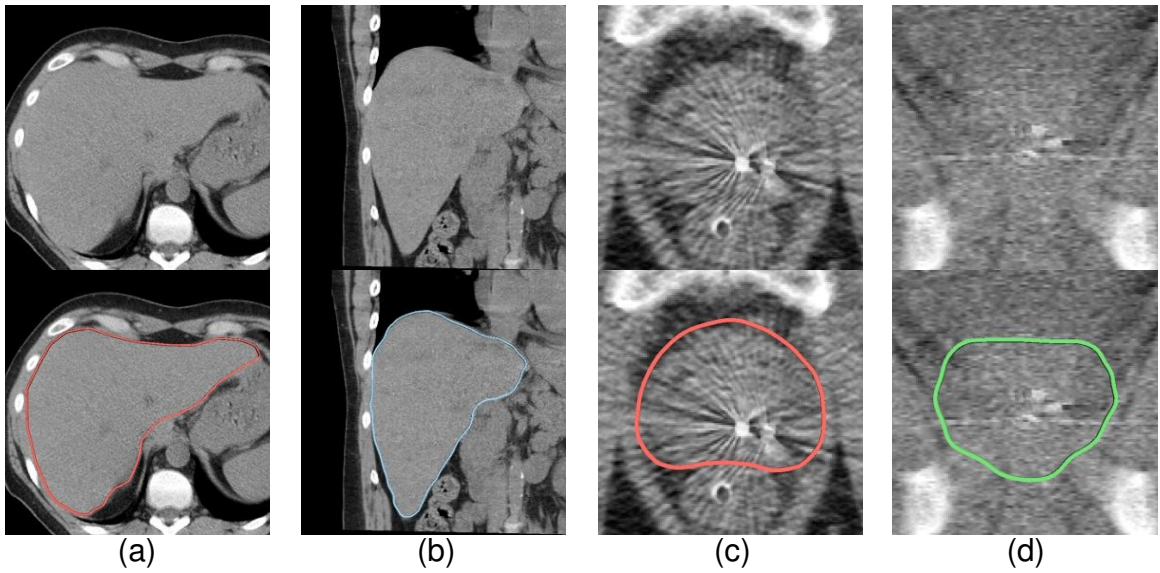


Figure 5.6: Example images from prostate and liver data sets. The top row shows example images from the four cases in the study: (a) parallel liver (b) nonparallel liver (c) parallel prostate (d) nonparallel prostate. The bottom row shows the same images together with the corresponding contours. Note the differences in boundary strength between the liver and prostate data sets as well as between the parallel and nonparallel prostate cases.

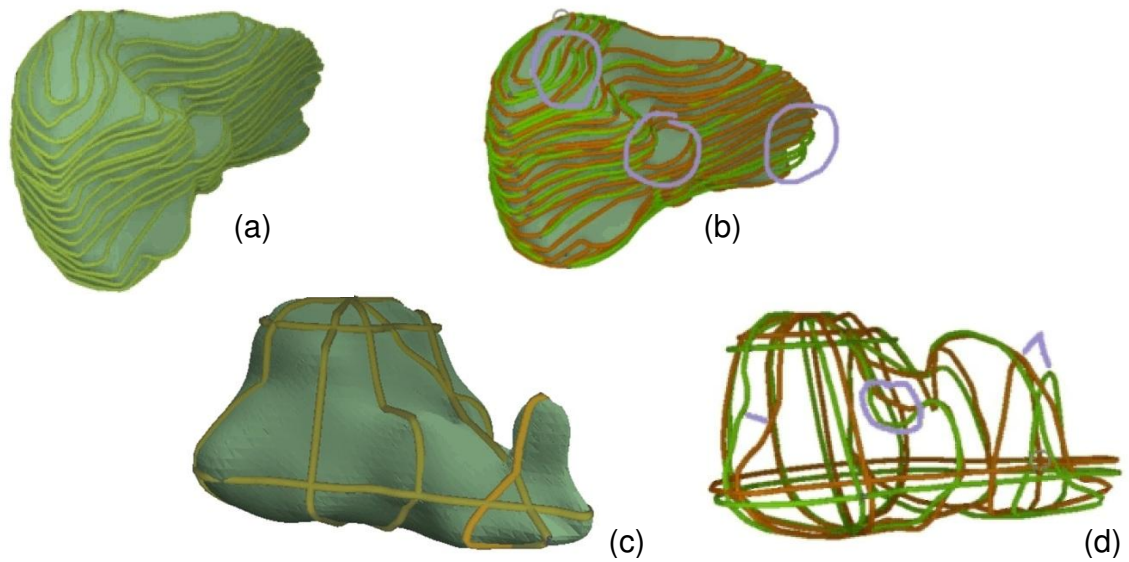


Figure 5.7: Comparison of user contours for parallel and nonparallel liver cases. Users contoured the liver data set using both a parallel (a) and nonparallel (c) protocol. The contours drawn by two different users are displayed together for comparison in the parallel case (b) and nonparallel case (d). The contours displayed in green were drawn by one user and those displayed in brown by another. Inconsistencies are highlighted in purple.



**Contouring tools** All the participants used identical workstations and contouring tools. The setup for the user study was similar to the previous studies (see Figure 5.1). The difference is that the interface provided to the users in this study was a full-featured version of the VolumeViewer interface (see Chapter 3). All navigation controls and surface reconstruction tools were available. Reference images were provided in all trials. The only difference between the parallel and nonparallel trials was in the protocol that was used to construct the initial surface.

**Training and duration** The only training the users received was a brief demonstration of using the VolumeViewer interface to segment an example data set, not included in the study. The focus of the demonstration was solely on the operation of the interface and not on contouring technique. The users were then allowed to practice using the interface with the example data set themselves, and were encouraged to ask any questions they had about the operation of the interface at this time. No user spent more than fifteen minutes on the training. The total duration of the user study, including training together with trials, was capped at sixty minutes. Users were allowed to take short breaks in between trials if needed. No user failed to complete all trials within the allocated time.

**Measures** We recorded both the contours drawn and the surface models created from them by our users. We also recorded the user interaction and timing information for each trial by recording a time stamp for each button press and labeling it with the associated action being taken in the user interface. We had seven users perform four trials each for a total of 28 trials.

We compare the consistency of the segmentations produced by starting from a traditional parallel contouring protocol to those produced using our nonparallel protocol. We performed pairwise comparisons within each group, computing the mean percent distance and Dice’s coefficient [19] between each pair of segmentations (see Section 2.1.1 for a further description of these measures). The mean *percent* distance is simply the mean distance [14] divided by a scalar value that corresponds to an estimated size of the structure. This has the effect of normalizing the distance measures across structures of varying size. We also compared the efficiency of the segmentation techniques by computing the average time it took to segment each structure.

## 5.4.2 Results

Examples of the image data for each of the four cases and the contours that were drawn on them are provided in Figure 5.6. Note the differences in properties between the data sets. In particular, the boundary strength of the liver data set is much stronger than that of the prostate for both parallel and nonparallel cases, and the parallel prostate image exhibits stronger boundaries than the nonparallel case.

An example of the set of user drawn contours and the types of inconsistencies observed between users is shown for both the parallel and nonparallel liver cases in Figure 5.7b,d. Since users may not draw on the same set of planes, we do not measure the difference between individual contours, but rather the difference between the 3D segmentations.

The efficiency and consistency results are shown in Figure 5.8. For the liver, the average contouring time for the nonparallel case was nearly half of what it was for the parallel case. The distance and Dice's coefficient consistency measures were similar, but with a smaller standard deviation in the nonparallel case. This indicates that the users were able to segment the liver with similar consistency and in nearly half the time by using the nonparallel protocol.

For the prostates, the contouring time was less for one parallel case than the other which had more planes. In contrast, the time was the same for both nonparallel cases. However, the time per contour for the parallel cases was nearly half that of the nonparallel cases, resulting in lower total contouring times since neither of the parallel cases had more than twice the number of contours as were in the nonparallel protocol.

The mean distance measures for the prostates were much higher than that of the liver, and were even higher in the nonparallel case. Similarly, the Dice's coefficient was much lower for the prostates than the liver, and were even lower in the nonparallel case.

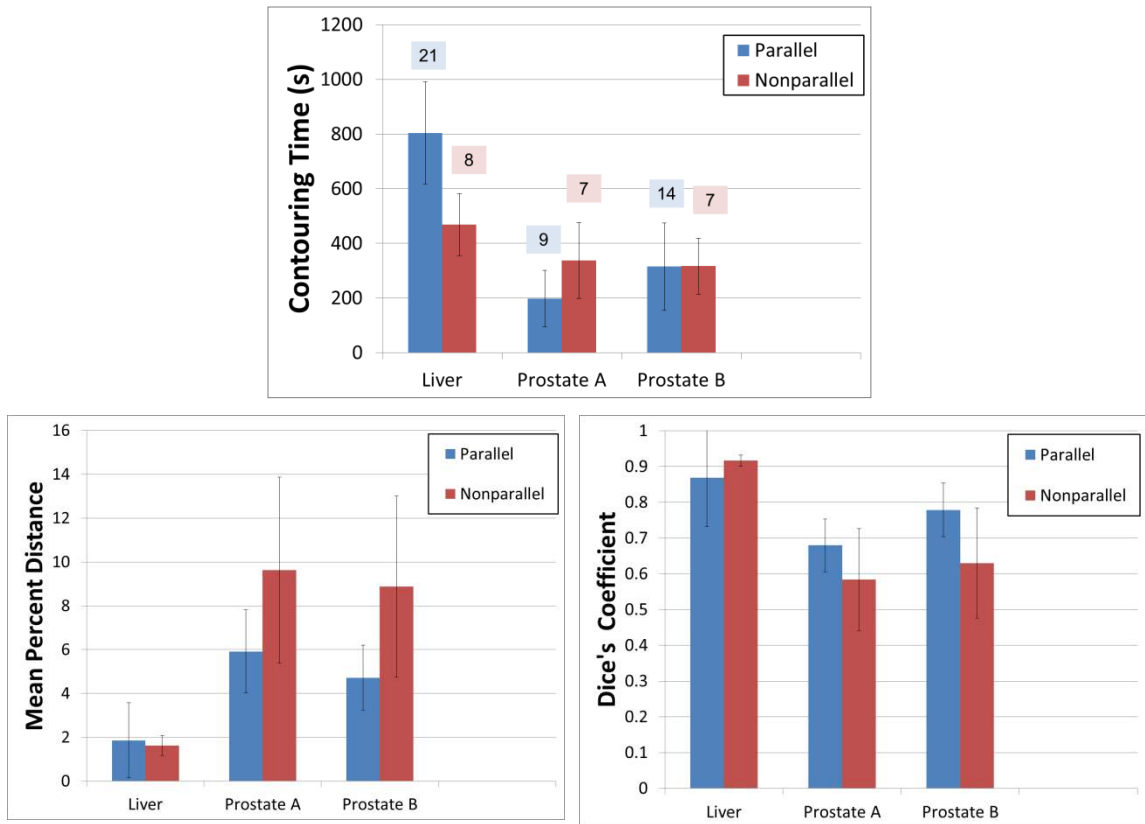


Figure 5.8: Efficiency and consistency of user segmentations. Graphs of the average contouring time (top), average mean percent distance (bottom left), and average Dice's coefficient (bottom right) are shown. For the contouring time graph, numbers in the shaded boxes above the bars are the average number of contours drawn.

### 5.4.3 Discussion

This study showed that the efficiency of the segmentation of the liver can be increased dramatically, and without a loss of consistency, by contouring with a nonparallel protocol as opposed to traditional parallel planes. However, a similar increase in efficiency was not observed for the prostate cases. In fact, users were no more efficient and less consistent when using the nonparallel protocol for the prostates. We can use several of our working hypotheses (see Section 2.3.3) to explain these mixed results.

The results from the liver case demonstrate that gains in efficiency can be had by allowing the user to contour on a sequence of nonparallel planes that increase the accuracy of the segmentation more quickly than the standard sequence of parallel contours (H1). We predict that we would see similar results for a prostate case that was larger and therefore required more parallel contours. We expect that the contouring time for a larger prostate would continue to increase linearly with each additional parallel contour while the contouring time would remain roughly the same for the seven-contour nonparallel protocol, as it did for the two prostate cases in the study.

The boundary strength was much stronger in the liver cases than in the prostate cases (see Figure 5.6), and we believe this is largely responsible for the users being much more consistent with the liver segmentations than the prostate segmentations (H3). When the boundary strength is weak, such as in the prostate cases, it becomes increasingly important for the users to have familiarity with those planes in order to create consistent results (H4). Users were much more familiar with the parallel planes than the nonparallel planes. Furthermore, the boundary strength in the parallel prostate images was stronger than that in the nonparallel prostate images (see Figure 5.6c,d), causing users to be less efficient and consistent in the nonparallel prostate cases (H3).

**Limitations** Ideally, we would have had the users segment the prostate cases using traditional parallel contouring in addition to the nonparallel contouring protocol. Instead, we compared the nonparallel contouring results to the parallel contouring results from Study I. Those segmentations were performed on the same data sets, but

by a different group of expert users. While we expect that the results for this group of users would be similar, we cannot know for sure.

As in our previous studies (see Section 5.3), we face similar limitations in terms of mimicking clinical conditions. Our users knew that their segmentations would not be used to treat patients and that their names would not be attached to the results in any way.

We ran trials with one liver data set and two prostate data sets. These results may not generalize to other data sets that differ significantly in terms of their properties, such as boundary strength, size, or shape (see Section 2.2). Indeed, in this study we observed significant differences in the results between cases that we attribute largely to a difference in the boundary strength of the image data in those cases.

We only tested one eight-contour nonparallel protocol for the liver data set and one seven-contour nonparallel protocol for the prostate data set. The results will also depend heavily on the choice of protocol. While we aimed to choose a small set of planes that would sufficiently capture the shape of the structure [2], a protocol that uses a different set of planes or even a different ordering of the planes might perform better or worse in practice. Determining the best protocol to use for a given data set is a ripe area for future research.

**Design implications** The primary implication of this study is that the use of a nonparallel contouring protocol for segmentation has the potential to yield significant gains in efficiency over traditional parallel contouring techniques, as seen in the liver case. The caveat is that the simple use of a nonparallel contouring protocol does not guarantee improvements over parallel techniques, and may in fact be worse, as seen in the prostate cases in this study in which the choice of nonparallel planes resulted in images with weaker boundaries than the parallel planes. Care must be taken to choose an effective protocol for a given data set. Automatically determining such a protocol remains an open problem. We describe one approach in Section 6.1.1.

## 5.5 Chapter summary

In this chapter we presented the results from a sequence of user studies that are a step in the direction of understanding how nonparallel contouring can aid the segmentation process. We described our initial investigation into the ability of parallel-plane experienced users to contour structures using nonparallel planes with similar consistency. We showed that the inter-observer and intra-observer variability of nonparallel contouring by expert users is much higher than for parallel contouring. We have presented our hypotheses as to why this disparity exists, and have shown that the consistency can be greatly improved in novice users simply by integrating a collection of reference images into the interface.

We used the results from these initial studies to inform the redesign of our VolumeViewer interface, and then we tested the ability of parallel-plane experienced users to segment structures consistently and more efficiently by using VolumeViewer to follow a nonparallel contouring protocol. We found that nonparallel contouring has the potential to yield significant gains in efficiency over traditional parallel contouring, but that such results are dependent on choosing an effective protocol for the data set.

# Chapter 6

## Conclusion and future work

Image data set interaction and segmentation in radiation oncology revolves around the 2D interpretation of 3D volume data which has been limited to parallel image planes. This convention is not based upon scientific consensus but rather limitations of early imaging and computing technology. New surface reconstruction algorithms have been developed that allow surfaces to be built from nonparallel contours. The use of nonparallel contours has the potential to increase the efficiency and accuracy of the segmentation process. They grant the user the freedom to contour on the planes that will increase the accuracy of the segmentation as much as possible, and that will be the easiest to contour by choosing the planes with the simplest shapes and the strongest boundaries. This dissertation has taken the first steps toward bringing nonparallel contouring to use in practice.

In order to study nonparallel contouring, we first had to build a system to enable it. We developed VolumeViewer, a complete nonparallel contouring system, that allows a user to navigate through a volumetric data set, sketch contours onto planar cross-sections, and reconstruct a surface model from those contours. We encountered several challenges to making nonparallel contouring feasible in practice, including users lacking familiarity with the nonparallel planes, the difficulty in determining which planes to contour, and the challenge of navigating from one plane to the next while maintaining a sense of orientation. To address these issues, we developed contouring protocols for segmenting specific structures, incorporated reference images into the interface, and made visualization enhancements to convey the 3D structure of the segmentation as it is being constructed. We also made algorithmic developments for supporting our nonparallel contouring system, including methods for ensuring the

consistency of user-drawn contours and extending an existing reconstruction algorithm to add support for partial nonparallel contouring.

We evaluated our system by studying the ability of medical professionals to segment structures from real patient data . We began with an investigation into the ability of users that have been trained to segment structures using parallel planes to segment the same structures using nonparallel planes. We found that the contouring variation was significantly higher for the nonparallel planes, but that this variation could be improved by integrating a set of reference images into the interface. We used our findings from these initial studies to improve the design VolumeViewer. We then evaluated our end-to-end system by comparing the ability of medical professionals to segment structures by using VolumeViewer to follow a nonparallel contouring protocol as compared with traditional techniques. We found that nonparallel contouring can yield significant gains in efficiency over traditional parallel contouring techniques, but that care must be taken to choose an effective protocol for the given data set.

By studying the contouring process in this manner, we have taken the first steps to understanding how nonparallel contouring can aid the segmentation process. We have arrived at a conceptual framework that outlines the factors that influence the effectiveness of a segmentation method and includes several working hypotheses on the design of effective contouring systems.

Our work on nonparallel contouring has exposed several challenges that we intend to address in the future. We now present these challenges in the context of our work and outline possible solutions.

## **6.1 Future work**

### **6.1.1 Plane selection and protocol authoring**

In this dissertation, we developed an interface that allows the user to build surfaces from volume data by sketching contours on arbitrarily oriented cross-sections of the volume. When building surfaces in this manner, we tried to choose planes that captured the major features of the object while using as few planes as possible. Through



experimentation, we found a set of nonparallel planes for the prostate and liver cases that were generally sufficient at capturing the shape of those structures, and we then developed protocols for moving from one plane to the next and guiding the user to perform the segmentation. The importance of choosing an effective protocol for a given data set was evidenced by our evaluation. However, no formal investigation has been conducted to determine what makes a “good” set of planes for reconstructing an arbitrary structure. We intend to develop semi-automated tools for selecting a set of planes for a structure and authoring a protocol. Criteria to consider when choosing planes include planes that pass through feature points of the structure, planes that contain strong gradients in the image data, and planes that are orthogonal to previously selected planes.

**Geometrical approach** One approach is to start with a collection of surface models for commonly segmented anatomical structures, and, for each structure consider, what are the best planes to use to reconstruct the original surface? Note that in this case, we know the ground truth, so we can measure accuracy. How many contours are needed to achieve 90 percent accuracy? 95 percent accuracy? How many are needed if we restrict ourselves to parallel contours? We expect that the answers to these questions will further make the case for nonparallel contouring.

There are an infinite number of planes, so we cannot exhaustively try all possibilities. However, we can automatically generate contours from the original surface and try a large subset of them. Given a base surface and a number of potential candidate contours, can we develop an effective strategy for choosing the next best contour to add?

**Filtering candidate contours** Ideally, when considering which contour to add next, we would try all candidate contours, reconstructing a new surface for each one. We would then compare all of these surfaces to the ground truth, and choose to add the contour that results in the surface that is closest to the ground truth. However, surface construction and comparison are expensive operations. We need a method for filtering candidate contours so that we can consider a large number of contours, but only perform the surface construction and comparisons with the contours that are likely to have the largest impact.

One simple metric to consider for contour filtering is the distance from the candidate contour to the base surface. A candidate contour with a small distance to the base surface is likely to have a much smaller impact than a candidate contour with a large distance to the base surface. Another potentially useful metric is the distance from each vertex on the base surface to the contour set (the set of contours used to construct the base surface together with the candidate contour). This metric tells us something about how well the candidate contour covers portions of the base surface that were not covered by other contours. A smaller distance, in this case, indicates better coverage. Other metrics are possible, and for the distance computations, we can consider using total distance, average distance, or maximum distance.

Such metrics for filtering candidate contours will need to be tested and verified. This can be done by considering a number of different base surfaces and candidate contour sets. In each case, all proposed metrics can be applied and used to rank the candidate contours in order of which contour should be added next. These rankings can then be checked against the ground truth ranking, found by constructing the surfaces for all candidate contours and comparing to the ground truth surface. We expect that the most effective ranking is likely to be a weighted combination of the metrics proposed above.

**Measuring other factors** The approach proposed above will give us a good idea of what are the best planes to choose based solely on the geometry of a structure. However, due to other factors discussed in our conceptual framework, these may not be the best planes on which to contour. For example, we know that boundary strength and contour complexity play roles. But how much of a role? How weak of a boundary is too weak? How much does contour size influence the results? Shape complexity? We intend to design experiments that will allow us to quantify how and to what extent these different factors influence contouring.

We can use the data that we have collected in our user studies (see Chapter 5) as a starting point in analyzing the influence of these factors on contouring. We can measure boundary strength as the magnitude of the gradient of the image in a narrow band around the region where the users marked contours. We can measure contour length as the perimeter of the polyline that defines the contour, and we can

measure shape complexity of the contour using a variety of metrics for estimating the complexity of 2D shapes [12, 42, 72, 13]. Then, we can perform regression analysis to measure the correlation of each of these variables with respect to consistency and efficiency. While this should give us some idea of how the different factors influence contouring, more experiments are necessary to isolate the effect of each individual variable.

To measure the effect of each individual variable, we need to have users perform contouring tasks in which the values of the independent variable under study are manipulated and all other variables are controlled. For boundary strength, we need a way to manipulate the boundary strength in an image, in order to have users contour on the same images, but with different levels of boundary strength. For the liver, which has strong boundaries to begin with, we can start from the original images and slowly begin adding noise to blur the boundary. Another option is to make use of multimodal imagery. For a structure like the prostate, which is known to have much sharper contrast in MRI images than CT images, we could acquire a set of coregistered CT/MRI images. We can then vary the boundary strength by gradually blending the MRI image with the CT image to get a variety of boundary levels. We can then have users contour these images and measure consistency and efficiency as related to the level of boundary strength.

We can run similar experiments to quantify the effects of contour length and shape complexity. In these cases, we do not even need image data, but can simply have users trace contours. For contour length, we can take a variety of different shapes, grouped according to complexity as a control, and then vary the length of the contour by scaling the contour up or down without changing its shape. For shape complexity, we can group the different shapes according to length as a control, and then study the effect of shape complexity with contours that have the same length. In each case, we are measuring the effect that one independent variable has on contouring consistency and efficiency, with all other variables controlled.

**Creating a model** Once we have quantified the effects of the different factors that influence contouring, we can then use that information to create a model to predict the levels of consistency and efficiency with which we can expect users to mark a given

contour on a given image. We can train our model on a portion of the data collected in our experiments, and we can test it on the portion not used for training. Further testing on other data sets will be required to verify the robustness of the model.

This model can then be incorporated into our plane selection algorithm. The geometrical approach described above will remain the first step in determining which contour to add next. However, the model can be used to eliminate some contours and improve the rankings based solely on geometry. For example, we would not want to include any contour for which we could not ensure a reasonable level of consistency. Also, if a set of contours are expected to provide roughly the same amount of improvement to the surface, then we want to choose the one that is expected to have the highest levels of consistency and efficiency.

**Adaptive protocols** In this section, we have described an algorithm that will allow us to find an effective protocol for contouring a data set in which we know the ground truth, but how can we use that information to create a protocol for a new data set in which we do not know the ground truth? We propose a method that adapts the protocol based on each new contour that the user draws. For an anatomic structure such as the prostate or liver, we can run our plane selection algorithm on many example data sets for which we do have a segmentation. Indeed, our collaborators have been creating such a structure library [1] by capturing patient image data and the segmentations that were used to treat them. This will give us protocols for a variety of different prostate models. Then, when presented with a new prostate data set, we can follow the protocol in our library that most resembles the prostate being segmented as determined by the contours drawn by the user thus far. As the user adds each additional contour, we can adjust the protocol as needed, to choose the next best plane.

This method can also detect and adapt to irregularities. For example, if a contour drawn for a prostate data set does not match anything in our structure library, then we can adapt the protocol to add more planes in that region to better capture this irregular feature. In this manner, we can create an effective protocol for a new data set and adapt it on-the-fly to the input provided by the user.

### 6.1.2 Path to clinical adoption

Ultimately, we would like to see nonparallel contouring adopted in clinical practice and used to reduce morbidity and improve patient care. To this end, we have focused on designing systems that can be used in practice and conducted our evaluations with medical professionals and real patient data. However, more testing is needed to evaluate the effectiveness of nonparallel contouring in the clinic.

While we continue to refine our nonparallel contouring system and to develop methods for choosing effective protocols, there is no reason that we could not go ahead and proceed with limited clinical trials. It has been shown that consistency is increased when users observe their parallel contours in orthogonal views [71]. Therefore, having users mark just one or two nonparallel contours in addition to the standard set of parallel contours may be enough to significantly improve segmentation accuracy.

For example, previous studies have found significant inter-observer variability in defining the clinical target volume of the prostate, particularly in the cranio-caudal direction [21, 11]. We propose to run a similar study, in which experienced radiotherapists will be asked to define the clinical target volume on actual patient data sets using parallel contours. For half the trials, in addition to the standard set of parallel, transverse contours, the radiotherapists will add two orthogonal contours — one in the sagittal plane and one in the coronal plane that each contain the centroid of the volume defined by the parallel contours. The radiotherapists will have the ability to modify their previously drawn parallel contours by adjusting the intersection points of those contours in the orthogonal planes. We hypothesize that we will observe a significant increase in consistency and accuracy in the trials in which the additional orthogonal contours are drawn.

This type of study should be conducted in the clinic and used to treat patients in order to observe patient outcomes. Repeated clinical trials showing significant improvements in outcomes will be necessary for nonparallel contouring to be recommended for adoption as a standard in clinical practice.

### **6.1.3 Error estimation for segmentation review**

This dissertation has focused on improving the accuracy and efficiency of the segmentation process, but an equally important process is that of segmentation review. Regardless of the method used to produce an initial segmentation, the quantitative review of these segmentations is required prior to clinical decisions using this data because the shape, size, and intensity of radiation beams depend directly on the segmentation results. In addition to improving the accuracy and efficiency of the contouring process, we would also like to look at methods of improving the accuracy and efficiency of the review process. One way this could be approached is by incorporating prior knowledge of the structure that is being reconstructed. For most anatomical structures the general shape is known. This knowledge could be used to estimate the error of the reconstructed surface, and then the user could be guided in the review process to regions of the surface containing the highest error-likelihood measures.

### **6.1.4 Extension to other applications and citizen science**

There are many applications, including but certainly not limited to radiation treatment planning, in which segmentation of volume data is required. Often, humans are better at performing the segmentation than even the most advanced automated techniques. However, the time it takes to segment the large amounts of data being captured is great, and therefore much data is captured and never fully analyzed. One approach to this problem is to employ the help of citizen scientists, as in projects such as SETI@Home [4] and the Galaxy Zoo [60]. We are interested in modifying the VolumeViewer interface for use by an untrained audience. This will require strategies that allow a user to visualize and understand typical examples of a structure and to give the user context in segmenting that same structure on novel data. New computational geometry approaches to integrate and validate the segmentation data created by multiple users on the same imagery will also be needed. The successful development of such a system would enable scientists to produce large libraries of segmented volume data that otherwise would have been impossible with their own

limited resources. This data can then be visualized and analyzed in its own right, and also used as training data for improving automated segmentation techniques.

# References

- [1] C. Abraham, D. Low, L. Liu, T. Ju, and J.O. Deasy. Development of a three dimensional anatomical library. *International Journal of Radiation Oncology\*Biology\*Physics*, 72(1, Supplement 1):S655 – S655, 2008. Proceedings of the American Society for Therapeutic Radiology and Oncology 50th Annual Meeting.
- [2] Christopher Abraham, Daniel Low, Lu Liu, Divya Khullar, Aditya Apte, Joseph Deasy, Tao Ju, Parag Parikh, Sasa Mutic, and Wade Thorstad. A process to determine optimal oblique contouring planes. *Medical Physics*, 35:2666, 2008.
- [3] MeVis Medical Solutions AG. Mevislab. <http://www.mevislab.de/>, 2012.
- [4] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public-resource computing. *Commun. ACM*, 45:56–61, November 2002.
- [5] Ulrich Baum, Holger Greess, Michael Lell, Anton Nmayr, and Martin Lenz. Imaging of head and neck tumors methods: Ct, spiral-ct, multislice-spiral-ct. *European journal of radiology*, 33(3):153–160, 2000.
- [6] A. L. Bogush, A. V. Tuzikov, and S. A. Sheynin. 3d object reconstruction from non-parallel cross-sections. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3*, pages 542–545, Washington, DC, USA, 2004. IEEE Computer Society.
- [7] Jean-Daniel Boissonnat and Pooran Memari. Shape reconstruction from unorganized cross-sections. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 89–98, 2007.
- [8] John M. Boone. Multidetector ct: Opportunities, challenges, and concerns associated with scanners with 64 or more detector rows1. *Radiology*, 241(2):334–337, November 2006.
- [9] E. Vital Brazil, I. Macedo, M. Costa Sousa, L. H. de Figueiredo, and L. Velho. Sketching variational hermite-rbf implicits. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium, SBIM '10*, pages 1–8, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.
- [10] Brigham and Women’s Hospital. Slicer. <http://www.slicer.org/>, 2012.



- [11] L F Cazzaniga, M A Marinoni, A Bossi, E Bianchi, E Cagna, D Cosentino, L Scandolaro, M Valli, and M Frigerio. Interphysician variability in defining the planning target volume in the irradiation of prostate and seminal vesicles. *Radiother Oncol*, 47(3):293–6, 1998.
- [12] B. Chazelle and J. Incerpi. Triangulation and shape-complexity. *ACM Trans. Graph.*, 3(2):135–152, April 1984.
- [13] Yinpeng Chen and Hari Sundaram. Estimating complexity of 2d shapes. In *In: Proceedings of the IEEE 7th Workshop on Multimedia Signal Processing*, 2005.
- [14] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17:167–174, 1998.
- [15] Scientific Computing and Imaging Institute (SCI). Seg3d: Volumetric image segmentation and visualization. <http://www.seg3d.org>.
- [16] Robert G. Cromley. Hierarchical methods of line simplification. *Cartography and Geographic Information Science*, 18:125–131(7), April 1991.
- [17] C.R. Dance and R.W. Prager. Delaunay reconstruction from multiaxial planar cross-sections. Technical Report CUED/F-INFENG/TR 273, Department of Engineering, University of Cambridge, Cambridge, England., 1997.
- [18] Jan Van de Steene, Nadine Linthout, Johan de Mey, Vincent Vinh-Hung, Cornelia Claassens, Marc Noppen, Arjan Bel, and Guy Storme. Definition of gross tumor volume in lung cancer: inter-observer variability. *Radiotherapy and oncology*, 62(1):37–49, 2002.
- [19] Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, July 1945.
- [20] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.
- [21] C Fiorino, M Reni, A Bolognesi, G M Cattaneo, and R Calandrino. Intra- and inter-observer variability in contouring prostate and seminal vesicles: implications for conformal treatment planning. *Radiother Oncol*, 47(3):285–92, 1998.
- [22] Franca Foppiano, Claudio Fiorino, Giovanni Frezza, Carlo Greco, and Riccardo Valdagni. The impact of contouring uncertainty on rectal 3D dose-volume data: results of a dummy run in a multicenter trial (AIROPROS01-02). *Int J Radiat Oncol Biol Phys*, 57(2):573–9, 2003.

- [23] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal surface reconstruction from planar contours. *Commun. ACM*, 20(10):693–702, 1977.
- [24] D.T. Gering, A. Nabavi, R. Kikinis, W..E.L. Grimson, N. Hata, P. Everett, F.A. Jolesz, and W.M. Wells III. An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging. *Med Image Comput Comput Assist Interv. MICCAI 1999*, 2:809–819, 09 1999.
- [25] Cindy Grimm. Manifold mesh processing. <http://sourceforge.net/projects/meshprocessing/>, 2010.
- [26] Khronos Group. Opendgl. <http://www.opengl.org/>, 1997.
- [27] Visualization Sciences Group. Open inventor. <http://www.vsg3d.com/open-inventor/sdk/>, 2011.
- [28] Philips Healthcare. Pinnacle. <http://www.healthcare.philips.com/main/products/ros/products/pinnacle3/index.wpd>, 2012.
- [29] R Hermans, M Feron, E Bellon, P Dupont, W Van den Bogaert, and A L Baert. Laryngeal tumor volume measurements determined with CT: a study on intra- and interobserver variability. *Int J Radiat Oncol Biol Phys*, 40(3):553–7, 1998.
- [30] Takanori Higashino, Yoshiharu Ohno, Daisuke Takenaka, Hirokazu Watanabe, Munenobu Nogami, Chiho Ohbayashi, Masahiro Yoshimura, Miyako Satouchi, Yoshihiro Nishimura, Masahiko Fujii, and Kazuro Sugimura. Thin-section multiplanar reformats from multidetector-row ct data: Utility for assessment of regional tumor extent in non-small cell lung cancer. *European journal of radiology*, 56(1):48–55, 2005.
- [31] Yoshio Hirayasu, Martha E. Shenton, Dean F. Salisbury, Chandlee C. Dickey, Iris A. Fischer, Paola Mazzoni, Tanya Kisler, Hajime Arakaki, Jun Soo Kwon, Jane E. Anderson, Deborah Yurgelun-Todd, Mauricio Tohen, and Robert W. McCarley. Lower left temporal lobe mri volumes in patients with first-episode schizophrenia compared with psychotic patients with first-episode affective disorder and normal subjects. *American Journal of Psychiatry*, 155(10):1384–1391, 1998.
- [32] Lennox Hoyte, Wen Ye, Linda Brubaker, Julia R. Fielding, Mark E. Lockhart, Marta E. Heilbrun, Morton B. Brown, Simon K. Warfield, and for the Pelvic Floor Disorders Network. Segmentations of mri images of the female pelvic floor: A study of inter- and intra-reader reliability. *Journal of Magnetic Resonance Imaging*, 33(3):684–691, 2011.
- [33] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, <http://www.itk.org/ItkSoftwareGuide.pdf>, second edition, 2005.

- [34] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [35] Hiroshi Imai and Masao Iri. Computational-geometric methods for polygonal approximations of a curve. *Comput. Vision Graph. Image Process.*, 36:31–41, November 1986.
- [36] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
- [37] Ahmedin Jemal, Freddie Bray, Melissa M. Center, Jacques Ferlay, Elizabeth Ward, and David Forman. Global cancer statistics. *CA: A Cancer Journal for Clinicians*, 61(2):69–90, 2011.
- [38] Olga A. Karpenko and John F. Hughes. Smoothsketch: 3d free-form shapes from complex sketches. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 589–598, New York, NY, USA, 2006. ACM.
- [39] David G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):pp. 87–99, 1989.
- [40] E. Keppel. Approximating complex surfaces by triangulation of contour lines. *IBM Journal of Research and Development*, 10:2–11, 1975.
- [41] Young Hoon Kim, Kyoung Ho Lee, Seong Ho Park, Hyung-Ho Kim, Seokyoung Hahn, Do Joong Park, and Hye Seung Lee. Staging of t3 and t4 gastric carcinoma with multidetector ct: Added value of multiplanar reformations for prediction of adjacent organ invasion1. *Radiology*, 250(3):767–775, March 2009.
- [42] A. F. Koschan, S. R. Sukumar, B. Roui-abidi, and M. A. Abidi. Shape analysis algorithm based on information theory. In *In: Proceedings of the IEEE International Conference on Image Processing*, pages 229–232, 2003.
- [43] E. Levina and P. Bickel. The earth mover’s distance is the mallows distance: some insights from statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 251–256 vol.2, 2001.
- [44] L. Liu, C. Bajaj, J.O. Deasy, D.A. Low, and T. Ju. Surface reconstruction from non-parallel curve networks. *Eurographics (Computer Graphics Forum)*, 27(2):155–164, 2008.
- [45] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

- [46] Ives Macedo, Joo Paulo Gois, and Luiz Velho. Hermite interpolation of implicit surfaces with radial basis functions. In *Proceedings of SIBGRAPI - XX Brazilian Symposium on Computer Graphics and Image Processing*. IEEE Computer Society, 2009.
- [47] C. L. Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, 43(2):pp. 508–515, 1972.
- [48] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, New York, NY, USA, 1995. ACM.
- [49] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [50] Peter Nowak, Erik van Dieren, John van Srsen de Koste, Henry van der Est, Ben Heijmen, and Peter Levendag. Treatment portals for elective radiotherapy of the neck: an inventory in the netherlands. *Radiotherapy and Oncology*, 43(1):81 – 86, 1997.
- [51] National Institutes of Health. Mipav. <http://mipav.cit.nih.gov/>, 2012.
- [52] S.D Olabarriaga and A.W.M Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2):127 – 142, 2001.
- [53] Shigeru Owada, Frank Nielsen, and Takeo Igarashi. Volume catcher. In *I3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 111–116, New York, NY, USA, 2005. ACM.
- [54] Bradley A. Payne and Arthur W. Toga. Surface reconstruction by multiaxial triangulation. *IEEE Comput. Graph. Appl.*, 14(6):28–35, 1994.
- [55] Primoz Petric, Johannes Dimopoulos, Christian Kirisits, Daniel Berger, Robert Hudej, and Richard Potter. Inter- and intraobserver variation in HR-CTV contouring: intercomparison of transverse and paratransverse image orientation in 3D-MRI assisted cervix cancer brachytherapy. *Radiother Oncol*, 89(2):164–71, 2008.
- [56] D. L. Pham, C. Xu, and J. L. Prince. A survey of current methods in medical image segmentation. In *Annual Review of Biomedical Engineering*, volume 2, pages 315–338. Annual Reviews, 2000.
- [57] S. Pieper, M. Halle, and R. Kikinis. 3d slicer. In *IEEE International Symposium on Biomedical Imaging ISBI 2004*, pages 632–635, 04 2004.

- [58] S. Pieper, B. Lorenzen, W. Schroeder, and R. Kikinis. The na-mic kit: Itk, vtk, pipelines, grids and 3d slicer as an open platform for the medical image computing community. In *Proc IEEE Intl Symp on Biomedical Imaging ISBI*, pages 698–701, 04 2006.
- [59] Miranda Poon, Ghassan Hamarneh, and Rafeef Abugharbieh. Efficient interactive 3d livewire segmentation of objects with arbitrarily topologies. *Computerized Medical Imaging and Graphics*, 32(8):639–650, 2008.
- [60] M. Jordan Raddick, Georgia Bracey, Pamela L. Gay, Chris J. Lintott, Phil Murray, Kevin Schawinski, Alexander S. Szalay, and Jan Vandenberg. Galaxy zoo: Exploring the motivations of citizen science volunteers. *Astronomy Education Review*, 9(1):010103, 2010.
- [61] Coen Rasch, Roel Steenbakkens, and Marcel van Herk. Target definition in prostate, head, and neck. *Seminars in Radiation Oncology*, 15(3):136 – 145, 2005.
- [62] S.P. Raya and J.K. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9(1):32–42, 1990.
- [63] Ryan Schmidt and Karan Singh. Sketch-based procedural surface modeling and compositing using Surface Trees. *Computer Graphics Forum*, 27(2):321–330, 2008. Proceedings of Eurographics 2008.
- [64] Will Schroeder, Kenneth M. Martin, and William E. Lorensen. *The visualization toolkit (2nd ed.): an object-oriented approach to 3D graphics*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.
- [65] Sunit Sebastian, Mannudeep K. Kalra, Pardeep Mittal, Sanjay Saini, and William C. Small. Can independent coronal multiplanar reformatted images obtained using state-of-the-art mdct scanners be used for primary interpretation of mdct of the abdomen and pelvis? a feasibility study. *European journal of radiology*, 64(3):439–446, 2007.
- [66] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, New York, NY, USA, 2004. ACM.
- [67] Ross Sowell. Volumeviewer. <http://volumeviewer.cse.wustl.edu/>, 2009.
- [68] Ross Sowell, Lu Liu, Tao Ju, Cindy Grimm, Christopher Abraham, Garima Gokhroo, and Daniel Low. VolumeViewer: an interactive tool for fitting surfaces to volume data. In *SBIM '09: Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 141–148, New York, NY, USA, 2009. ACM.

- [69] Bill Spitzak. Fast light toolkit. <http://www.fltk.org/>, 1998.
- [70] Detlev Stalling, Malte Westerhoff, and Hans christian Hege. Amira: A highly interactive system for visual data analysis. In *The Visualization Handbook*, pages 749–767. Elsevier, 2005.
- [71] Roel J H M Steenbakkens, Joop C Duppen, Isabelle Fitton, Kirsten E I Deurloo, Lambert Zijp, Apollonia L J Uitterhoeve, Patrick T R Rodrigus, Gijbert W P Kramer, Johan Bussink, Katrien De Jaeger, Jose S A Belderbos, Augustinus A M Hart, Peter J C M Nowak, Marcel van Herk, and Coen R N Rasch. Observer variation in target volume delineation of lung cancer related to radiation oncologist-computer interaction: a 'Big Brother' evaluation. *Radiother Oncol*, 77(2):182–90, 2005.
- [72] Godfried Toussaint. Efficient triangulation of simple polygons. *The Visual Computer*, 7:280–295, 1991.
- [73] Fan-Yin Tzeng, Eric B. Lum, and Kwan-Liu Ma. A novel interface for higher-dimensional classification of volume data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 66, Washington, DC, USA, 2003. IEEE Computer Society.
- [74] Stacy J. UyBico, Carol C. Wu, Robert D. Suh, Nanette H. Le, Kathleen Brown, and Mayil S. Krishnam. Lung Cancer Staging Essentials: The New TNM Staging System and Potential Imaging Pitfalls1. *Radiographics*, 30(5):1163–1181, September 2010.
- [75] Marcel van Herk. Errors and margins in radiotherapy. *Seminars in Radiation Oncology*, 14(1):52 – 64, 2004. High-Precision Radiation Therapy of Moving Targets.
- [76] Inc. Varian Medical Systems. Eclipse. [http://www.varian.com/us/oncology/radiation\\_oncology/eclipse/](http://www.varian.com/us/oncology/radiation_oncology/eclipse/), 2012.
- [77] Simon K. Warfield, Kelly H. Zou, and William M. Wells. Simultaneous truth and performance level estimation (staple): An algorithm for the validation of image segmentation. *IEEE TRANS. MED. IMAG*, 23:903–921, 2004.
- [78] Elisabeth Weiss, Susanne Richter, Thomas Krauss, Silke I Metzelthin, Andrea Hille, Olivier Pradier, Birgit Siekmeyer, Hilke Vorwerk, and Clemens F Hess. Conformal radiotherapy planning of cervix carcinoma: differences in the delineation of the clinical target volume. A comparison between gynaecologic and radiation oncologists. *Radiother Oncol*, 67(1):87–95, 2003.
- [79] P.N. Werahera, G.J. Miller, G.D. Taylor, T. Brubaker, F. Daneshgari, and E.d. Crawford. A 3-d reconstruction algorithm for interpolation and extrapolation of

- planar cross sectional data. *IEEE Transactions on Medical Imaging*, 14(4):765–771, 1995.
- [80] Charles M. Williams. An efficient algorithm for the piecewise linear approximation of planar curves. *Computer Graphics and Image Processing*, 8(2):286 – 293, 1978.
- [81] Ivo Wolf, Marcus Vetter, Ingmar Wegner, Thomas Bittger, Marco Nolden, Max Schbinger, Mark Hastenteufel, Tobias Kunert, and Hans-Peter Meinzer. The medical imaging interaction toolkit. *Medical Image Analysis*, 9(6):594 – 604, 2005.
- [82] Xiaoru Yuan, Nan Zhang, Minh X. Nguyen, and Baoquan Chen. Volume cutout. *The Visual Computer (Special Issue of Pacific Graphics 2005)*, 21(8–10):745–754, 2005.
- [83] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

# Vita

Ross Taylor Sowell

- Degrees** Ph.D. Computer Science, August 2012  
Washington University, St. Louis, MO
- B.S. Summa Cum Laude, Computer Science, May 2005  
The University of the South, Sewanee, TN
- Professional Societies** Association for Computing Machinery  
Consortium for Computing Sciences in Colleges
- Publications** Abraham, C., Low, D., Sowell, R., Gokhroo, G., Grimm, C., and Ju, T. VolumeViewer: A tool for examining the use of non-axial image planes in treatment planning. In *Proceedings of the 16th International Conference on the Use of Computers in Radiation Therapy* (Amsterdam, Netherlands, May 31 - June 3, 2010).
- Sowell, R., Gill, C., Chamberlain, R. D., Grimm, C., Goldman, K. J., and Tranel, M. The active-learning transformation: A case study in software development and systems software courses. *Journal of Computing Sciences in Colleges* 25, 5 (May 2010), 165-172.
- Sowell, R., Chen, Y., Buhler, J., Goldman, S. A., Grimm, C., and Goldman, K. J. Experiences with active learning in CS 3. *Journal of Computing Sciences in Colleges* 25, 5 (May 2010), 173-179.



Sowell, R., Liu, L., Ju, T., Grimm, C., Abraham, C., Gokhroo, G., and Low, D. VolumeViewer: An interactive tool for fitting surfaces to volume data. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based interfaces and Modeling* (New Orleans, Louisiana, August 01 - 02, 2009), 141-148.

Lien, J., Bayazit, O., Sowell, R., Rodriguez, S., and Amato, N. Shepherding behaviors. In *Proceedings of the IEEE International Conference on Robotics and Automation* (New Orleans, Louisiana, April 26 - May 4, 2004), 4159-4164.

August 2012

**Segmentation Using Oblique Contours, Sowell, Ph.D. 2012**