

# **Stony Brook University**



OFFICIAL COPY

**The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.**

**© All Rights Reserved by Author.**

# **Improving Cancer Detection Through Visualization**

A Dissertation Presented

by

**Joseph Marino**

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy**

in

**Computer Science**

Stony Brook University

**December 2012**

Copyright by  
Joseph Marino  
2012

**Stony Brook University**

The Graduate School

**Joseph Marino**

We, the dissertation committee for the above candidate for the  
Doctor of Philosophy degree, hereby recommend  
acceptance of this dissertation.

**Arie Kaufman – Dissertation Advisor**  
**Distinguished Professor, Computer Science Department**

**Klaus Mueller – Chairperson of Defense**  
**Professor, Computer Science Department**

**Xianfeng Gu**  
**Associate Professor, Computer Science Department**

**Wei Li**  
**Research Scientist**  
**Siemens Corporate Research**

This dissertation is accepted by the Graduate School

Charles Taber  
Interim Dean of the Graduate School



Abstract of the Dissertation

**Improving Cancer Detection Through Visualization**

by

**Joseph Marino**

**Doctor of Philosophy**

in

**Computer Science**

Stony Brook University

**2012**

The proliferation of medical scanning technology, especially computed tomography (CT) and magnetic resonance imaging (MRI), has led to various research works on methods of intuitively visualizing the large amounts of resulting slice data. The visualization of this data consists not only of display methods, but ideas covering topics such as navigation, exploration, surface parameterization, and analysis. These imaging modalities can play a large role in diagnosing a multitude of diseases, including carcinomas of the colon, lungs, and prostate. The work presented in this dissertation develops varied visualization techniques for application to the detection and localization of these three prevalent cancers. All techniques presented herein were developed and tested using real patient data acquired from CT or MRI scanning modalities.

There has been substantial research performed in the development of clinical virtual colonoscopy (VC) systems based on CT scans, but methods to meld multiple sources of diagnostic information are still under active development. Presented are beginning methods to merge VC with the traditional optical colonoscopy, such that VC information can be better utilized on a patient referred for the optical procedure. This work includes a method for removing the radial distortion introduced by the fisheye lens on the endoscope and a method for correlating the path of the physical colonoscope with the VC path. Techniques are also presented here which can enhance a traditional VC environment. For CT scans of the colon acquired

with the patient in different orientations, a method is presented to register the two scans together in a one-to-one and onto manner. This direct mapping between the two surfaces allows for the corresponding visualization of the same location within each of the two colon models.

Mesh models are often used as a map to show the user's location within an object. In the case of VC, this 3D model can often have occlusions due to the twisted shape of the colon. A method is presented to create a planar map which preserves the global shape of the colon and does not contain any overlapping sections. This allows for the user to observe the entire colon surface at once, ensuring that there are no occlusions which could lead to an ambiguity in determining one's location. Similar problems exist for exploring the lungs in virtual bronchoscopy applications, and can be even worse, due to the large number of bifurcations. The idea of a context preserving map is expanded from tubular structures to treelike structures, being applicable not only to the bronchi, but other branching structures such as blood vessels. As there are different aspects to be addressed compared to a simple tubular structure, this work is not simply an extension, but presents new methods which have been developed to deal with the challenges inherent in treelike structures.

For the prostate, multiple MRI modes are typically used for the detection of cancer. Compared to colon and lung cancer, the prostate has so far seen relatively little in terms of visualization research, and work is presented here on how to combine the multiple MR modes to assist in the detection of prostate cancer. This includes an upsampling and analysis of the slices to identify regions of interest and the display of these regions within the prostate along with the surrounding anatomy using multi-volume rendering. Also developed is a method of visibility persistence to allow for easy viewing of occluded regions of interest, the ability of the user to paint custom regions into the data, and an extension of the visibility persistence to these user painted regions.

# Contents

<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Algorithms</b>	<b>xii</b>
<b>Publications</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Volumetric Visualization . . . . .	2
1.2 Surface Rendering and Parameterization . . . . .	5
1.3 Medical Imaging for Cancer Diagnosis . . . . .	8
1.3.1 Virtual Endoscopy . . . . .	8
1.3.2 Prostate Imaging . . . . .	11
1.4 Contributions . . . . .	12
<b>2 Virtually Assisted Optical Colonoscopy</b>	<b>15</b>
2.1 Undistortion of Fisheye View . . . . .	15
2.1.1 Motivation . . . . .	15
2.1.2 Method . . . . .	16
2.1.3 Results . . . . .	17
2.2 Path Correspondence . . . . .	18
2.2.1 Motivation . . . . .	18
2.2.2 Method . . . . .	20
2.2.3 Results . . . . .	22
<b>3 Supine and Prone Colon Registration</b>	<b>26</b>
3.1 Motivation . . . . .	26

3.2	Anatomical Landmarks . . . . .	29
3.2.1	Taeniae Coli . . . . .	29
3.2.2	Flexures . . . . .	30
3.3	Surface Features . . . . .	31
3.4	Quasi-Conformal Mapping . . . . .	35
3.5	Results and Visualization . . . . .	37
3.5.1	Quantitative Results . . . . .	37
3.5.2	Corresponding Visualization . . . . .	39
3.5.3	Visualization of Quasi-Conformality . . . . .	42
<b>4</b>	<b>Contextual Maps of Colon Surfaces</b>	<b>46</b>
4.1	Motivation . . . . .	46
4.2	Skeleton Layout . . . . .	48
4.2.1	Orthogonal Projection . . . . .	48
4.2.2	Lengthening . . . . .	50
4.2.3	Untangling . . . . .	51
4.2.4	Spacing . . . . .	53
4.3	Boundary Placement . . . . .	54
4.3.1	Slicing . . . . .	55
4.3.2	Skeletal Correspondence . . . . .	55
4.3.3	Radius Along Skeleton . . . . .	57
4.3.4	Boundary Position . . . . .	58
4.4	Surface Mapping . . . . .	58
4.4.1	Rectangular Mapping . . . . .	59
4.4.2	Deformation . . . . .	61
4.4.3	Rendering . . . . .	61
4.5	Results . . . . .	62
<b>5</b>	<b>Contextual Maps of Treelike Structures</b>	<b>65</b>
5.1	Motivation . . . . .	65
5.2	Algorithm . . . . .	69
5.2.1	Orthographic Projection . . . . .	70
5.2.2	Radial Layout . . . . .	71
5.2.3	Shape Recovery . . . . .	72
5.2.4	Boundary Creation . . . . .	74

5.2.5	Surface Mapping . . . . .	76
5.3	Visualization . . . . .	78
5.3.1	Map Surface View . . . . .	78
5.3.2	Depth Cues . . . . .	79
5.3.3	Direct Inspection . . . . .	80
5.4	Results . . . . .	80
5.5	Evaluation . . . . .	82
<b>6</b>	<b>Prostate Visualization</b>	<b>87</b>
6.1	Motivation . . . . .	87
6.2	Score Volumes . . . . .	89
6.2.1	Standard Patient Dataset . . . . .	89
6.2.2	Label Map Upsampling . . . . .	91
6.2.3	Composite Label Map . . . . .	93
6.2.4	Score Volume Creation . . . . .	94
6.2.5	2D Slice-Based Visualization . . . . .	96
6.3	3D Visualization . . . . .	96
6.3.1	Standard Prostate Region . . . . .	97
6.3.2	Surrounding Anatomy . . . . .	99
6.4	Exploration and Interaction . . . . .	100
6.4.1	Visibility Persistence . . . . .	100
6.4.2	Interactive Region Painting . . . . .	102
6.4.3	Visibility Persistence in Painted Regions . . . . .	103
6.5	Results . . . . .	104
6.5.1	Performance . . . . .	104
6.5.2	Score Evaluation . . . . .	105
6.5.3	Physician Evaluation . . . . .	107
<b>7</b>	<b>Conclusions and Future Work</b>	<b>109</b>
7.1	Conclusions . . . . .	109
7.2	Future Work . . . . .	110
7.2.1	Short Term . . . . .	110
7.2.2	Long Term . . . . .	111

## List of Figures

1.1	Basic volume rendering . . . . .	4
1.2	Direct volume rendering with opacity peeling . . . . .	4
1.3	Rendering normals on triangular mesh . . . . .	5
1.4	Conformal mapping for a colon surface . . . . .	6
1.5	Optical and virtual colonoscopy views . . . . .	9
1.6	Typical prostate MRI slices . . . . .	13
2.1	Geometric pattern undistortion . . . . .	19
2.2	Colon undistortion . . . . .	20
2.3	Centerline and shortest paths . . . . .	21
2.4	Colonoscopy diagram . . . . .	21
2.5	Path correlation method . . . . .	22
2.6	Extracted centerline and shortest path . . . . .	23
2.7	Example correlations between VC and OC paths . . . . .	23
2.8	Simulated correspondence of VC and OC path views . . . . .	25
3.1	Supine and prone colon models . . . . .	27
3.2	Supine-prone registration pipeline . . . . .	28
3.3	Taenia coli . . . . .	29
3.4	Four colon flexures . . . . .	31
3.5	Conformal modulus comparison . . . . .	32
3.6	Color encoding mean curvature . . . . .	33
3.7	Feature detection and matching . . . . .	34
3.8	Geometric meaning for Beltrami coefficients . . . . .	36
3.9	Visual verification for colon registration . . . . .	38
3.10	Visualization of registration results . . . . .	41
3.11	Consistent endoluminal views . . . . .	42

3.12	Visualization of quasi-conformality . . . . .	43
3.13	Visualization of Beltrami coefficients . . . . .	44
4.1	Common clinical diagram of colon . . . . .	47
4.2	Context preserving map of a colon . . . . .	49
4.3	Colon surface viewed from above . . . . .	50
4.4	Colon skeleton processing . . . . .	51
4.5	Types of intersections . . . . .	52
4.6	Refined 2D skeleton . . . . .	54
4.7	Slicing path . . . . .	56
4.8	Correspondence between boundary and skeleton points . . . . .	57
4.9	Boundary placement in 2D . . . . .	59
4.10	Colon surface mapped to rectangle . . . . .	60
4.11	Example conformal modules . . . . .	60
4.12	Colon mapping results . . . . .	62
4.13	Straight map of colon . . . . .	63
4.14	Failure examples . . . . .	64
5.1	Planar visualization for bronchial tree . . . . .	66
5.2	Mesh of cranial blood vessels . . . . .	70
5.3	Embedding of the 2D projected skeleton . . . . .	74
5.4	Importance of consistent mesh slicing . . . . .	76
5.5	Boundary adjustment . . . . .	76
5.6	Flattened surface of cranial blood vessels . . . . .	78
5.7	Map of cranial vessels . . . . .	79
5.8	Visualization for direct inspection . . . . .	80
5.9	Map of lower limb blood vessels . . . . .	84
5.10	Modifying placement weights . . . . .	85
5.11	Texture mapping . . . . .	85
5.12	Context preserving map results . . . . .	86
6.1	Visualization of prostate and surrounding anatomy . . . . .	88
6.2	Natural alignment of prostate MR slices . . . . .	90
6.3	Ternary labelmap interpolation . . . . .	93
6.4	Results of composite segmentation upsampling . . . . .	94
6.5	Score volume in slice-based viewing . . . . .	96

6.6	Rendering examples of individual score values . . . . .	97
6.7	Seminal vesicles . . . . .	98
6.8	Rendering examples of score values . . . . .	98
6.9	Angle between prostate and rectum . . . . .	99
6.10	Rendering example with visibility persistence . . . . .	101
6.11	Screenshot of prostate user interface . . . . .	102
6.12	Persistence visibility for custom painted regions . . . . .	104
6.13	Rendering performance for prostate visualization . . . . .	106
7.1	Flattened volumetric colon wall . . . . .	111
7.2	Electronic biopsy of supine and prone polyp . . . . .	112



## List of Tables

3.1	Statistics for conformal mapping . . . . .	39
3.2	Computational time for quasi-conformal registration . . . . .	39
3.3	Distance errors comparison . . . . .	40
5.1	Running times for treelike maps . . . . .	81
5.2	Average angular distortion . . . . .	82
5.3	User study results . . . . .	83
6.1	Typical prostate dataset properties . . . . .	105
6.2	Sextant evaluation for prostate cancer . . . . .	107
6.3	Seminal vesicle evaluation for cancer . . . . .	107
6.4	Sextant evaluation for hemorrhage . . . . .	108

## List of Algorithms

5.1	Calculation of placement weights . . . . .	72
5.2	Radial layout using placement weights . . . . .	73
5.3	Shape recovery for skeleton . . . . .	75

## Publications

- J. Marino and A. Kaufman. Planar visualization of treelike objects. *IEEE Trans. on Visualization and Computer Graphics*, Submitted for publication, 2012
- J. Marino and A. Kaufman. Prostate visualization with overlapping regions of interest. *IEEE Pacific Visualization*, Poster, Feb. 2012
- J. Marino, W. Zeng, X. Gu, and A. Kaufman. Context preserving maps of tubular structures. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1997–2004, Dec. 2011
- J. Marino and A. Kaufman. Prostate cancer visualization from MR imagery and MR spectroscopy. *Computer Graphics Forum*, 30(3):1051–1060, June 2011
- W. Zeng, J. Marino, A. Kaufman, and X. Gu. Volumetric colon wall flattening using harmonic differentials. *Computers & Graphics*, 35(3):726–732, June 2011
- R. Jiang, J. Marino, W. Zeng, X. Zhao, X. Gu, and A. Kaufman. Registration of volumetric prostate scans using curvature flow. *Technical Report, Computer Science Department*, March 2011
- J. Marino, W. Du, M. Barish, E. Li, W. Zhu, and A. Kaufman. Evaluation of electronic biopsy for clinical diagnosis in virtual colonoscopy. *Proc. of SPIE Medical Imaging*, 7964:796419, Feb. 2011
- J. Marino and A. Kaufman. Upsampling of prostate volumes. *Proc. of CEWIT International Conference*, Sept. 2010
- J. Marino and A. Kaufman. Colon visualization using shape preserving flattening. *Proc. of MICCAI Workshop on Virtual Colonoscopy & Abdominal Imaging*, pages 155–159, Sept. 2010

- W. Zeng, J. Marino, X. Gu, and A. Kaufman. Conformal geometry based supine and prone colon registration. *Proc. of MICCAI Workshop on Virtual Colonoscopy & Abdominal Imaging*, pages 149–154, Sept. 2010
- W. Zeng, J. Marino, K. C. Gurijala, X. Gu, and A. Kaufman. Supine and prone colon registration using quasi-conformal mapping. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1348–1357, Nov. 2010
- J. Marino, F. Qiu, and A. Kaufman. Co-registration of virtual and optical colonoscopy views. *Proc. of MICCAI Workshop on Virtual Colonoscopy*, pages 122–126, Sept. 2008
- F. Qiu, J. Marino, , and A. Kaufman. Computer aided polyp detection with texture analysis. *Proc. of MICCAI Workshop on Virtual Colonoscopy*, pages 148–152, Sept. 2008
- J. Marino, F. Qiu, and A. Kaufman. Virtually assisted optical colonoscopy. *Proc. of SPIE Medical Imaging*, 6916:69160J, Feb. 2008
- W. Hong, F. Qiu, J. Marino, and A. Kaufman. Computer-aided detection of colonic polyps using volume rendering. *Proc. of SPIE Medical Imaging*, 6514:651406, Feb. 2007

# Chapter 1

## Introduction

Over the past decades, the use of medical imaging techniques such as magnetic resonance imaging (MRI) and computed tomography (CT) have steadily increased. As improvements to the scanning technology have moved forward, visualization techniques have been developed to help deal with the massive amounts of data which are acquired [143]. Volume rendering has become a standard technique for visualizing medical data, and volumetric ray casting, which can be accelerated on the GPU, is the preferred method [129]. For cancer in solid organs, such techniques have been applied for detection of tumors within the breast or brain [20] and for treatment of liver tumors [122]. Rendering of whole body datasets for virtual autopsy applications has also been presented [79]. As new interfaces are developed to allow user interaction, these techniques are being expanded for new applications in surgery planning using multi-touch tables [83].

Carcinomas form one of the leading causes of death among people worldwide. Specifically, colon, lung, and prostate cancer are three of the most common types of cancer [105]. In the United States, lung cancer and colorectal cancer are the first and second leading causes of cancer-related mortality [52]. Prostate cancer is the most incident cancer across all demographics, even when including both men and women, and is the second leading cause of cancer-related mortality among men only. The work presented here focuses on these three very prevalent cancers and on developing new visualization methods to assist in exploring and locating said cancer.

Given this medical data, visualization techniques are used to present the data to the user in a useful manner and in accordance with the requirements for diagnosis. In this chapter, a brief introduction to these topics, both technical and medical, is presented. Volume rendering is used to create images from the medical data. Surface mapping techniques based on conformal geometry are used to map 3D surfaces to a 2D plane. Background information on the diagnosis of colon, lung, and prostate cancers is also discussed, as well as the current use of visualization

and computer graphics in those fields.

## 1.1 Volumetric Visualization

Stacks of medical image data can be thought of as represented on a regular Cartesian grid of elements called voxels, which form a volume in 3D space. To display this data, volume rendering methods have been developed [24, 71, 77]. In a traditional computer graphics rendering pipeline, pre-defined surfaces are rendered [30]. Volumes can be rendered in a similar manner by first extracting an isosurface [82]. It is also possible to use continuous representations, such as non-uniform rational B-spline (NURBS) surfaces [112]. On the other hand, direct volume rendering (DVR) allows for the volume itself to be directly rendered without the need to extract surfaces which would be typically rendered in a computer graphics pipeline. Maximum intensity projection (MIP) rendering of volumes can also be accomplished, which yields a 2D image that appears similar to an x-ray [100]. However, this technique is not applicable for a great many applications due to its visual limitations.

In order to perform DVR, an optical model must be established, and there are several possibilities [98]. The most common model is the emission-absorption model, in which the elements of the volume are considered as particles within a cloud which are able to both emit their own light and absorb incident light. This leads to the following so-called volume rendering integral:

$$I(D) = I_0T(D) + \int_0^D g(s)T'(s)ds. \quad (1.1)$$

In this equation, the result  $I(D)$  represents the radiance reaching the camera from  $D$ . The first term represents the background illumination ( $I_0$ ) multiplied by the transparency of the cloud ( $T(D)$ ). The second term represents the integration over all sample positions  $s$ , multiplying each sample's source value ( $g(s)$ ) with the transparency between  $s$  and the eye ( $T'(s)$ ).

This volume rendering integral in the continuous domain can be discretized into compositing for use with discretely sampled data. In this case, composition is possible using both back-to-front and front-to-back methods. For back-to-front compositing, the following is performed at each sample step:

$$C_{dst} \leftarrow (1 - \alpha_{src})C_{dst} + C_{src}. \quad (1.2)$$

For front-to-back compositing, the following is performed at each sample step:

$$\begin{aligned} C_{dst} &\leftarrow C_{dst} + (1 - \alpha_{dst})C_{src}, \\ \alpha_{dst} &\leftarrow \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src}. \end{aligned} \quad (1.3)$$

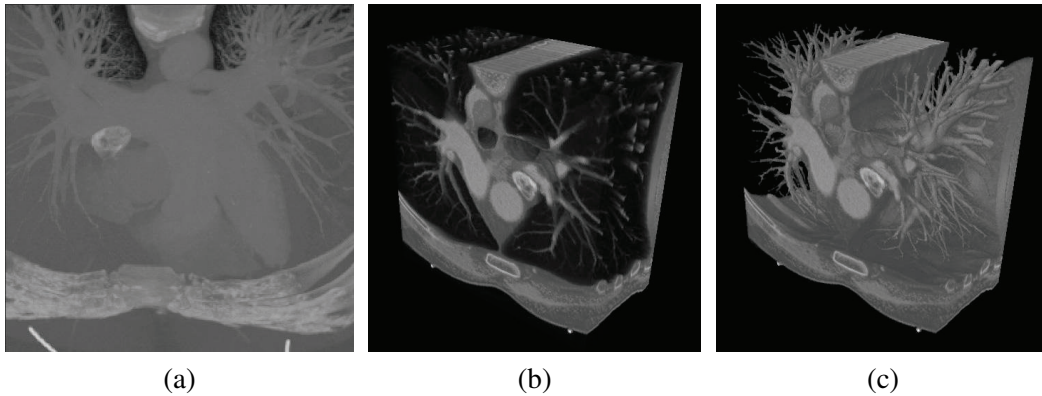
In both instances,  $C_{dst}$  and  $C_{src}$  are the destination and source colors, while  $\alpha_{dst}$  and  $\alpha_{src}$  are the destination and source opacities. Note that while front-to-back compositing requires extra maintenance of the opacity term through the integration, it also allows for early termination of the composition when the opacity reaches a sufficient level (often  $\alpha_{dst} = 0.95$ ).

There are a number of methods for performing the actual volume rendering. A technique has been introduced where the voxels are *splatted* onto the screen space and rendered as disks [151]. A method known as *shear warp* was also proposed where the viewing transformation would be factored into a 3D shear parallel to the volume slices, a projection would create a distorted image, and a 2D warp would then be used to undistort the final image [67]. The volume as a whole can also be decomposed into individual slices which are then rendered and composited with typical 2D texture rendering in the graphics pipeline. The method known as ray casting involves shooting rays through the volume data and sampling at regular points along the rays [72]. Ray casting typically provides the best image quality and is the preferred method of performing DVR [129].

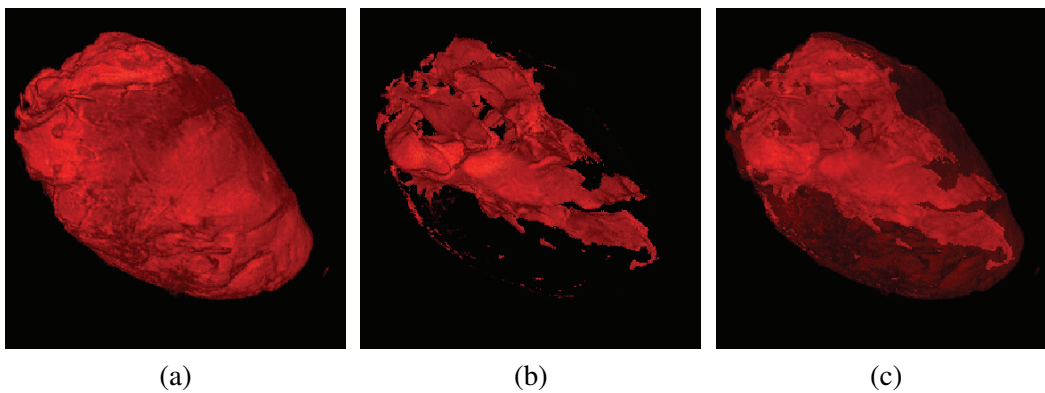
When sampling through the volume, interpolation is needed to obtain sample values not lying at an exact voxel. Nearest neighbor interpolation is the quickest and easiest method, though it leads to undesirable quality for the rendered image. Trilinear interpolation provides significantly better results for moderately more work, and is the most commonly used interpolation method for volume rendering. More advanced methods have also been used to achieve better results, such as tricubic interpolation or the use of Voronoi splines [99], though the tradeoff between speed and quality is not advantageous in typical circumstances.

When rendering volumes, which often consist of singular scalar density values for each voxel, it is desirable to map these scalar values to optical properties, such as color and opacity. A 1D transfer function is used to provide such a mapping [110]. Figure 1.1 shows the difference between general MIP rendering, ray casting with no transfer function, and ray casting with opacity values adjusted in the transfer function. Multi-dimensional transfer functions can also be used to give more control over the mapping. Most commonly, a 2D transfer function is utilized with density and gradient magnitude as the two axes. The rendering of isosurfaces within volumes can be improved by utilizing peak finding [65].

To convey greater realism and recognition of the objects being rendered, illumination and shading are used. Based on the gradient information at each sampling point, shading calculations can be performed using a typical illumination model, such as Blinn shading. Gradients are often estimated as a simple central difference in each of the three axes. These gradients can be calculated on-the-fly or pre-computed, based on the needs of the renderer. When more precision is required, better methods of gradient estimation have been proposed [51].



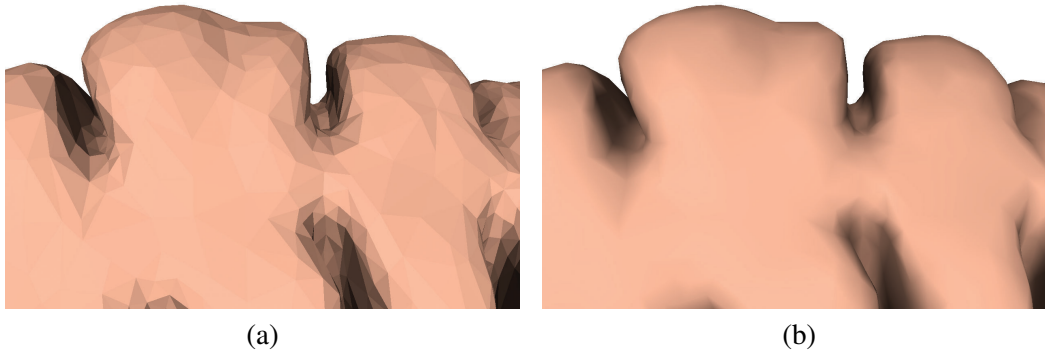
**Figure 1.1:** Basic volume rendering. (a) Straight-on maximum intensity projection (MIP). (b) Perspective density composition. (c) Perspective density composition with transfer function to further attenuate lower opacity values.



**Figure 1.2:** Direct volume rendering (DVR) with opacity peeling. (a) Outer layer. (b) Inner layer. (c) Blending result of outer and inner layers.

There have been a number of improvements to increase the quality of rendering, both in visual quality and speed. Stochastic jittering can be used to slightly offset the positions of rays so that wood-grain effects are reduced [25]. Pre-integrated transfer functions have been introduced to account for values between two discretely sampled values [26]. Empty space skipping has been introduced to allow a ray caster to not sample from regions which do not contain displayable voxels [74]. Direct interval volume visualization has also been presented as a way of representing sharp isosurfaces so that they are visualized accurately [2]. Different ways of presenting the information can also be used, such as using opacity peeling to show inner structures that are otherwise occluded by an outer layer with similar density values [120]. Figure 1.2 shows an example of DVR with shading and





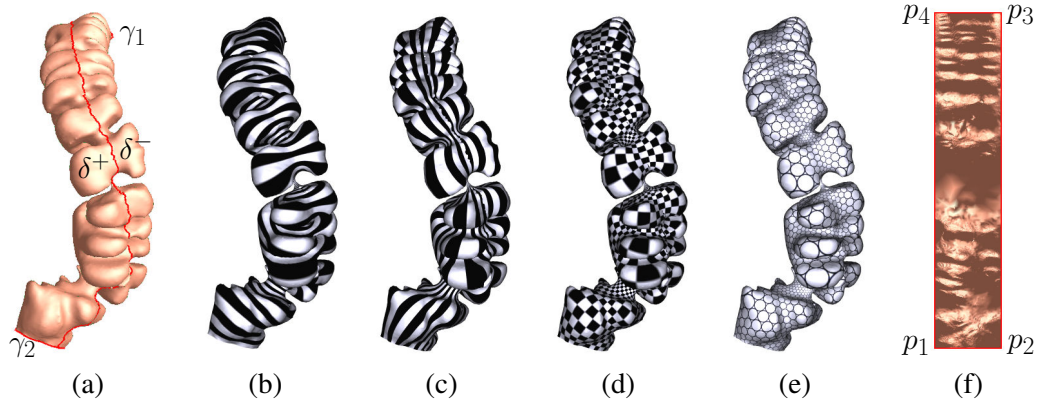
**Figure 1.3:** Effect of normal calculation on the rendering of triangular meshes. (a) Shading with per-face normals. (b) Shading with per-vertex normals.

opacity peeling.

Although specialized hardware for volume rendering was developed [108, 109], it has become common today to utilize commodity graphics hardware for this task [25]. The graphics processing unit (GPU) was originally developed for gaming applications, but provides a high level of SIMD performance at a relatively low cost and can allow for real-time ray casting [131]. More recently, general processing on the GPU has become popular, with Nvidia’s C-like CUDA language [21] and the somewhat similar open standard OpenCL [104] both becoming widely adopted.

## 1.2 Surface Rendering and Parameterization

In computer graphics, surfaces are often represented as discrete meshes, with triangular meshes being a popular choice due to the guarantee that the vertices of each mesh will lie within a plane. There are several ways in which this information can be represented in memory such that connected relationships between vertices, faces, edges, and the neighbors of such items are easily accessed. Some popular methods include the winged-edge, half-edge, and quad-edge data structures [61]. From the connectivity information of the vertices and faces, other information can be easily calculated, such as normals. Normals can be calculated on a per-face or per-vertex basis. For per-face normals, the normal is simply the direction perpendicular to the plane in which the face lies, with the direction based on the order of the vertices (clockwise or counterclockwise). From these flat normals, per-vertex normals can be calculated as the weighted average of the per-face normals around a vertex (typically weighted by the face’s area). By using these smoother normals, a discrete triangular mesh can take on a more organic and less faceted appearance when rendering with shading (see Figure 1.3).



**Figure 1.4:** Conformal mapping for a colon surface. (a) The colon surface, which is a topological cylinder, reconstructed from a CT scan. (b) The level set of the harmonic 1-form  $\omega$ . (c) The level set of the conjugate harmonic 1-form  $*\omega$ . (d) Checker-board texture mapping for the resulting holomorphic 1-form  $\tau$ , where the right angles are preserved. (e) Circle-packing texture mapping, where the shapes of the circles are preserved. (f) The rectangular fundamental domain  $\tilde{D}$ .

There are several methods of mapping 3D surfaces to a 2D plane, but conformal mapping is very applicable for visualization needs because it is angle, and thus shape, preserving [35]. The use of conformal geometry for mapping triangular meshes, where local angles are preserved, has been well established in the field of computer graphics, especially in the creation of texture maps [130]. Texture mapping allows for a 3D image to be mapped to the 2D surface. This is accomplished through the parameterization of the 3D vertices to a 2D domain.

Given a triangular mesh  $Q$  of genus zero with two boundaries,  $\gamma_1$  and  $\gamma_2$ , the boundary of  $Q$  is given as  $\partial Q = \gamma_1 - \gamma_2$ , where  $\gamma_1$  denotes the outer boundary and  $\gamma_2$  the inner boundary.  $\delta$  is the slicing line between  $\gamma_1$  and  $\gamma_2$ , which cuts  $Q$  open to  $\tilde{Q}$ , and  $\delta^+$ ,  $\delta^-$  are the two splits of  $\delta$ . In order to find a conformal mapping  $\phi : Q \rightarrow \mathbb{C}$  which maps  $Q$  to a planar rectangle, two harmonic functions  $f_1, f_2 : Q \rightarrow \mathbb{R}$  are computed such that:

$$\begin{cases} \Delta f_1 = 0 \\ f_1|_{\gamma_1} = 1 \\ f_1|_{\gamma_2} = 0 \end{cases}, \begin{cases} \Delta f_2 = 0 \\ f_2|_{\delta^+} = 0 \\ f_2|_{\delta^-} = 1 \\ \frac{\partial f_2}{\partial \mathbf{n}}|_{\gamma_1 \cup \gamma_2} = 0 \end{cases}.$$

Here,  $f_1$  is computed on  $Q$ , and  $f_2$  on  $\tilde{Q}$ . Figures 1.4 (b) and (c) show the level sets of the harmonic functions  $f_1$  and  $f_2$ , respectively, on a sample colon segment.

The cotan formula [34] is used to approximate the Laplace-Beltrami operator. For each edge  $[v_i, v_j]$ , whose the two adjacent triangles are  $[v_i, v_j, v_k]$  and  $[v_j, v_i, v_l]$ ,

the angles against the edge are  $\theta_{ij}^k$  and  $\theta_{ji}^l$  in each triangle respectively, then the cotangent weight is defined as:

$$w_{ij} = \cot \theta_{ij}^k + \cot \theta_{ji}^l.$$

The Laplacian of a function  $f : V \rightarrow \mathbb{R}$  is then defined as:

$$\Delta f(v_i) = \sum_{[v_i, v_j] \in E} w_{ij} (f(v_i) - f(v_j)).$$

Then,  $\nabla f_1$  is a closed harmonic 1-form, and  $\nabla f_2$  is an exact harmonic 1-form. It is necessary to find a scalar  $\lambda$  such that  ${}^* \nabla f_1 = \lambda \nabla f_2$ . This can be achieved by minimizing the following energy:

$$E(\lambda) = \sum_{[v_i, v_j, v_k] \in F} |\nabla f_1 - \lambda \mathbf{n} \times \nabla f_2|^2 A_{ijk},$$

where  $\nabla f_1$  and  $\nabla f_2$  are the constant gradient vectors of  $f_1$  and  $f_2$  on the face  $[v_i, v_j, v_k]$ ,  $\mathbf{n}$  is the normal vector to the face, and  $A_{ijk}$  is the area of the face. By solving a linear equation,  $\lambda$  can be obtained.

The desired holomorphic 1-form is  $\omega = \nabla f_1 + \sqrt{-1} \lambda \nabla f_2$ . The induced conformal mapping  $\phi : Q \rightarrow \mathbb{C}$  is given by:

$$\phi(p) = \int_q^p \omega,$$

where  $q$  is the base point and the path from  $q$  to  $p$  is arbitrarily chosen. Then, the surface is conformally mapped to a planar domain. By tracing the straight line perpendicular to the two boundaries  $\gamma_1$  and  $\gamma_2$ , a rectangular fundamental domain is obtained, as shown in Figure 1.4(f). The checker-board (d) and circle-packing (e) texture mappings demonstrate the conformality of the mapping, where the right angles of the squares and the shape of the circles are preserved. This conformal mapping based on holomorphic differentials has linear computational complexity, and the conjugate gradient method is used to solve the linear system.

Discrete Ricci flow is a more recent method of computing conformal maps of structures [59]. This process acts similar to heat diffusion, with the deformation of the edge lengths being driven by the discrete Gaussian curvature. A circle packing metric is used to approximate the discrete conformal deformation. The final mapping result can be affected by setting the target Gaussian curvature for each vertex in the mesh.

There are many applications where discrete Ricci flow is useful, such as in surface parameterization for texture mapping and construction of geometric structures

[58] as well as to obtain optimal surface parameterizations using inverse curvature maps [158]. Applications of Ricci flow will typically use a target Gaussian curvature of zero everywhere, or constrain the boundary to a canonical shape, such as a disc. Discrete Ricci flow has also been used for 3D shape analysis [164]. Based on circle packing, the Ricci flow method has typically required the triangular meshes to contain triangles that are not obtuse. The discrete Ricci flow method has been generalized using inversive distance circle packing to remove this requirement for the triangulation to lack skinny triangles [157].

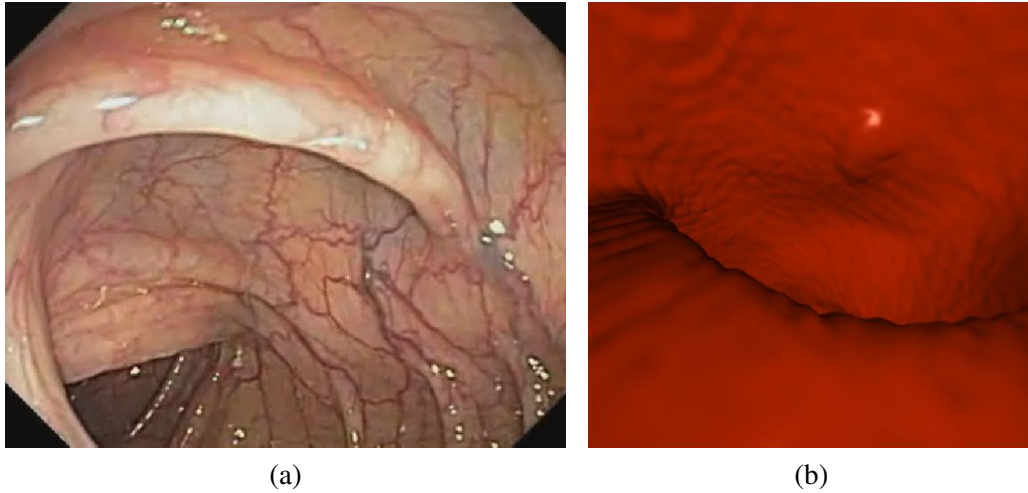
## 1.3 Medical Imaging for Cancer Diagnosis

### 1.3.1 Virtual Endoscopy

As with all carcinomas, prevention is the best way of reducing mortality from colorectal cancer. For this, it is recommended for people over fifty to be regularly screened for colon polyps, the precursors of cancer [141]. For this screening, optical colonoscopy (OC) has been the traditional method and still remains as the gold standard. In OC, a colonoscope is inserted through the rectum and snaked through the colon until the end at the cecum. At this point, the gastroenterologist will proceed to slowly pull the scope out, inspecting the colon walls along the way to identify any polyps. If a polyp is found, it can be removed and biopsied to determine whether it was benign (hyperplastic) or precancerous (adenomatous).

Regular OC screenings have been shown to successfully reduce the mortality rate for colon cancer, are well established, and allow for suspicious polyps to be immediately removed. However, OC does contain several disadvantages. The first disadvantage starts the night before, when the patient is required to render the colon clear of all materials so that the scope may be inserted. The procedure itself is invasive, requiring anesthetization, and comes with a risk, albeit small, of complications [31]. A third disadvantage is that not all of the colon surface can be observed due to the singular direction of the scope's view and the geometry of the colon; approximately one quarter of the surface is missed in a typical OC examination [50].

To combat the disadvantages of traditional OC screening with a more patient friendly method, virtual colonoscopy (VC) based on CT scans has been developed [46]. For VC, volume rendering is used to generate high quality images of the colon surface. The 3D volume rendered view is referred to as the endoluminal view. Figure 1.5 shows a comparison between an OC view and a VC view. The surface of the colon in the VC rendering is purposely left untextured and with simple lighting. Though methods have been proposed for texturing endoluminal views and adding distance-based lighting effects for nasal passages [66], these effects are deemed invasive in the VC and could distract the user from identifying actual polyps and



**Figure 1.5:** Typical views for (a) an optical colonoscopy and (b) a virtual colonoscopy.

regions of interest.

Electronic cleansing of the colon allows for an easier prep for the patient on the day before the examination. Material remaining in the colon which has been tagged with barium can be digitally removed. A method based on casting so-called segmentation rays through the data was proposed to identify locations of intersection between two distinct intensity regions [68]. More recently, statistical based image processing methods utilizing expectation-maximization have been presented [147, 149]. Given a cleansed colon, a region growing segmentation can be performed to extract the interior of the colon, allowing for the colon wall to then be identified. Research has also been performed on segmenting the outer colon wall boundary using level sets [139].

An important task is the extraction of an accurate centerline, or medial axis, through the colon. Much work has been done over the years on methods of extracting these skeleton curves, both in general and specifically related to the colon. Skeleton extraction from volumetric medical data by connecting centerlines based on a medial voxel path has been presented [168]. A method has been proposed using a distance from boundary field with penalty weights to obtain general skeletons of tree-like structures [10]. Obtaining a skeleton directly from a mesh by contracting the surface, without any need for a volumetric representation, has been suggested [63]. Both surface and curve skeletons from genus 0 structures have been extracted through the use of advection to move mass from the object's surface to its skeleton [119]. The use of least squares optimization has been suggested for extracting curve skeletons in an efficient manner which is insensitive to noise [148]. Curve skeletons can also be approximated for data from incomplete point clouds based on

a generalized rotational symmetry axis for a set of points [135].

A mesh of the colon surface is typically extracted to use for navigation and to give the user an external view of the colon. Several methods have been proposed for extracting triangular meshes from volumes, the most popular method being the well-known marching cubes technique [82]. In order to improve the quality of triangular meshes extracted from medical imagery, a technique using surface nets was developed [32]. A method based on dual contouring, which seeks to preserve the original topology of the object being extracted, has also been presented [165].

Within the VC interface, several tools are presented to the user to aid in interrogating suspicious regions. For example, the user is able to measure the size of the polyp in 3D and adjust the lighting to better observe shapes. A more interesting tool is a so-called electronic biopsy [145]. This electronic biopsy allows the user to gain insight into the internal structure of the colon wall.

Virtually unfolding the colon surface to a plane has been presented [7, 78]. Conformal mapping has also been used to map the colon surface to a rectangular plane [47]. These methods allow the user to view the entire inner colon surface at once as a long strip in a 2D plane. A method of volumetric curved planar reformation has also been proposed to allow the user to view the 3D colon surface set into a 2D CT slice [153].

There has been significant work in the computer-aided detection (CAD) of colonic polyps. Conformal mapping has been used as the basis for CAD based on texture analysis of colonic polyps [48]. Other methods have also been used for colon CAD, most commonly based on shape analysis methods [134, 159]. The use of rendering curvature lines on the inner colon surface has also been suggested for detecting polyps [167].

Surgery simulation is a common visualization task, and techniques have been developed for endoscopic tasks. Specific to the colon, there has been work on generating images to simulate a general colonoscopy procedure [38]. For more advanced colon cancer surgery, newer simulation systems involving collision detection and reaction have also been developed [117]. A system for endoscopic pituitary procedures has also been developed to assist in training and for surgery planning [103].

Similar to virtual colonoscopy, virtual bronchoscopy has been developed for noninvasive screening in the identification of lung nodules [23]. Similar techniques are used, including the extraction of the medial axis for navigation and a rendering of an endoluminal view [45]. Path planning can then be performed to assist the user in navigation [64]. However, navigational aids are still needed, such as global maps, trail markers, cross-references to two-dimensional views, and so on [133], and further research into more advanced techniques is still being undertaken [107].

### 1.3.2 Prostate Imaging

Common methods used to identify the possibility of prostate cancer include a blood test to check the level of prostate specific antigen (PSA) and the digital rectal examination. However, these tests are non-specific; for example, an elevated PSA could indicate prostate cancer, but it could also indicate another problem with the prostate, such as benign prostatic hyperplasia (BPH). When there is elevated suspicion for the presence of prostate cancer, ultrasound guided biopsies will be taken. However, this is still in essence stabbing in the dark, and a large number of biopsy samples (143 in one case) are sometimes necessary to hit the tumor [81].

The prostate is divided into three zones: the peripheral zone (PZ), transitional zone (TZ), and central zone (CZ). The PZ is the largest of the three, accounting for approximately 70% of the prostate, and roughly 70% of CaP cases originate there [170]. Being on the periphery of the prostate, cancer from this region is more likely to spread beyond the prostatic capsule. The TZ and CZ are often considered together as a single region, and as such are referred to as the central gland (CG). Cancer from this area is less likely to spread outside the prostate because it is generally bordered by the PZ and the fibromuscular stroma. Located superior to the prostate are the seminal vesicles (SVs), which are fluid filled sacks. Invasion of the SVs by CaP is also of concern. For the prostate, data is acquired using MRI and three modes are commonly used:  $T_2$ -weighted,  $T_1$ -weighted, and spectroscopy (MRSI). Examples from the three types of MRI acquired for a typical prostate examination are shown in Figure 1.6.

$T_2$ -w images provide good image quality of the prostate, allowing for a differentiation between the PZ and CG. Normal prostatic tissue in the PZ will have high intensity, while cancerous tissue will generally show a decreased intensity. In the CG, however, normal tissue already has a decreased intensity. Due to this, and since the CG is of relatively low importance compared to the PZ, typical MR examinations of  $T_2$ -w data are focused on detecting CaP in the PZ [150]. The prostatic capsule surrounding the gland appears with low intensity in  $T_2$ -w images.

Unlike  $T_2$ -w images,  $T_1$ -w images are of low image quality with respect to the prostate and cannot be used to identify zonal anatomy. Cancerous regions and normal tissue both appear as areas of low intensity, and thus cancerous regions are not apparent. Rather, the  $T_1$ -w images are used to exclude areas which may still contain blood from earlier biopsies. Such post-biopsy hemorrhages will appear similar to cancer in the PZ in  $T_2$ -w images. In  $T_1$ -w images, such regions will have increased intensity from normal prostate tissue.

MRSI measures chemical spectra in large regions, each covering many voxels. For CaP detection, there are three chemicals of interest: choline, creatine, and citrate. Specifically, the ratios of choline to creatine and of choline plus creatine to

citrate appear elevated in cancerous regions [152]. MRSI is not suitable for specific localization due to its coarse resolution, but can be useful as a broad tool.

The surrounding anatomy can also be important in identifying CaP. Invasion of the SVs can be identified using the  $T_2$ -w images. Normal SVs appear as regions of increased intensity surrounded by walls of decreased intensity. In SV invasion, the SVs will appear with decreased intensity throughout. An abnormal angle between the prostate and the rectum can also be indicative of a problem, and thus it is important to be able to view the location of the rectal wall.

A common treatment for organ confined CaP is a radical prostatectomy, where the entire prostate is removed. Imaging modalities to be able to tell if the cancer is indeed organ confined would help in making the decision as to whether this is the proper choice for treatment. This procedure has common undesirable side effects of incontinence and impotence, making it an unpleasant option for many people. More recently, brachytherapy is being used, where radioactive seeds are placed in the prostate near tumors to kill off the cancerous cells. This procedure helps to eliminate the undesired side effects from radical prostatectomy. Imaging technology is needed here to assist in placing the seeds near cancer sites without radiating healthy tissue or important tissues in nearby organs. Imaging techniques could also help to visualize the dosage and coverage information.

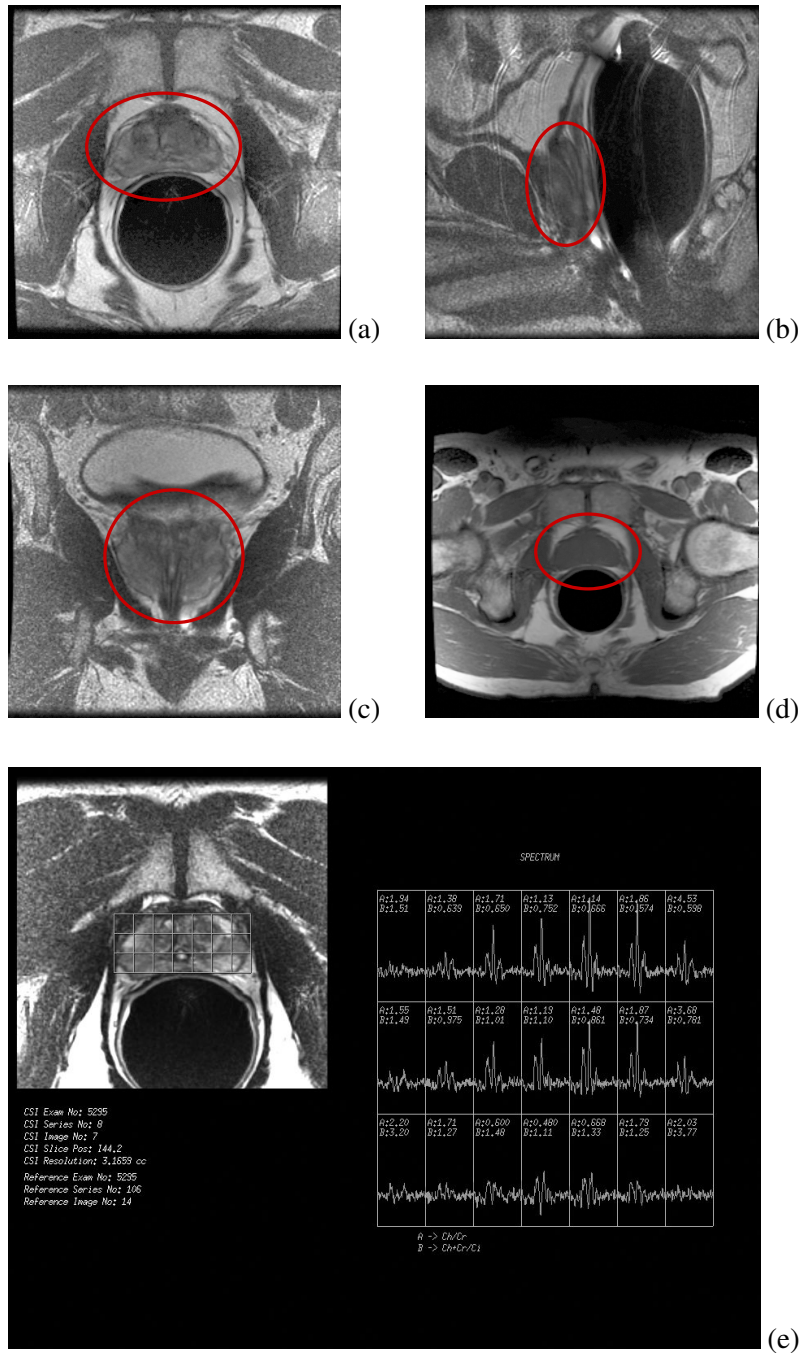
There have been several attempts at automatically segmenting the prostate [171, 172], though manual segmentation still remains the current gold standard. There have also been attempts at CAD using MR images [86] and MRS data [136]. These works are focused on utilizing only the 2D MRI slices, not in a 3D rendered environment, with an outlook towards providing full CAD.

## 1.4 Contributions

This dissertation features the following topics, which address various subtopics in the visualization of volumetric and surface data:

- **Virtually Assisted Optical Colonoscopy:** Two techniques as the beginning steps towards a system combining virtual and optical colonoscopies; undistortion of the fisheye lens and a path correlation between virtual and optical colonoscopies.
- **Supine and Prone Colon Registration:** Given two colon surfaces of the same patient extracted from two scans in different positions, a one-to-one and onto image space registration is found between the two colon models.
- **Contextual Maps of Colon Surfaces:** Given a model of a colon surface, a





**Figure 1.6:** Sample slice from each of the five typical image sequences acquired in MR prostate imaging. The prostate is within the red circular regions. (a)  $T_2$ -weighted endorectal axial. (b)  $T_2$ -weighted endorectal sagittal. (c)  $T_2$ -weighted endorectal coronal. (d)  $T_1$ -weighted pelvic axial. (e) MRSI.

2D map is created without occlusions, providing an undisturbed view of the entire colon surface and locations therein.

- **Contextual Maps of Treelike Structures:** Given a model of a treelike structure, such as the bronchi, a 2D map is created without occlusions which preserves the general shape of the object.
- **Prostate Visualization:** A system to visualize in 3D prostate data obtained from several MRI scans and provide various modes of interaction to allow the user to explore the data.

These works all feature aspects of visualization and exploration to aid users in analyzing the large amounts of medical imaging data acquired from CT and MRI scanners. Though different organs are explored in different specific works, they all focus on the end product of better visualization for the user. This improved visualization includes techniques to combine and view multiple data at the same time as well as succinctly view all data at once. The forthcoming chapters will detail each of these works, including the motivation for each work, the methods developed, and the final visualization results achieved.

## Chapter 2

### Virtually Assisted Optical Colonoscopy

Though VC is a proven system for locating polyps within the colon, OC is still necessary if any polyps are found and need to be removed. Due to this, there is interest in exploring methods of augmenting OC with VC data such that the VC information can be used to guide the OC when a VC reading has already been performed and results are present. In this chapter, two methods are discussed which are beginning steps towards a virtually assisted OC system [95]. These methods include an undistortion of the fisheye view captured by the colonoscope and a correlation between the view paths taken in VC and OC. Though these two techniques can be used together in such a system, their methods and purpose are quite distinct. As such, they are each presented separately in the following discussion.

#### 2.1 Undistortion of Fisheye View

##### 2.1.1 Motivation

The images obtained in OC are captured through a fisheye lens, which causes a high level of radial distortion in the image. Radial undistortion of these images would yield a more normal perspective view, in which the size and shape information from the inside of the colon will be more correctly presented. Since decisions on whether an item on the colon wall is a polyp or not is heavily dependent on size and shape characteristics, this would allow for an improvement in the gastroenterologist's ability to correctly identify potentially cancerous polyps. There has been much research devoted to radial undistortion, with methods using images captured with calibration grids [4, 39, 137] and methods using strictly numerical approaches [84, 87]. In order to maintain an easy to use system, it is best to avoid the need to capture images with calibration grids, and hence a simple numerical method is needed.

### 2.1.2 Method

Radial distortion can be represented using an infinite series, with the distortion then calculated using the equation [84]:

$$F(r) = rf(r) = r(1 + k_1r^2 + k_2r^4 + k_3r^6 + \dots), \quad (2.1)$$

where  $r^2 = x^2 + y^2$ , with  $(x, y)$  being the normalized undistorted projected points in the image frame, and  $k$  are the scalar distortion coefficients. The distorted coordinates in the camera frame can then be calculated as:

$$p_d = p_u \cdot f(r), \quad (2.2)$$

where  $p_u$  are the undistorted coordinates  $(x_u, y_u)$  and  $p_d$  are the distorted coordinates  $(x_d, y_d)$  in the camera frame. It has been found that because the image space, where the work will be performed, contains noise, modeling the distortion above the second distortion coefficient does not improve the results [87], so the distortion can be modeled as:

$$f(r) = 1 + k_1r^2 + k_2r^4. \quad (2.3)$$

It has also been found that the  $r$  values can be reduced [84], such that the distortion can now be modeled more simply as:

$$f(r) = 1 + k_1r + k_2r^2. \quad (2.4)$$

Using this simplified model, the edges of the undistorted image are less prone to distortion artifacts from the image inverting back in on itself.

The work is performed in the image space, not in the space of the camera frame, and so it is necessary to calculate the distortion in the  $(u, v)$  space of the image, rather than in the  $(x, y)$  space of the camera frame. The distortion in the image  $(u, v)$  space can be calculated as:

$$\begin{aligned} u_d - u_0 &= (u - u_0)f(r), \\ v_d - v_0 &= (v - v_0)f(r), \end{aligned} \quad (2.5)$$

where  $(u, v)$  are the image coordinates of the original undistorted image point,  $(u_d, v_d)$  are the coordinates of the corresponding distorted image point, and  $(u_0, v_0)$  are the coordinates of the image center. The adjustment using the image center coordinates is necessary to ensure that the radial distortion occurs around the center of the image, since the  $(x, y)$  coordinates of the pixels will be in the range  $[1, \text{width}]$  for  $x$  and  $[1, \text{height}]$  for  $y$ , with the center point being at  $(\text{width}/2, \text{height}/2)$ . In the camera frame, the coordinates  $(0, 0)$  are at the center, with the values for  $x$  in

the range  $[-x, x]$  and the values for  $y$  in the range  $[-y, y]$ , and hence no adjustment would be necessary.

In calculating the distortion, it is necessary to calculate the value of  $r$ , used in the equation for  $f(r)$  (Equation 2.4). Since  $r^2 = x^2 + y^2$ , this value must be calculated in the 2D projection space of the camera frame, not in the image space. This is accomplished using the affine transformations:

$$x = \frac{u - u_0}{m_u}, \quad y = \frac{v - v_0}{m_v}, \quad (2.6)$$

where  $m_u$  and  $m_v$  are the number of pixels per unit distance in the  $u$  and  $v$  direction, obtained from previous work in colonoscopy calibration [50].

The typical method to undistort an image using these formulae is to solve for the undistorted  $(u, v)$  values for each pixel  $(u_d, v_d)$  in the distorted image. The undistortion is performed on the GPU, using the coordinates of the framebuffer as the output for the undistorted image. Because of this, the problem can be thought of as knowing each pixel location on the undistorted image, and from there calculating where on the distorted input image to obtain the color value from. Using this method, the values for  $(x, y)$  in Equation 2.6 can be easily calculated. Likewise, the distorted pixel locations  $(u_d, v_d)$  can be easily calculated using Equation 2.5 as follows:

$$\begin{aligned} u_d &= (u - u_0)f(r) + u_0, \\ v_d &= (v - v_0)f(r) + v_0. \end{aligned} \quad (2.7)$$

The undistorted image is also formed to be larger than the original, distorted image, as the undistortion pushes the image information past the boundaries in the distorted image.

Rather than locking the scalar distortion coefficient values for  $k_1$  and  $k_2$  to specific values or necessitating individual colonoscopes to be calibrated before use to obtain these values, a simple interface is provided with two thumbwheels to allow for easy adjustment of the two values. Since barrel distortion (the type of radial distortion present in colonoscopes) occurs when the value of  $k < 0$ , the thumbwheels should be adjusted to negative values to perform the undistortion.

### 2.1.3 Results

The undistortion technique was tested on still images which were captured through an Olympus colonoscope, Model CF-Q160L, with a field of view of 140 degrees. The images were acquired of boards with regular geometric patterns on them (such as a checkerboard), so that the effectiveness of the undistortion algorithm could be clearly seen. Figure 2.1 shows examples of performing the undistortion technique on two such images. The radial distortion can be readily observed

in Figure 2.1(a), where the square boxes of a checkerboard have become warped, with the straight lines being curved. In the undistorted images 2.1(c) and 2.1(e), the lines have been straightened and appear as when viewed normally in perspective space. The undistortion technique was also applied to Figure 2.1(b), with similar visual results.

In performing the undistortion, the resulting images are larger than the original images. In essence, the center of the image (where there is very little distortion) has remained the same, while the sides have been pulled outwards. It can be noted from the undistortions that adjusting the value of  $k_2$  in Figures 2.1(e) and 2.1(f) does not produce a noticeably better result than Figures 2.1(c) and 2.1(d), which were created with  $k_2$  ignored (left at a value of 0.0). It can be inferred from this that the use of  $k_2$  is unnecessary, and future tests that were performed utilized only  $k_1$  for adjustments. Being necessary to adjust only  $k_1$  also increases the simplicity of the system for the user.

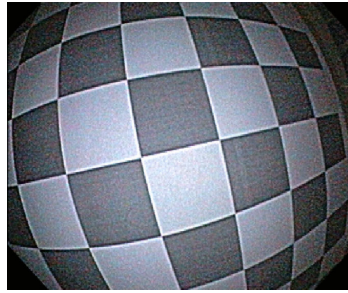
Figure 2.2 shows the result of undistorting an endoscopic image from an OC video of an NIH dataset from the Pickhardt et al. study [111]. The undistorted image in Figure 2.2(b) presents a perspective view more similar to what one would observe in the real world inside the colon if a fisheye lens were not used. The empty black spots at the top, bottom, left, and right are areas that fall outside of the image area in the distorted image, and hence contain no image data. These areas can be ignored, or, if they are found to be too distracting, they can be simply cropped out at the expense of losing some information from the image corners. This undistortion, as well as those in Figure 2.1, were created by experimentally determining the appropriate value for  $k_1$  (and  $k_2$  for Figures 2.1(e) and 2.1(f)) necessary to achieve the desired result.

The radial undistortion is performed on the GPU. The speed of the GPU allows for real-time performance, so the value of  $k_1$  can be adjusted interactively. Interactive performance with an average frame rate of approximately 35 frames per second was achieved using a 3.6 GHz Xeon PC running Windows XP Professional x64 with 4G RAM and an NVIDIA Quadro FX 4500 GPU. This real-time performance also means that it would be possible to perform the radial undistortion on the video from the colonoscope in real-time.

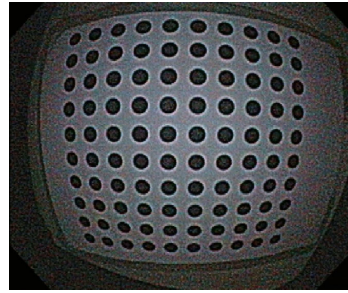
## 2.2 Path Correspondence

### 2.2.1 Motivation

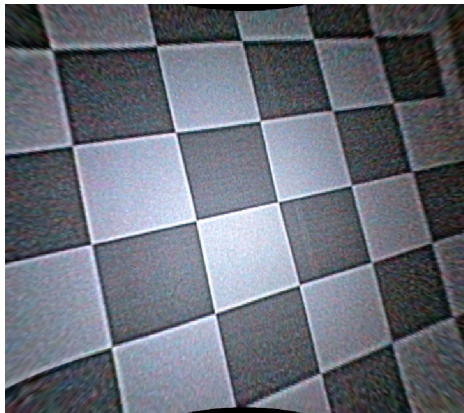
Integrating the information from a VC session or a CAD system can help to ensure that the gastroenterologist does not miss investigating critical sections. This is not trivial, as the exact path of the physical colonoscope is not known. A basic



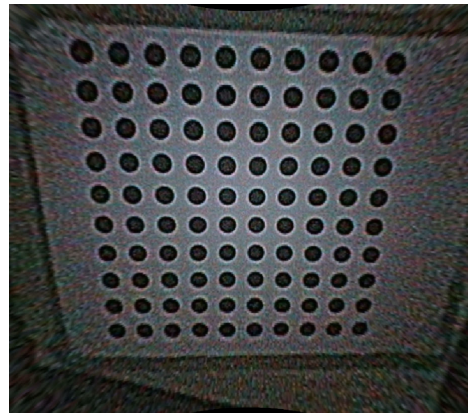
(a) Original distorted image from colonoscope.



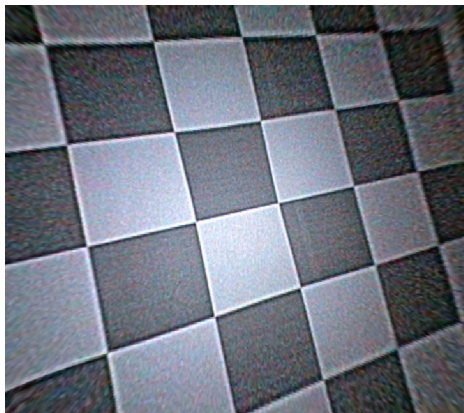
(b) Original distorted image from colonoscope.



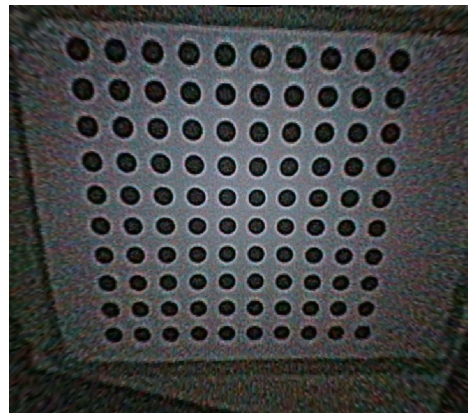
(c) Undistorted image (a) with parameters  $k_1 = -0.024000$  and  $k_2 = 0.000000$ .



(d) Undistorted image (b) with parameters  $k_1 = -0.024000$  and  $k_2 = 0.000000$ .



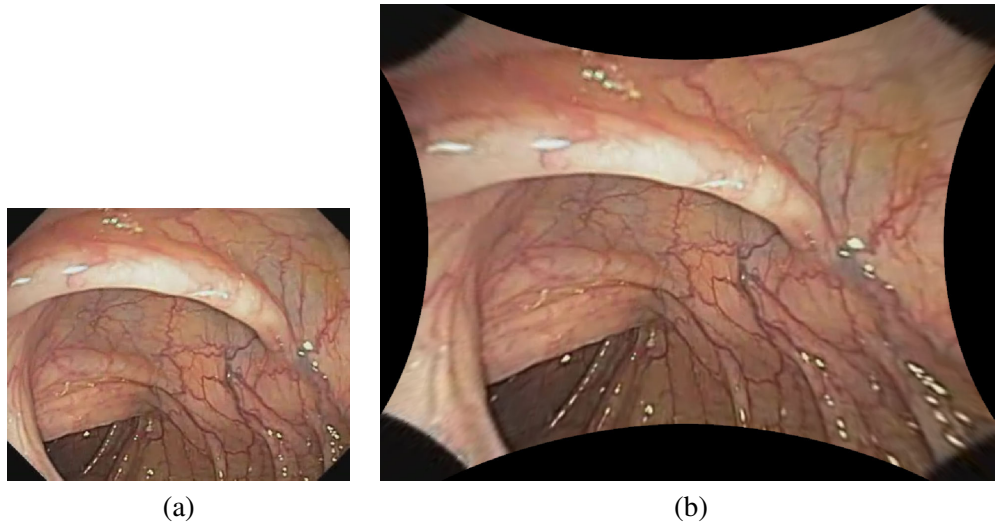
(e) Undistorted image (a) with parameters  $k_1 = -0.027000$  and  $k_2 = 0.000150$ .



(f) Undistorted image (b) with parameters  $k_1 = -0.027000$  and  $k_2 = 0.000175$ .

**Figure 2.1:** Performing undistortion on images of regular geometric patterns captured with a colonoscope.





**Figure 2.2:** (a) Original image of a colon wall captured with an endoscope. (b) Undistorted version of (a).

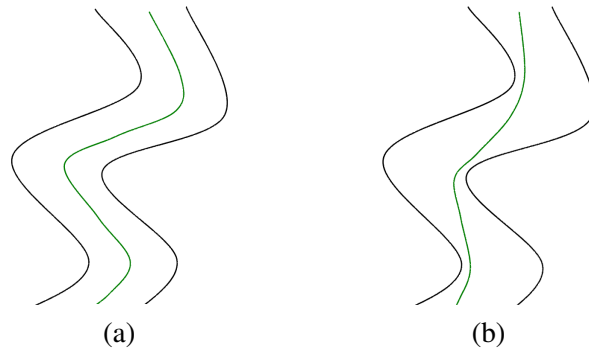
approach is to estimate this path, and correlate it to the centerline calculated for VC. The distance from the rectum along the path can then be matched to a point on the VC centerline. This allows for a rough alignment between the two views.

In addition to the ordinary centerline that is calculated for VC, also calculated is a hugging corner shortest path. This approximates more closely the actual path of the physical colonoscope that is snaked through the patient's colon [50]. An example of the difference between the conventional centerline that is calculated for VC and the hugging corner shortest path that is calculated to approximate the physical colonoscope is shown in Figure 2.3. Figure 2.4 shows the location of the lens on the distal end of a colonoscope (other items on the distal end are not shown). Because the lens center is not directly on the edge of the distal end, the hugging corner shortest path will always remain some minimum distance from the colon wall, never exactly touching the colon surface. This distance can range (in the current example colonoscope) from 2.8 mm to 10 mm, depending on how the colonoscope is oriented, with the average distance being 6.4 mm.

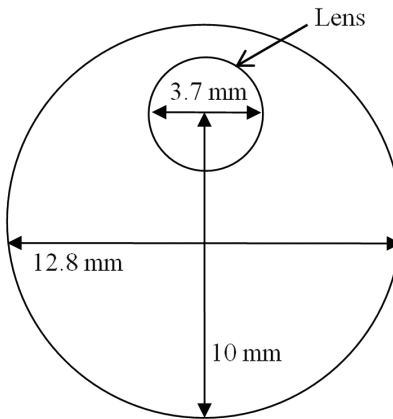
### 2.2.2 Method

The centerline and shortest path are both represented as spline curves, and are discretized into a certain number of points for display and visualization. Knowing what distance the colonoscope is inserted, the discrete point for that location on the shortest path can be calculated. The centerline is correlated to the shortest path





**Figure 2.3:** Examples of (a) a centerline and (b) a hugging corner shortest path.

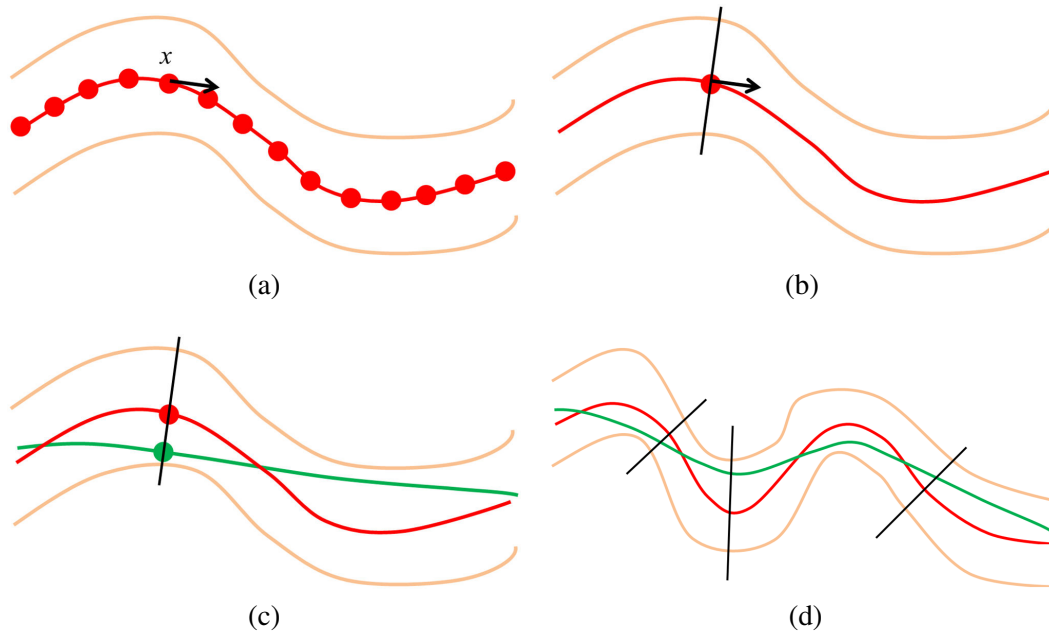


**Figure 2.4:** The location of the lens on the distal end of a colonoscope.

so that any point along the centerline in the VC can be matched to a point on the shortest path in the simulated OC (and vice versa).

The idea of the path correlation is that for each point on one path, a corresponding point on the other path can be found such that the views inside the colon generated from both of these points should be similar. For this, simply finding the nearest point on the other path is not appropriate, as the bends in the paths might make a physically closer point be further away from the area of interest. Rather, it is necessary to find matching points that are in the same cross section of the colon lumen.

Since the centerline follows the contours of the colon more closely than the shortest path, it is used as the starting point in calculating the correlation. The normalized direction of the centerline at a point  $x$  is obtained using the next and previous points on the centerline. To ensure a smooth curve for this calculation, several points before and after  $x$  are averaged and used to calculate the direction



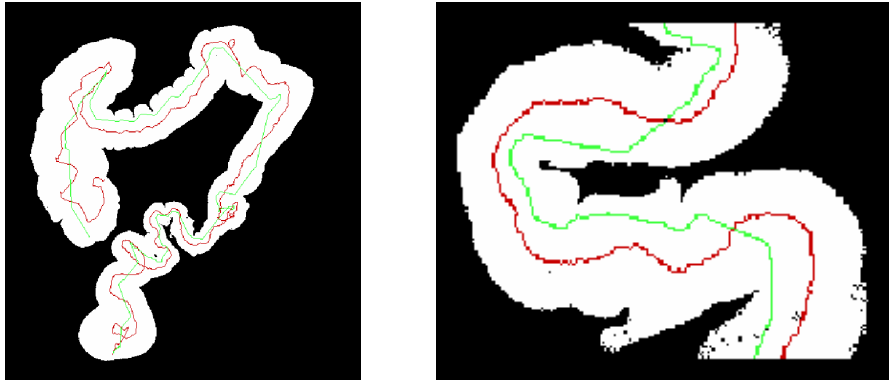
**Figure 2.5:** Method of path correlation. (a) Obtain normalized direction vector for point  $x$  on centerline. (b) Obtain plane orthogonal to point  $x$ . (c) Find nearest point on hugging path which intersects the plane from (b). (d) Examples of three correspondence planes.

vector. This normalized direction vector is then taken to be the normal of a plane that is perpendicular to point  $x$ . Since the centerline closely follows the contours of the colon, this plane can be said to approximate the cross section of the colon which contains point  $x$ . Then the nearest point to  $x$  on the shortest path which is within some tolerance of being on the plane is found. This point  $y$  on the shortest path is then also in the same cross section as point  $x$ . Since they are in the same cross section, points  $x$  and  $y$  can be considered correlated. The method is illustrated in Figure 2.5

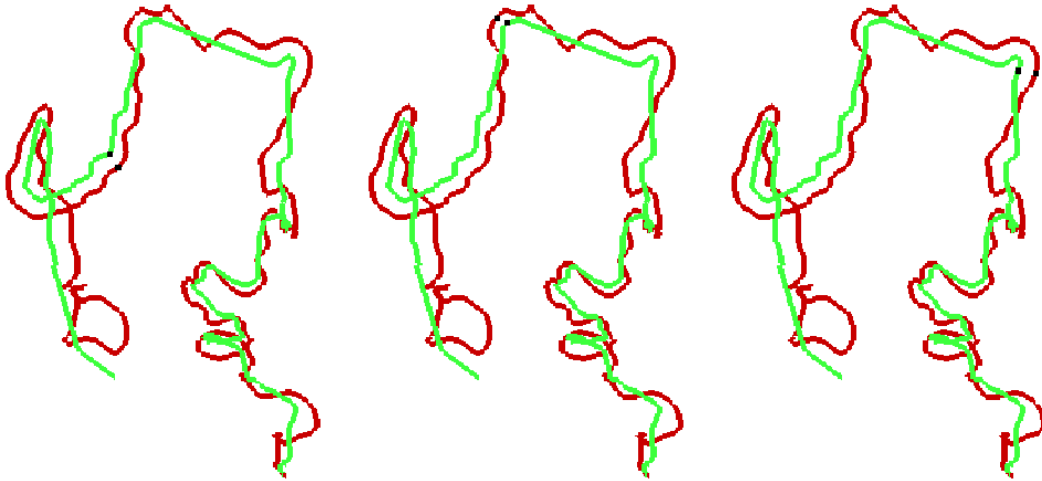
### 2.2.3 Results

To test the path correlation method, the centerline and shortest path that were extracted were used and correlation was checked by comparing the locations of the correlated points and the views from these points inside the colon lumen. Figure 2.6 shows an example of the extracted centerline and hugging corner shortest path with the colon shape. It can be seen that the name hugging corner is appropriate, as the shortest path will stay near the bends of the colon as it curves around, hugging it, whereas the centerline will remain in the center of the colon lumen.

To observe the locations of the corresponding points, just the splines and the



**Figure 2.6:** Example of an extracted centerline (red) and hugging corner shortest path (green) for a virtual colon model, showing the entire model and a zoomed in section.



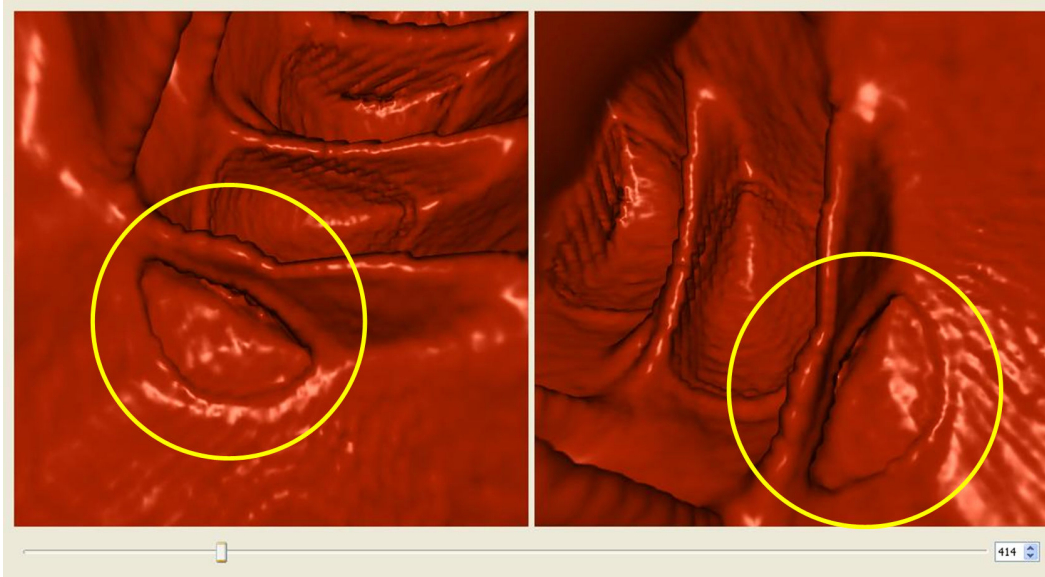
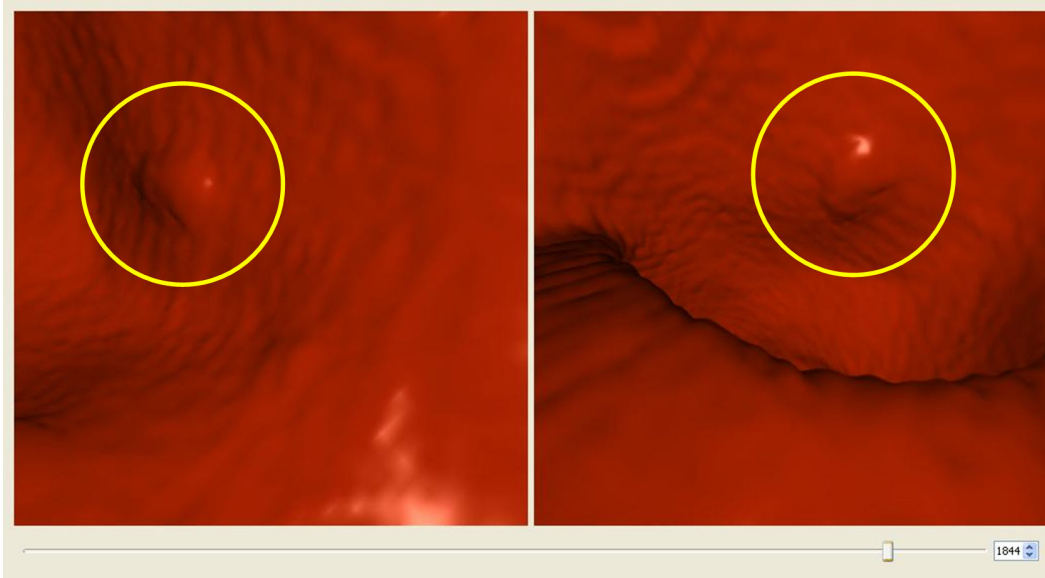
**Figure 2.7:** Three examples of the correlation between the centerline (red) and hugging corner shortest path (green). Calculated corresponding points are illustrated by a black dot on each line.

two corresponding points were visualized. As the point on one spline is moved, the corresponding point on the other spline is updated. The two points should remain close and within the same cross section. Some images of the correlated points can be seen in Figure 2.7. Using this method, it was observed that the two black points appear correlated throughout the length of the colon.

To evaluate how the system might perform in a real OC, the experience was simulated by volume rendering the correlated view from both the centerline (the view from the VC) and from the shortest path (an approximation of the view from

the OC). The two views are shown side-by-side, allowing for easy comparison. Images from this simulation are shown in Figure 2.8. It is possible to identify structures which are present in both views, showing that the two views are properly correlated.

The simple setup would allow for easy insertion of CAD results, where the suspicious areas could be painted on the VC image but not the OC image. Such a method can help ensure that the doctor will observe the entire OC image uninterrupted, but be able to use the CAD information from the VC view.



**Figure 2.8:** Screenshots of the program simulating the correlation between the two views. In both cases, the left image shows the view from the centerline, while the right image shows the view from the shortest path. Landmarks (circled in yellow) can be easily identified in both views, showing that they are properly correlated.

## Chapter 3

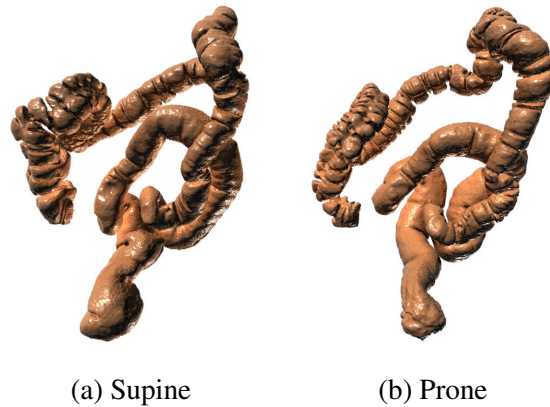
### Supine and Prone Colon Registration

The ability to compare multiple scans of the same patient is an important task in evaluating medical imagery. In this chapter, a technique is presented to register scans of the colon acquired with the patient in both supine and prone orientations [162]. The result of this registration forms a one-to-one mapping between the two colon surfaces, allowing the user to easily identify the same region on both models.

#### 3.1 Motivation

A persistent challenge throughout the development of VC has been the registration of colon models extracted from separate supine and prone scans [1, 22, 102]. It is typical for separate scans to be obtained with the patient in supine (face up) and prone (face down) positions so that regions which may be occluded in one view due to the elastic nature of the colon will be clear in the other. Due to this elasticity, colon models extracted from the two scans will not match (see Figure 3.1). Being able to register these two scans is useful for both a routine VC system and for a CAD system. In the case of a VC system, providing the user the ability to jump from one area in one scan to the same area in the other scan would allow for the easy comparison of these areas when something might be unclear in one of the scans, or for confirming a finding. For a CAD system, a proper registration could help achieve greater accuracy while at the same time reducing false positive results. In terms of visualization, supine-prone registration presents two interesting possibilities: viewing corresponding regions of the two colon surfaces and observing the elastic deformation of the colon between the supine and prone scans.

Early work on supine-prone registration for VC applications focused on using the centerlines and various landmarks. A basic method applies linear stretching and shrinking operations to the centerline, where local extrema are matched and used to drive the deformations [1]. Improved methods of centerline correlation were also investigated for use in reducing false positive results [102]. Correlating individual



**Figure 3.1:** A colon reconstructed from its (a) supine and (b) prone scans.

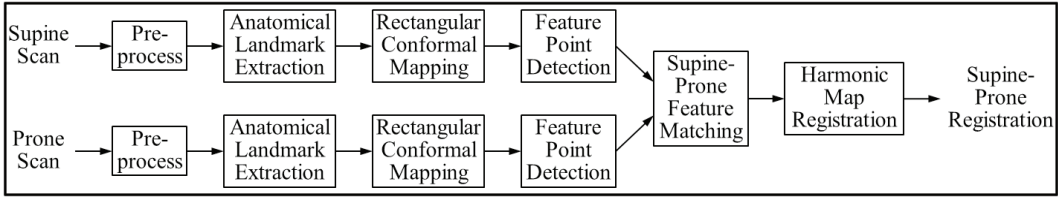
points along the centerline through the use of dynamic programming has also been suggested [22, 101].

More recently, the taeniae coli (three bands of smooth muscle along the colon surface) have been used as features which can be correlated between the two scans [53, 54]. This relies on a manual identification of one of the three taeniae coli, and then an automatic algorithm repeats the line representing the identified taenia coli at equal distances. Further progress has been made where the haustral folds and the points between them can be automatically detected, and the taeniae coli are identified by connecting these points [140]. However, this method is only feasible on the ascending and transverse portions of the colon.

Deformation fields have also been suggested for use in supine-prone registration. Motion vectors can be identified for matched centerline regions, interpolated for non-matched regions, and then propagated to the entire volume [132]. It has also been proposed to use a free-form deformation grid to model the possible changes in the colon shape from supine to prone [114]. For general surface registration, conformal mapping has been used to compute smooth one-to-one maps between closed surfaces of the same topology [75].

The characteristics of the deformation between supine and prone are determined by the elasticity properties of the tissues and muscles. The strain-stress relations for different types of tissues or muscles are different. If the strain-stress relation deformation is independent of the orientation, then local deformation is in a practical sense a scaling, and the global deformation is conformal. In that case, it is possible to flatten each supine and prone surface to a planar rectangle conformally, and those two rectangles should be similar. By aligning the planar rectangles, it is possible to obtain a one-to-one and onto mapping, and obtain the registration between them.

An elastic deformation with bounded angle distortion is referred to as a quasi-



**Figure 3.2:** The pipeline for supine-prone colon registration.

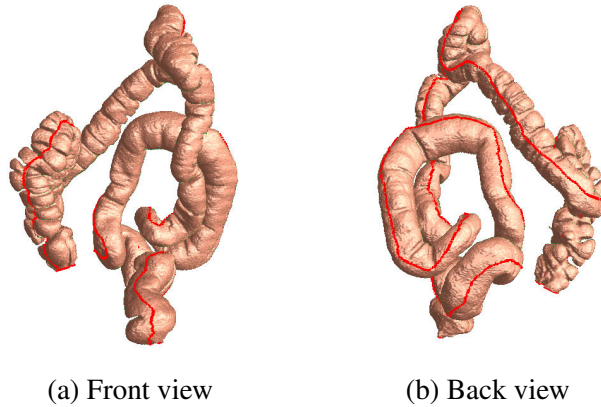
conformal map. The angle distortion for a quasi-conformal map can be represented as a complex valued function, the so-called Beltrami coefficient. The Beltrami coefficient fully controls the quasi-conformal map, which is determined by the elasticity of the underlying material. By better understanding the deformation between supine and prone, it is possible to understand the elasticity of the tissues and muscles and vice versa.

This chapter discusses a method of supine-prone colon registration based on conformal mapping [162]. The pipeline of the algorithm is shown in Figure 3.2. According to conformal geometry theory, there exists an angle preserving map which periodically flattens the colon surface onto a planar rectangle. This mapping minimizes the total stretching energy. Because the conformal mapping is locally shape preserving, it offers an effective way to visualize the entire colon surface and exposes all of the geometric structures hidden in the original shape embedded in 3D.

The non-rigid elastic deformation between supine and prone colons poses a great challenge for shape registration. In this work, the anatomical landmarks and internal features are located and matched on conformally flattened supine and prone surfaces, and a harmonic map is computed with these feature constraints in linear computational complexity. The resulting supine-prone registration is a quasi-conformal map (diffeomorphism), which reflects the elastic deformation of the muscle and tissue. The distortion is evaluated by the Beltrami coefficients. Experiments on 6 pairs of supine-prone colon segments obtained an average  $\mathbb{R}^3$  distance error of 7.85mm for the feature points and polyps evaluated.

Because the deformation from supine to prone is not conformal, the conformal colon flattening method is generalized to a quasi-conformal flattening for registration purposes. Most existing colon registration methods cannot guarantee the mapping to be a diffeomorphism (one-to-one and onto mapping). Due to the complicated geometric structure of the colon surface, the existing methods often introduce folding or singularities in the mapping. This method is based on the biophysical nature of the deformation from supine to prone, which is physically elastic and anisotropic. Due to the material properties of the muscles and tissues, the deformation can be exactly modeled as a quasi-conformal map. The experimental results validate that the mapping is indeed quasi-conformal, and this work is the first to use





**Figure 3.3:** Taenia coli (in red) on the prone colon surface.

quasi-conformal mapping for supine-prone registration.

## 3.2 Anatomical Landmarks

Firstly, anatomical landmarks are extracted which are used in splitting the colon for later processing. These landmarks include the taenia coli and flexures, which are important anatomical landmarks of the colon which do not change despite the change in position of the patient. Corresponding anatomical landmarks in the supine and prone colon models are extracted and then used to cut each colon into its anatomical segments, as well as slicing the colon open for flattening. The main taenia coli is identified and used as a consistent cutting line between the supine and prone colons. This consistent cutting line is important for the robustness of the registration as the boundaries should be the same between the two colon models. The flexures are used to split the colon into five consistent sections which can be processed individually. This splitting reduces the computational burden and allows for better results by having multiple aligned boundaries along the length of the colon.

### 3.2.1 Taeniae Coli

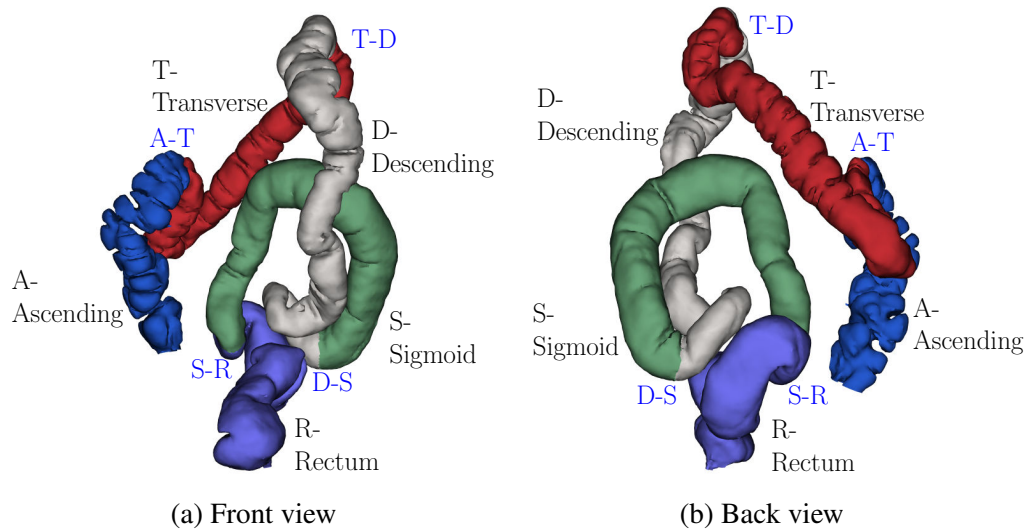
Taeniae coli are three bands of longitudinal muscle on the surface of the colon which run from the appendix to the sigmoid colon and are ideal references for virtual navigation. The three taeniae coli are named taenia omentalis, taenia mesocolica, and taenia libera according to the position on the transverse colon. Taeniae coli can be regarded as ridge breakers for the haustral folds (that is, located between the haustral folds). It is relatively easy to extract the taenia omentalis as it

is clearly visible on the transverse and ascending colons. Similar to reported approaches [17, 69], taeniae coli are identified by detecting the haustral folds, which are computed initially using the heat diffusion and fuzzy C-means clustering algorithms [69]. Using these haustral folds, the taenia omentalis is extracted. The other two taeniae coli are also extracted during this procedure [53]. Although all three bands of taeniae coli are extracted, the taenia omentalis is mostly used for subsequent processing. Figure 3.3 shows the front and back views of a taenia coli on the prone colon surface. Automatic extraction of the taeniae coli through the entire colon is often possible. However, if fully automatic extraction is impossible in some dataset, a semi-automatic method using manually placed markers can be employed to improve reliability.

### 3.2.2 Flexures

The five segments of the colon, starting from the cecum, are the ascending colon (A), the transverse colon (T), the descending colon (D), the sigmoid colon (S), and the rectum (R). A method is needed to identify the locations of the four major flexures between these five segments in the colon. These flexures are further anatomical landmarks which help in virtual navigation and supine-prone alignment and splitting. The first major flexure occurs between the ascending colon and the transverse colon (A-T flexure). This is the flexure close to the liver and is called the hepatic flexure. The second major flexure occurs between the transverse colon and the descending colon (T-D flexure). This flexure is close to the spleen and is named the splenic flexure. The third flexure occurs between the descending colon and the sigmoid (D-S flexure), and the final flexure is between the sigmoid and the rectum (S-R flexure). All of these flexures form very sharp bends and are distinguishable from other smaller bends. Theoretically, the A-T flexure forms the topmost point of the ascending colon and the T-D flexure forms the topmost point of the descending colon.

For the detection of these flexures, the 3D centerline through the colon is utilized. This centerline is projected onto a 2D coordinate system in the positive  $z-x$  and positive  $y-z$  planes. The bends in the centerline are identified by iteratively evaluating the slopes along the projected curves in the two planes. Not all of the bends are important, so small bends are discarded based on a threshold and only the major bends are retained. All of these detected bends are sorted based on their  $z$ -coordinate (up direction). The T-D flexure is identified as the bend with the highest  $z$ -coordinate and the A-T flexure is identified as the bend with second highest  $z$ -coordinate. In the event that the A-T flexure has a higher  $z$ -coordinate than the T-D flexure, the order of the two points along the centerline can be used to confirm the correct identifications. The S-R flexure is the bend with the lowest  $z$ -coordinate.



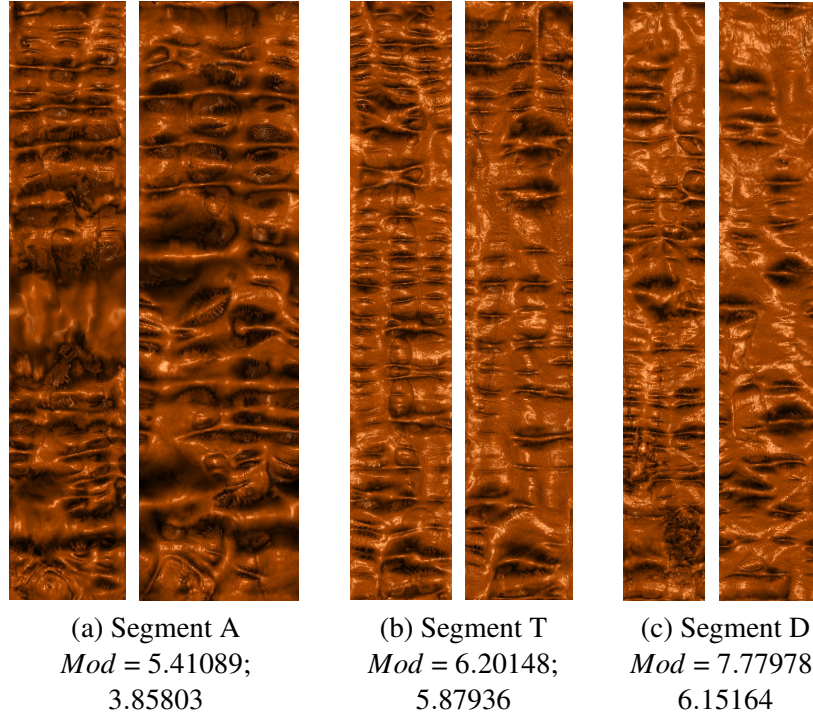
**Figure 3.4:** The four flexures on the prone colon, which divide the colon surface into five segments, depicted in various colors.

The next bend in the sorted order after the T-D flexure whose  $y$ -coordinate is comparable to that of the T-D flexure forms the D-S flexure. These four positions are then mapped back to the 3D coordinate system and the corresponding 3D coordinates of the centerline are obtained.

However, the points that are obtained are those on the centerline and not on the surface. Using these points, the areas on the surface where the flexures are present need to be marked. For each point, a plane is defined which is orthogonal to that point on the centerline. The intersection of the plane with the mesh surface is computed and a polyline is marked by joining the intersection points using the well-known Dijkstra’s shortest path algorithm. Consequently, the flexures on the surface are extracted. The four extracted flexures between the five segments of the colon are illustrated in Figure 3.4 on the prone colon surface.

### 3.3 Surface Features

In order to align the two colon surfaces, it is necessary to identify and correlate feature points on the two surfaces. For this task, the problem of feature detection and matching in 3D is changed into a 2D image matching problem. This is accomplished through the use of conformal mapping. Each colon segment is mapped conformally to a planar rectangle, allowing for all work to then be accomplished within the 2D domain. The results for corresponding segments are shown in Fig-



**Figure 3.5:** Conformal modulus comparison between three different segments of the supine and the corresponding segments of the prone (Ascending, Transverse, and Descending). In each column, the left image shows the flattened rendering of the supine and the right image shows the flattened rendering result of the prone. The conformal module for each segment is defined as the ratio of height to width for the flat rectangle map,  $Mod = height/width$ . Each of the segment pairs are not conformally equivalent.

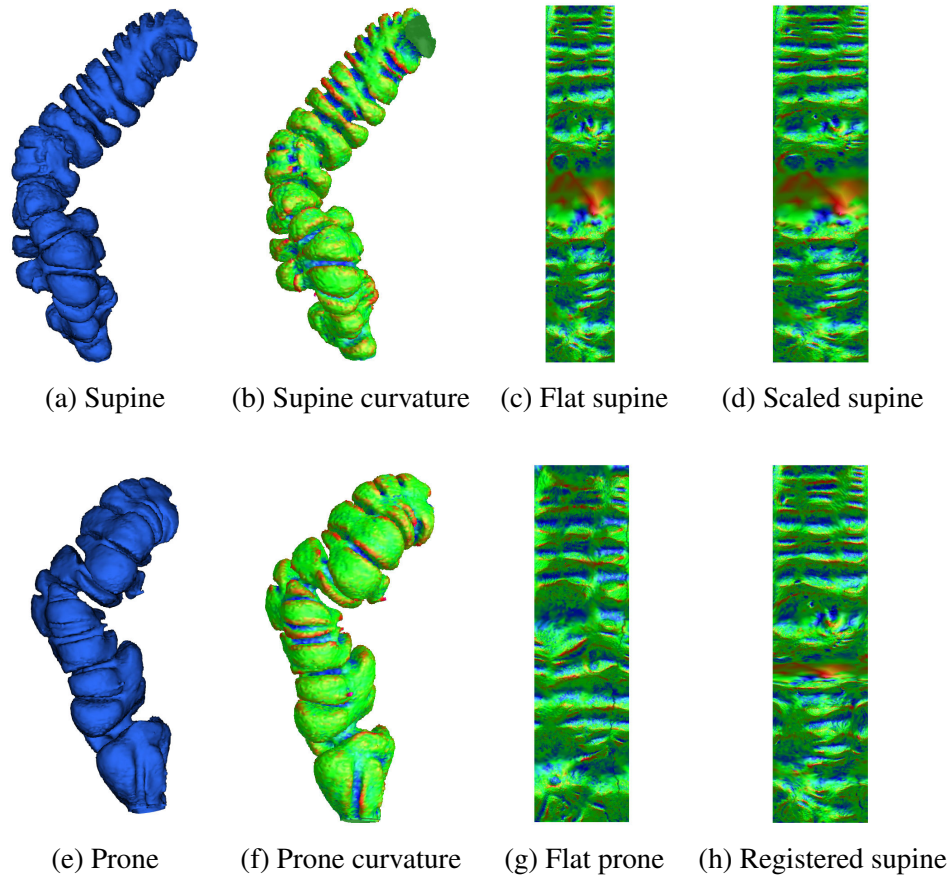
ure 3.5. Since the conformal modulus values of corresponding segments are not equal, the segments are not conformally equivalent, and thus the requirement for additional feature points to align the two segments.

Given a mapping of each colon segment onto a planar rectangle, the mean curvature is color encoded to generate color images. The mean curvature on  $v_i$ ,  $H(v_i)$ , is approximated by:

$$H(v_i) = \sum_{[v_i, v_j] \in E} w_{ij} \langle \mathbf{v}_j - \mathbf{v}_i, \mathbf{n}_i \rangle,$$

where  $\mathbf{n}_i$  is the normal at the vertex  $v_i$ ,  $\langle, \rangle$  is the inner product in  $\mathbb{R}^3$ , and  $w_{ij}$  is the cotangent weight. An example of the color encoded colon surfaces in 3D and 2D are shown in Figure 3.6.

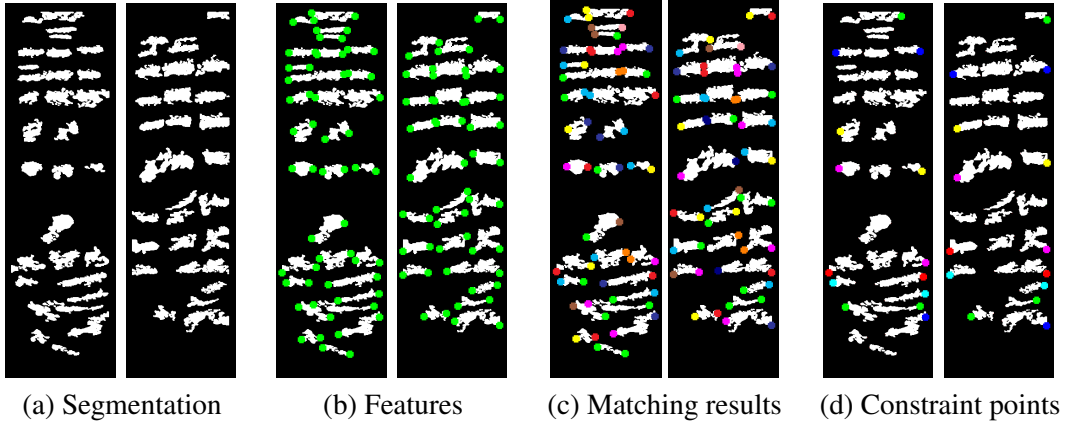
These color coded flattened colon segments are analyzed to obtain specific fea-



**Figure 3.6:** Color encoding mean curvature on corresponding supine and prone segments. (a) and (e) Original colon surface segments. (b) and (f) Color encoded mean curvatures. (c) and (g) Color encoded mean curvature on the flattened surfaces. (d) Scaled flat image of (c) with the same size as (g). (h) Deformed flat image of (c) after registration to (g).

ture points for the registration. The well-known graph cut approach from computer vision is used to find the regions of interest. In the case of the colon, the desired regions of interest on the flattened supine and prone images are the folds. A graph is constructed using the pixels of the image as the nodes. Assuming a virtual sink and source, edges are constructed with appropriately assigned weights. Using the max-flow min-cut method to solve the energy minimization problem [12], all of the pixels belonging to the folds are obtained. The appropriate assignment of weights ensures efficient detection of the folds.

Due to the drastic change in position of the patient in supine and prone, the colon tends to stretch, contract, and move, resulting in the distortion of the folds. In addition, some areas might be densely populated with folds, while other areas are



**Figure 3.7:** Feature detection and matching for the flat images between supine and prone (see Figures 3.6 (d) and (g)). In each frame, supine is on the left and prone is on the right. (a) Segmentation results using graph cut. (b) Feature detection results (in green). (c) Feature matching results using graph matching. (d) Matching features used to constrain the registration. Two corresponding feature points are encoded in the same color on the supine and prone flat images in (c) and (d).

sparsely populated. To prevent any ambiguities, only the significant folds are chosen which have a lower chance of distortion. Therefore, all of the folds whose length and size are below a certain threshold are discarded. For this, a pre-processing step is performed using a breadth-first search to find all of the connected components among the extracted folds. The prominent folds form larger connected components and thus by defining an experimental threshold value, all the smaller components are discarded. By varying the threshold, it is possible to modify the number of detected folds based on the requirements. Figure 3.7(a) shows the segmentation results for both the supine and prone surfaces.

After applying the threshold all of the significant folds are now detected. These folds can be approximated as ellipses and hence the axial end points form good feature points for registration. These axial points of the detected folds are extracted and form the feature point set. Figure 3.7(b) shows the detected features (in green) on both supine and prone surfaces.

Feature matching is performed on each segment of the colon separately. Finding a correspondence between the extracted feature points in supine and prone is again formulated as an energy minimization problem by defining an objective function. The dual decomposition approach is employed for energy optimization, which is solved as an instance of the well-known graph matching problem [138].

Suppose the feature point sets are  $S_1$  and  $S_2$  on supine and prone respectively,

and the matching is  $\phi : S_1 \rightarrow S_2$ , then the objective function is defined as:

$$E(\phi) = \lambda \sum_{p \in S_1} |p - \phi(p)|^2 + (1 - \lambda) \sum_{q \in S_2 - \phi(S_1)} |q|^2,$$

where  $\lambda$  is determined experimentally.

The objective function is chosen depending on the geometrical compatibility of feature correspondences and the spatial coherence of the matched features. It is defined by considering the weighted sum of two energy terms. The first term includes the geometric distance between the feature points within a pre-defined neighborhood. The second term evaluates the effect of including the distance of the unmatched features in the objective function. Minimization of this function using the dual decomposition approach will result in a unique set of feature correspondences.

Due to the large deformation between supine and prone, their folding patterns are highly inconsistent. Figure 3.7(a) shows that the number of significant folds is different for supine and prone. Therefore, it is impossible to find the correspondence for all feature points on supine and prone, as shown in Figure 3.7(b) and (c). In the graph matching algorithm, it is attempted to try and utilize all of the matched features, and ignore those that are unmatched. In practice, it was found that the feature points near the taenia coli (that is, the border of the rectangle) are more reliable and accurate than those in the middle. Therefore, the algorithm emphasizes the feature points near the borders more than those in the middle. The experimental results demonstrate that the registration result is satisfactory even without the unreliable middle feature points. The points which are used as constraints for the alignment step of the registration algorithm are shown in Figure 3.7(d).

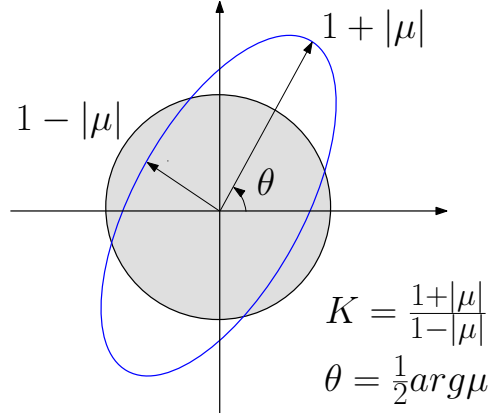
### 3.4 Quasi-Conformal Mapping

Since the mappings of supine and prone segments are not conformally equivalent (see Figure 3.5), a quasi-conformal registration between the two segments is necessary [160]. Let  $S$  be a surface embedded in  $\mathbb{R}^3$  and  $\mathbf{g}$  is the induced Euclidean metric. Given two surfaces with Riemannian metrics  $(S_1, \mathbf{g}_1)$  and  $(S_2, \mathbf{g}_2)$ , then  $\phi : S_1 \rightarrow S_2$  is a diffeomorphism between them. It is said that  $\phi$  is a conformal map if the pull back metric induced by  $\phi$  satisfies the following relation:

$$\phi^* \mathbf{g}_2 = e^{2\lambda} \mathbf{g}_1.$$

Intuitively, this means that the conformal mapping  $\phi$  maps infinitesimal circles on  $S_1$  to infinitesimal circles on  $S_2$ . In general, a diffeomorphism  $\phi$  maps infinitesimal ellipses on  $S_1$  to infinitesimal circles on  $S_2$ . As shown in Figure 3.8, the eccentricity





**Figure 3.8:** Geometric meaning for Beltrami coefficients  $\mu$ . The quasi-conformal deformation can be measured by the local stretching  $K$  and the angle distortion  $\theta$  through mapping infinitesimal circles to ellipses.

and the orientation of the ellipse can be described by a complex differential  $\mu$ , which is called the *Beltrami coefficient*. The *dilatation*  $K$  is defined as:

$$K = \sup_{p \in S_1} \frac{1 + |\mu(p)|}{1 - |\mu(p)|},$$

which gives the upper bound of the distortion (local stretching). A *quasi-conformal (QC) map* with dilatation  $K$  is called a  $K$ -QC map.

It is possible to choose special local complex coordinates  $z$  of  $S_1$  and  $w$  of  $S_2$  such that  $\mathbf{g}_1 = e^{2\eta_1} dzd\bar{z}$  and  $\mathbf{g}_2 = e^{2\eta_2} dwd\bar{w}$ . Then, the local representation of  $\phi : z \rightarrow w$  satisfies the following *Beltrami equation*:

$$\frac{\partial \phi}{\partial \bar{z}} d\bar{z} = \mu \frac{\partial \phi}{\partial z} dz, \quad (3.1)$$

where  $|\mu| < 1$ . If  $\mu$  is zero, then the mapping is conformal. The fundamental theorem in quasi-conformal geometry claims that the solution to the Beltrami equation exists and is essentially unique.

Due to the elastic deformation of the colon, the deformation from supine to prone is not conformal, but is a  $K$ -QC mapping. Furthermore, the Beltrami coefficient  $|\mu|$  shows the anisotropy of the deformation at each point, which reflects the anatomic characteristic of the colon wall tissue in the neighborhood. Given the detected and matched feature points, it is possible to achieve this quasi-conformal mapping between the supine and prone colon segments.

Suppose that the feature points are  $\{p_0, p_1, \dots, p_n\}$  on the supine surface and  $\{q_0, q_1, \dots, q_n\}$  on the prone surface, such that  $p_k$  corresponds to  $q_k$ . In order to



enforce the alignments among these features, an affine map,  $\eta : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , is first used which maps the rectangle of the supine to that of the prone. Then two harmonic functions are computed,  $\mathbf{h} = (h_1, h_2)$ , such that:

$$\mathbf{h}(p_k) = \phi_2(q_k) - \eta \circ \phi_1(p_k), 0 \leq k \leq n,$$

and furthermore:

$$\Delta h_1 = 0, h_1|_{\gamma_2 \cup \gamma_4} = 0, \frac{\partial h_1}{\partial \mathbf{n}}|_{\gamma_1 \cup \gamma_3} = 0,$$

and:

$$\Delta h_2 = 0, h_2|_{\gamma_1 \cup \gamma_3} = 0, \frac{\partial h_2}{\partial \mathbf{n}}|_{\gamma_2 \cup \gamma_4} = 0.$$

Then, the final registration map from supine to prone,  $\Phi : S_1 \rightarrow S_2$ , is:

$$\Phi := \phi_2^{-1} \circ (\mathbf{h} + \eta \circ \phi_1).$$

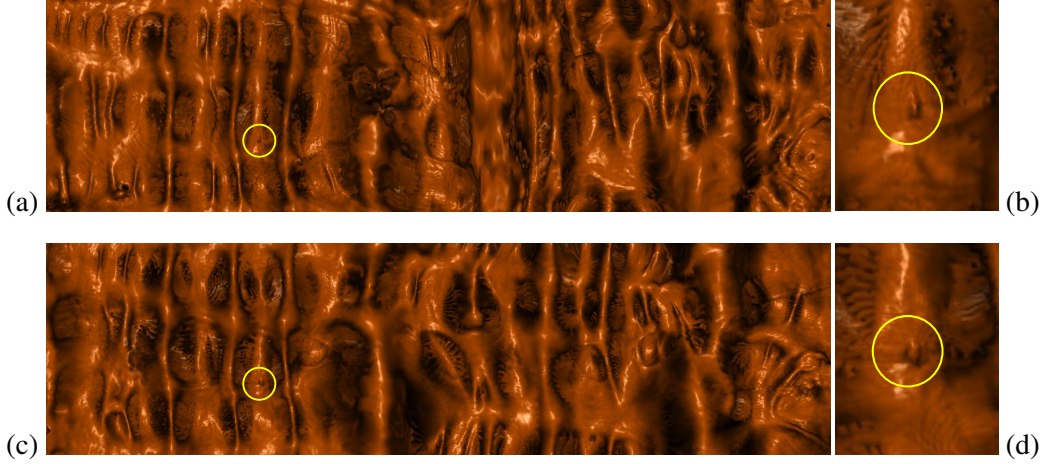
This step adjusts the conformal mapping to be quasi-conformal and has linear complexity. This quasi-conformal correction is necessary due to the elastic deformation of the colon. As the elastic deformation of the colon between supine and prone cannot be known before registration, this value is not used in creating the map. Rather, the feature points are used to guide the creation of a quasi-conformal map that will approximate the elastic deformations of the colon segments.

## 3.5 Results and Visualization

### 3.5.1 Quantitative Results

Two methods are used to quantitatively evaluate the registration results. The first evaluation is a distance measurement between corresponding features located on the registered colon surfaces. Since the registration is in the 2D space using the flattened colon surfaces, a point  $(x, y, z)$  in  $\mathbb{R}^3$  on the original colon surface will be at a location  $(u, v)$  in  $\mathbb{R}^2$  on the registered surface. Thus, for a point  $p_0 = (u_p, v_p)$  on the supine colon surface and its corresponding point  $q_0 = (u_q, v_q)$  on the prone colon surface, a perfect registration with zero error is present when  $u_p = u_q$  and  $v_p = v_q$ . For two corresponding points, the  $L^2$  norm of their 2D coordinates is computed with the width of the flattened images fixed to a unit length of 1. Note that the width along the flattened image is equivalent to the colon circumference for that location.

Also computed is the 3D distance error in millimeters. For the two corresponding points  $p_0$  and  $q_0$  in  $\mathbb{R}^2$ , their locations  $r_0$  and  $s_0$  in  $\mathbb{R}^3$  are known. If the supine



**Figure 3.9:** Visual verification for supine-prone colon registration using volume rendering. (a) and (c) The flattened views of the ascending colon segments of the supine and prone surfaces, respectively. After registration, a polyp found on the flattened supine surface (yellow circle in (a)) can be located on the flattened prone surface (yellow circle in (c)) at nearly the same position. (b) and (d) Enlargements of the neighborhoods of the same polyp on the supine and prone surfaces, respectively.

surface (containing  $p_0$ ) is taken as the truth and the measurement of the registration error on the prone surface (containing  $q_0$ ) is desired, the point  $p_1 = (u_q, v_q)$  in  $\mathbb{R}^2$  can be identified on the supine surface and similarly its location  $r_1$  in  $\mathbb{R}^3$ . The distance error in millimeters is then given to be  $|r_1 - r_0|$ .

The corresponding feature points for evaluation are obtained in two ways. The first is by manually identifying points of interest (e.g., a polyp) on both colon surfaces. The centers of regions of interest on the registered flat images are manually selected. For the polyp shown in Figure 3.9, the distance error is 0.0265 in  $\mathbb{R}^2$  and 5.31 mm in  $\mathbb{R}^3$ . The algorithm has been evaluated using a total of 20 pairs on 6 datasets, and the average distance error is 0.0305 in  $\mathbb{R}^2$  and 5.65 mm in  $\mathbb{R}^3$ .

The second method of finding corresponding points is to use a subset (in this case, half) of the features obtained from the feature extraction and matching process. These features are used for evaluation purposes, and only the remaining half are used in the harmonic mapping step. Note that this inherently reduces the quality of the registration, and thus a registration using all feature points will contain less error. In general, 16 pairs of features points are computed for each segment. For the registration shown in Figure 3.9, the distance error in terms of feature points evaluation is computed to be 0.0325 in  $\mathbb{R}^2$  and 7.51 mm in  $\mathbb{R}^3$ . The average  $\mathbb{R}^3$  distance on 6 colon segments is 7.85 mm.

A comparison between the current method and other methods is performed us-

**Table 3.1:** Statistics for conformal mapping.

Colon	Segment	# Vertices	# Faces	Time(s)
Supine	Ascending (A)	35,251	70,366	21
	Transverse (T)	43,255	86,364	27
	Descending (D)	44,910	89,659	23
	Sigmoid (S)	37,237	74,296	22
	Rectum (R)	39,712	79,315	23
Prone	Ascending (A)	36,629	73,139	20
	Transverse (T)	42,452	84,781	25
	Descending (D)	44,587	88,981	26
	Sigmoid (S)	38,499	76,823	22
	Rectum (R)	38,177	76,276	23

**Table 3.2:** Computational time for the supine and prone feature matching (including feature extraction) and the harmonic map registration.

Time (s)	A	T	D	S	R
Feature matching (FM)	18	19	21	18	16
Harmonic mapping (HM)	10	11	12	10	10

ing the analytic evaluation results in  $\mathbb{R}^3$ . For those papers that present their distance error, a comparison of their results with the current results is shown in Table 3.3. The current method produces a registration with significantly smaller distance error between corresponding points than other methods.

### 3.5.2 Corresponding Visualization

Perhaps a better indicator of the utility of this registration is a visual evaluation of the results, as this mimics most closely how the user of a VC system would use the results. For this, both the flattened rendering and the 3D correlated endoluminal renderings are utilized. It was observed that the correlation between the flattened renderings was good, as was the endoluminal views. Figure 3.10 shows two supine-prone colon segments, flat rendered, both unregistered and registered. The images of the registered segments clearly show very good alignment of the supine and prone colon structures, whereas the unregistered segments show poor alignment.

Another use of the one-to-one and onto mapping is the ability to create a correspondence between two views in the endoluminal view. In order to visualize both supine and prone in a consistent manner, the camera frustum viewing the supine inner surface (mucosa) is mapped to the camera frustum viewing the prone mucosa.

**Table 3.3:** Comparison of average distance error between the presented quasi-conformal registration method and other registration methods.

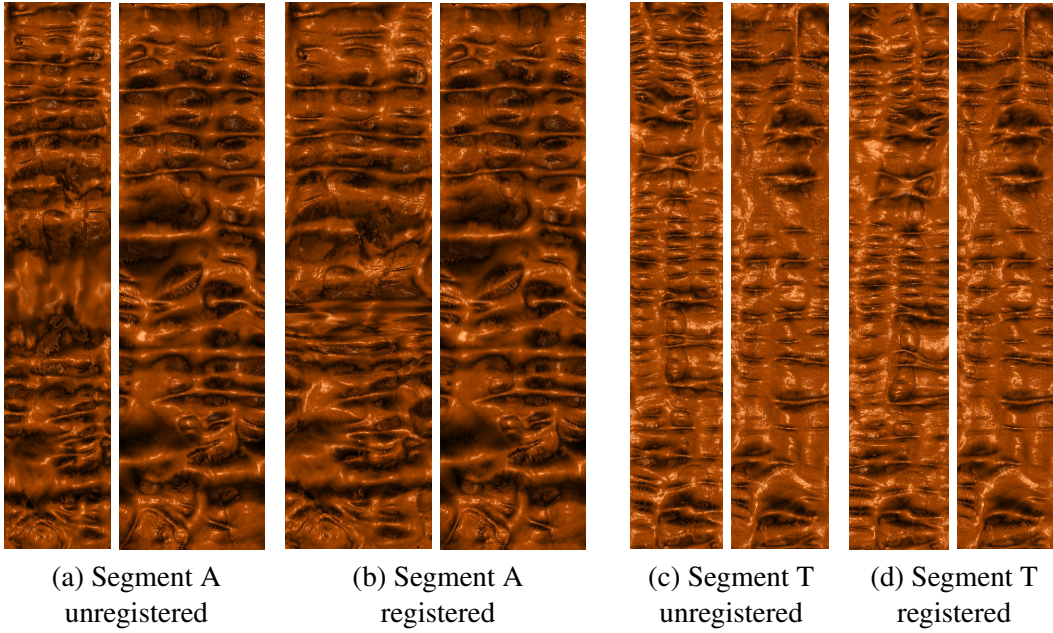
Methods	Dist. Error
<b>Presented Quasi-Conformal Mapping</b>	<b>7.85 mm</b>
Centerline registration + statistical analysis (Li et al., 2004 [73])	12.66 mm
Linear stretching / shrinking of centerline (Acar et al., 2001 [1])	13.20 mm
Centerline feature matching + lumen deformation (Suh & Wyatt., 2009 [132])	13.77 mm
Centerline point correlation (de Vries et al., 2006 [22])	20.00 mm
Taeniae coli correlation (Huang et al., 2005 [53])	23.33 mm

This is achieved by first matching the skeletons through the entire colon lumen, and then matching the camera orthonormal coordinate frames.

A new path through the colon is introduced and referred to as the *flattened centerline*. With a colon segment mapped to a plane such that the segment boundaries are on top and bottom (resulting in a plane taller than it is wide), horizontal line segments then correspond to loop curves on the colon surface. The mass center of a loop is taken to be the central point for that plane of the colon segment. By calculating these mass centers along the entire flattened colon, a path of points is generated which generally mimics that of a centerline. However, since this is calculated from the flattened colons, it is referred to as the flattened centerline. To avoid confusion, the term *skeleton* is used to refer to the 3D medial axis centerline which is conventionally calculated for VC systems.

The flattened centerline can be efficiently calculated as a series of points through the colon lumen using the GPU to render the flattened colon mesh. For a desired number of points in the flattened centerline, the colon mesh is scaled so the longest side of the rectangular mesh is equal to this desired number of points. The mesh is rendered such that the color of each vertex is the original coordinates of the vertex in the volume, with the values interpolated to fill the triangles. Each row of pixels can then be averaged to obtain the mass center for that row, which is the point for that position on the flattened centerline.

The skeletons are matched through the use of the flattened centerlines. Given a point  $c_0$  on the discretized supine skeleton, it is necessary to find the corresponding point  $c_1$  on the discretized prone skeleton. First the point  $d_0$  is found on the flattened centerline, which is the point nearest  $c_0$  in  $\mathbb{R}^3$ . Given this point  $d_0$ , the

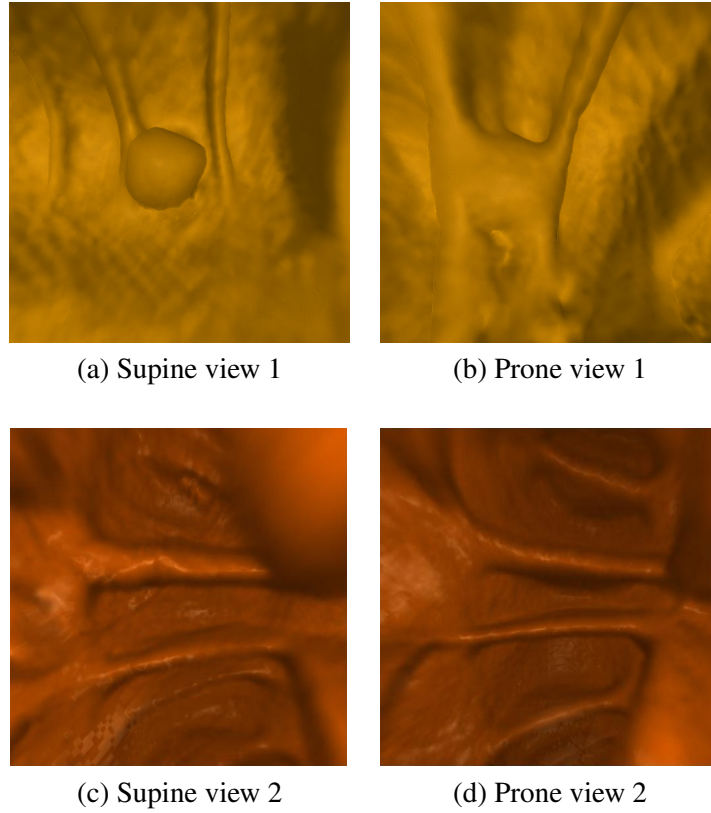


**Figure 3.10:** Visualization of registration results for segments A and T with 16 and 15 feature constraints, respectively. For each image pair, supine is shown on the left and prone on the right.

corresponding point  $d_1$  on the prone flattened centerline is found at the same index.  $c_1$  can then be found as the point on the prone skeleton closest in  $\mathbb{R}^3$  to  $d_1$ .

The endoluminal correlation work is focused on situations where the user is looking at something on the colon wall and wants to see the corresponding location in the other scan. Using the correspondence along the skeletons, it is possible to create a correlated automatic navigation, though running two fly-throughs side-by-side was found to be distracting and more of a hindrance than help. Orienting rotation around the skeleton for two automatic navigation views is possible by adapting the below described technique into a pre-processing step for each point along the discretized path.

It is assumed that the optical center of the camera is at the skeleton as in a typical VC navigation. Suppose  $o$  is the optical center of the camera for supine and  $c$  is the point corresponding to  $o$  on the skeleton of prone. Furthermore, the  $z$ -axis of the camera for supine intersects the surface at the point  $p_0$ , and the  $y$ -axis intersects the supine at  $p_1$ , then let  $q_0$  and  $q_1$  be the corresponding points on the prone. Then, it



**Figure 3.11:** Consistent endoluminal views for supine and prone colons.

is possible compute the frame of the camera on prone as follows:

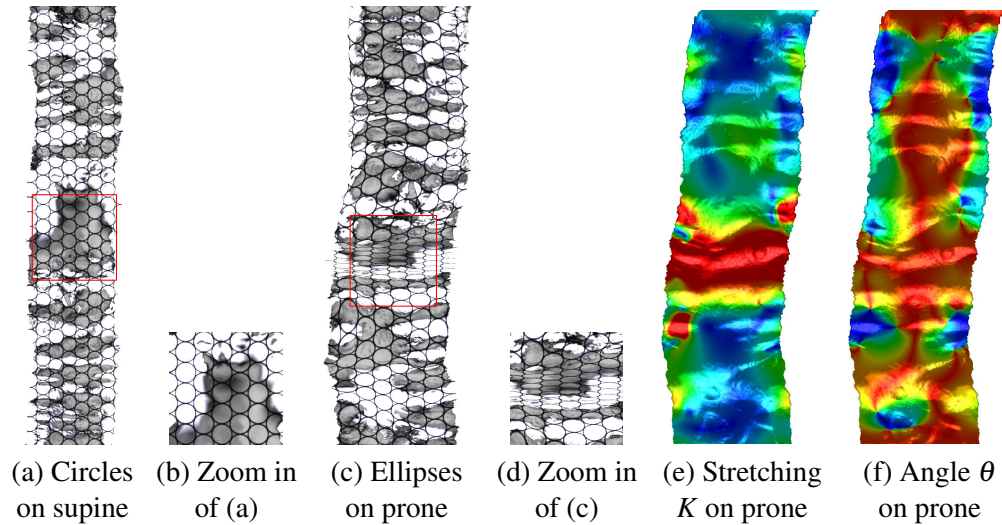
$$\begin{aligned} \mathbf{e}_3 &= \frac{q_0 - c}{|q_0 - c|}, \\ \mathbf{e}_1 &= \frac{(q_1 - c) \times \mathbf{e}_3}{|(q_1 - c) \times \mathbf{e}_3|}, \\ \mathbf{e}_2 &= \mathbf{e}_3 \times \mathbf{e}_1, \end{aligned}$$

and then the camera frame is given by  $\{c; \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ . These camera coordinates and frame are used by a GPU accelerated ray caster to render the final endoluminal images, allowing the user to jump from a position on the supine to the same position on the prone. Figure 3.11 demonstrates the consistent volume rendering results using this camera frame matching method.

### 3.5.3 Visualization of Quasi-Conformality

The elastic deformation between supine and prone is measured by the Beltrami coefficients in Equation 3.1 through the resulting registration. In general, this information would likely not be of interest to a radiologist reading a VC study. However,



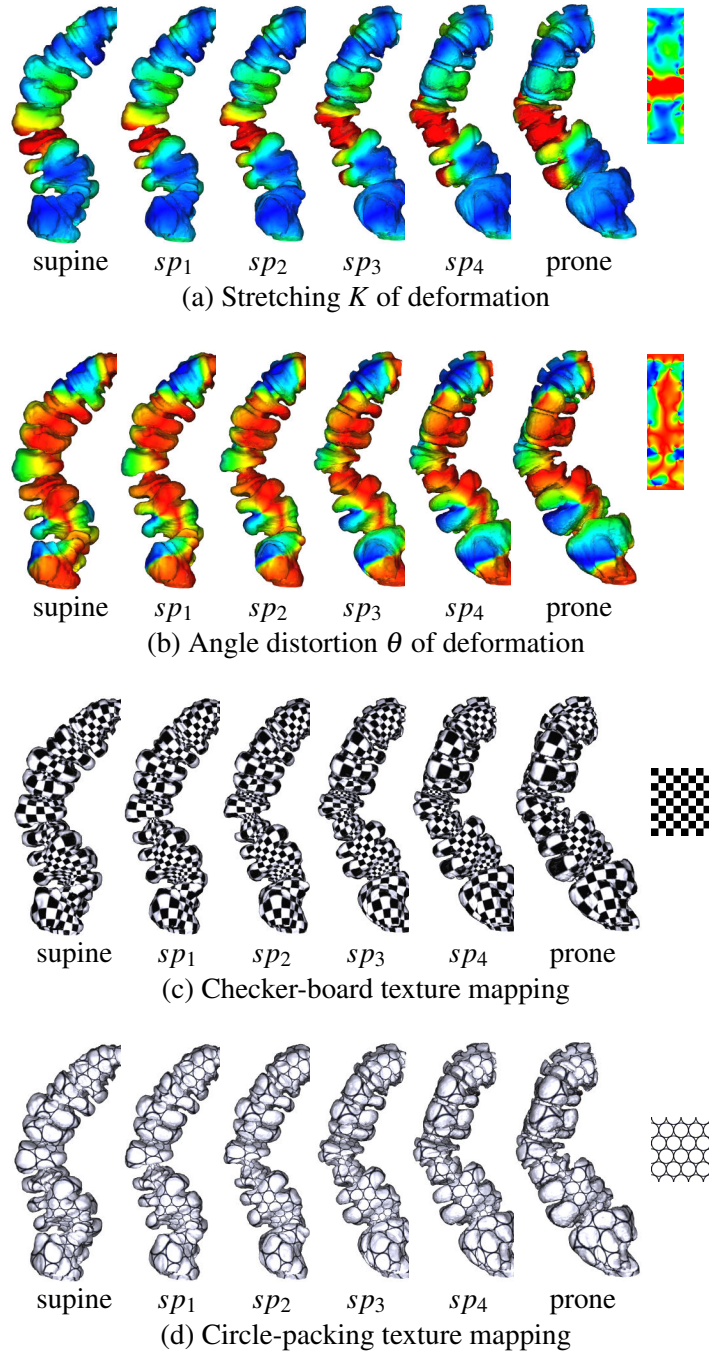


**Figure 3.12:** Visualization of quasi-conformality. The (a) circles on the source (supine) are mapped to (c) ellipses on the target (prone) by the registration map. The narrower ellipses correspond to the region with large deformation (in red) in the local (e) stretching and (f) angle distortions.

visualizing the amount of distortion in this way may be useful to a doctor seeking to better understand how the colon deforms in certain regions.

The quasi-conformal mapping deforms circles to ellipses. Figure 3.12 illustrates that the deformation from supine to prone is quasi-conformal. A general fundamental domain is chosen for the supine segment. Its flat coordinates are used as the texture coordinates and the circle-packing texture mapping shown in (a) is computed, where the circularity is preserved. After registration to the flattened map of the prone segment, the supine planar image is changed and the circles are accordingly deformed to ellipses, shown in (c). Using the one-to-one and onto mapping between supine and prone, the Beltrami coefficients can be efficiently computed on the triangular mesh surface. The quasi-conformality of the mapping is evaluated by the local stretching and angle distortion. Their color encoded texture images are shown in (e) and (f), respectively. The maximal local stretching is 11, and therefore mathematically called *11-Quasi-Conformal Mapping*.

Based on the diffeomorphism constructed between supine and prone, the deformation process can be simulated by a linearly interpolated morphing sequence. Then, the texture coordinates are transferred to each intermediate mesh surface consistently. Through the texture correspondence, it is possible to visually gain a better understanding of how the deformation behaves from supine to prone and vice versa. After registration, the supine and prone have the same mesh connectivity, but dif-



**Figure 3.13:** Visualization of the Beltrami coefficients. The views  $sp_1, sp_2, sp_3, sp_4$  are generated by linear interpolation of the registration. In (a) and (b), red indicates the largest stretching/angle distortion and blue indicates the least. The quasi-conformality of the deformation is visualized by sequences (c) and (d).



ferent geometry. As shown in Figure 3.13, the morphing views  $sp_1, sp_2, sp_3, sp_4$  between supine and prone are generated by the linear interpolation of geometry at each vertex. Directly from the stretching-colored views (a), one can easily comprehend where the main deformation happens. The global surface distortion is mainly affected by the local stretching deformation, valued by the ratio between the long axis and short axis of the ellipse locally. The local angle distortion, however, is intrinsic to the material properties of the organ muscle and tissue. Compared with the stretching, the angle distribution is more uniform through the whole deformation, as shown in (b). Here, red denotes the largest stretching (angle distortion), blue the lowest stretching, and yellow and green in between. The consistent texture mapping of checker-board (c) and circle-packing (d) textures also demonstrates the quasi-conformality of the deformation.

## Chapter 4

### Contextual Maps of Colon Surfaces

This chapter presents a method of creating a 2D parameterization (or map) of a colon surface such that the entire surface can be viewed at once, there are no overlaps, and the global shape of the patient's specific colon is preserved [96].

#### 4.1 Motivation

In a VC system and other virtual endoscopic systems, it is typical to use a 3D mesh model of the organ's surface as a map. This map can be used to indicate the current view position and present other information such as the location of user defined bookmarks or CAD results. This is important in the colon since the surface often looks similar at different regions, and therefore it can be difficult to identify one's location by simply observing the endoluminal view. However, using a 3D mesh model for this purpose leads to its own problem, where sections can occlude each other when they exist in the same region on the viewing plane and at different depths along the view direction (see Figures 4.2(a) and 4.3). Such problems necessitate a rotation of the 3D structure, distracting the user from their main task. A solution to this problem would be to find a mapping of the colon's surface to a plane in such a way that occlusions do not exist. Such a mapping, however, should preserve the global shape of the colon and therefore make it immediately recognizable to the user. In essence, it is desirable to present a view which is similar to the diagrammatic view utilized clinically for labeling regions (see Figure 4.1), but which is tailored to a specific colon.

A flattened representation of a colon can be useful for providing an overview of the entire structure in a single 2D view. This can allow for a virtual dissection of the surface, permitting the user to view the entire colon at once, and can also allow for a mapping with no occlusions. However, common methods of creating such views will straighten the object, such that it is difficult to locate where positions would correspond to by simply looking at the flattened representation. This is evident, for



**Figure 4.1:** Common diagram view of the colon used in clinical practice.

example, in the case of flattening a colon to a long linear strip. Though the 3D colon structure is often classified into separate sections based on the major flexures it contains, this information is lost in the flattened representation. While labels could be introduced to identify each of the flexures, such a technique would still not retain the simplicity of viewing a recognizable colon shape.

There has been varied work on flattening the colon and other tubular structures from the medical domain to planar representations. It has been proposed to virtually unfold the surface by rendering rays cylindrically around the centerline [7]. A similar technique that also preserves some z-depth information along the flattened mesh has been proposed [78]. Conformal mapping of the colon surface to the plane has also been utilized for triangular mesh models [37], and conformal mapping with volume rendering for image generation has been presented [47]. Flattening vessels to the plane is also a subject of interest, and similar techniques have been applied to flatten vessels in an angle preserving manner [169]. Further flattening and exploding of vessels into diagrams has also been shown to improve users' abilities to detect areas of interest [11]. However, all of these methods will greatly distort the global shape of the object and cannot be considered ideal for a contextual map of the colon.

The skeleton (centerline in the case of the colon) is a good descriptor of shape and has great importance in many applications. Because of its utility, there has been much research in methods of accurately extracting the skeleton of an object. Skeleton extraction from volumetric medical data by connecting centerlines based on a medial voxel path has been presented [168]. A method has been proposed using a distance from boundary field with penalty weights to obtain general skeletons of tree-like structures [10]. Obtaining a skeleton directly from a mesh by contracting the surface, without any need for a volumetric representation, has been suggested

[63]. Both surface and curve skeletons from genus 0 structures have been extracted through the use of advection to move mass from the object's surface to its skeleton [119]. The use of least squares optimization has been suggested for extracting curve skeletons in an efficient manner which is insensitive to noise [148]. Curve skeletons can also be approximated for data from incomplete point clouds based on a generalized rotational symmetry axis for a set of points [135].

There are many uses for skeleton curves in the fields of computer graphics and visualization, and their 1D representation of 3D objects is useful for compact shape descriptions, navigation, animation, and analysis [19]. Skeleton curves can be used to drive shape deformations [156]. They have also been used in describing shape to allow for the creation of shape-based transfer functions [115]. They are commonly used for navigation in endoscopic visualizations, such as virtual colonoscopy [146], and accurate curves are important for such kinds of navigation [142]. Due to the importance of the skeleton curve and its ability to well represent the shape of an object, it is reasonable to use the centerline as the basis for creating a context preserving map of the colon.

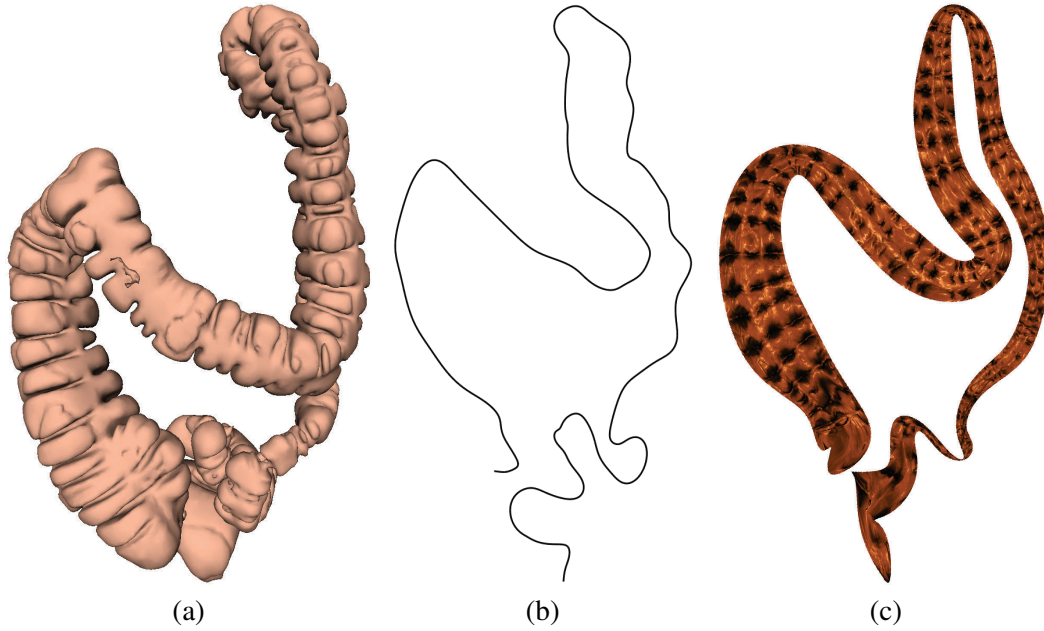
This chapter presents a method of creating flattened maps of the colon surface which preserve the geometric context of the overall colon structure [96]. The general idea is illustrated in Figure 4.2. The centerline of the original 3D colon model is first extracted. This centerline is then projected into 2D and molded such that it preserves the general shape it had in 3D, but without intersections. Finally, this molded centerline is used to create a boundary placement in 2D for the final mapping and the colon surface is mapped to this 2D boundary. Although this has been developed with a focus on the colon, it is generalizable to other tubular structures.

## 4.2 Skeleton Layout

The creation of a context preserving map is based upon manipulating the skeleton of the structure, which can be found using any of the standard methods. Given the 3D skeleton, it is first projected to an optimal 2D plane. This projection is then corrected for length, untangled to remove intersections, and spaced properly to prevent self-intersections of the structure's surface during the later mapping step. All operations are performed on a discretized version of the skeleton with an initial equal 1mm spacing between points.

### 4.2.1 Orthogonal Projection

The original 3D skeleton of the structure is first projected to a plane, such that all  $z$  values become 0. Since this projection is to be used as the basis for a map, per-



**Figure 4.2:** Context preserving map of a colon. (a) The original colon mesh from a standard view point. (b) The projected 2D skeleton of the colon after untangling and spacing. (c) The context preserving flattened map of the colon surface based on the 2D skeleton in (b).

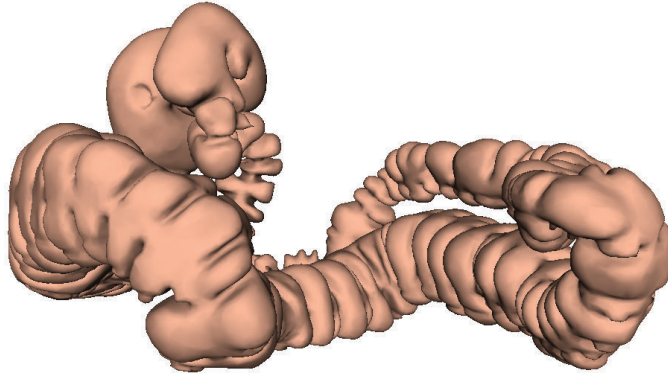
spective distortion is unnecessary and even detrimental, and thus orthogonal projection is used. The projection is performed using the standard orthogonal projection matrix:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (4.1)$$

where  $u$  and  $v$  are the right and up vectors of the projection plane.

The projection plane is chosen to be at the orientation most commonly expected for the domain experts who will be viewing the map. In the case of the human colon, the projection plane would be expected to be parallel to and above the body of a patient lying in a supine position. Assuming that a patient in 3D space is lying oriented with their head towards the negative  $z$ -axis and their body facing up in the positive  $y$ -axis, the projection plane will have  $u = (1, 0, 0)$  and  $v = (0, 0, -1)$ . The position and orientation information for the CT slices can be obtained directly from the DICOM header information, allowing for the view plane to be determined automatically.

The 2D projected skeleton of a colon is shown in Figure 4.4(a). Although the focus here is on the colon, for most types of objects, especially in medical appli-



**Figure 4.3:** Colon surface from Figure 4.2(a) viewed from above. From this vantage point, it is easy to see how two long sections of the colon occlude each other when viewed from the front or back.

cations, a canonical view is known which provides the best view and is familiar to the user. If such a view is not known a priori, a plane which minimizes structural overlap and skeletal intersections can be used.

#### 4.2.2 Lengthening

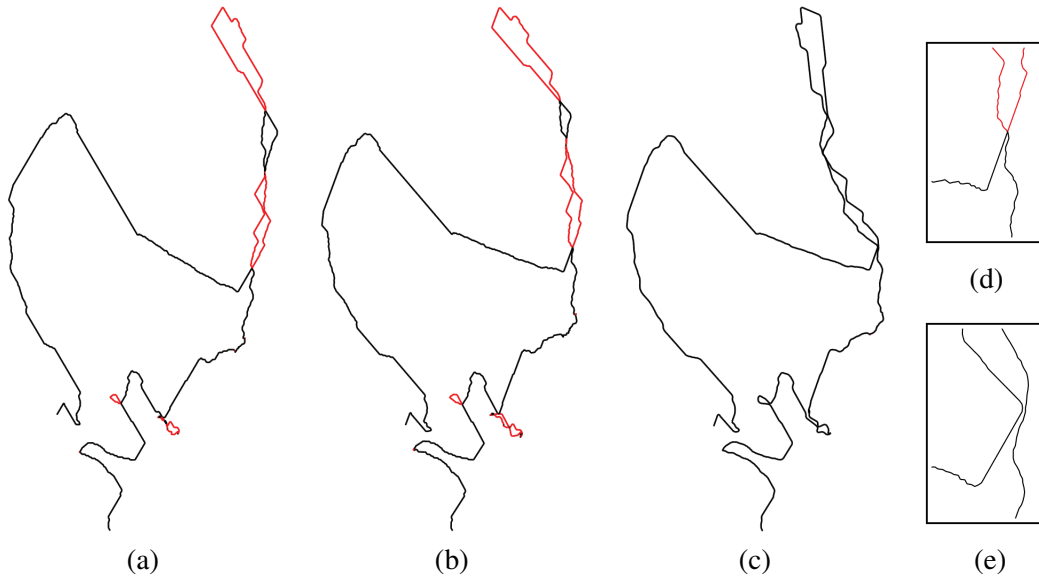
When performing the orthogonal projection, the lengths of portions of the skeleton can be substantially diminished if the segment direction is aligned with the view direction of the projection plane. The closer to parallel the segment direction and view direction are, the greater the loss of length. To recover this length in the 2D skeleton, a lengthening procedure is applied that bends a straight skeleton with even spacing into the shape of the projected skeleton.

For every point along the projected skeleton (except endpoints), its two neighboring skeleton points are used to create two vectors  $v_1$  and  $v_2$ , and the angle between them is calculated in the standard way based on the law of cosines:

$$\theta = \cos^{-1} \frac{v_1 \cdot v_2}{|v_1||v_2|}. \quad (4.2)$$

All of the points in the skeleton are then set in a straight line, with equal spacing between them equivalent to the original spacing of the 3D discretized skeleton. Starting at the beginning, the corresponding angle  $\theta$  is used to rotate all of the following points  $p = (x, y)$  around the current point  $p_0 = (x_0, y_0)$  as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}. \quad (4.3)$$



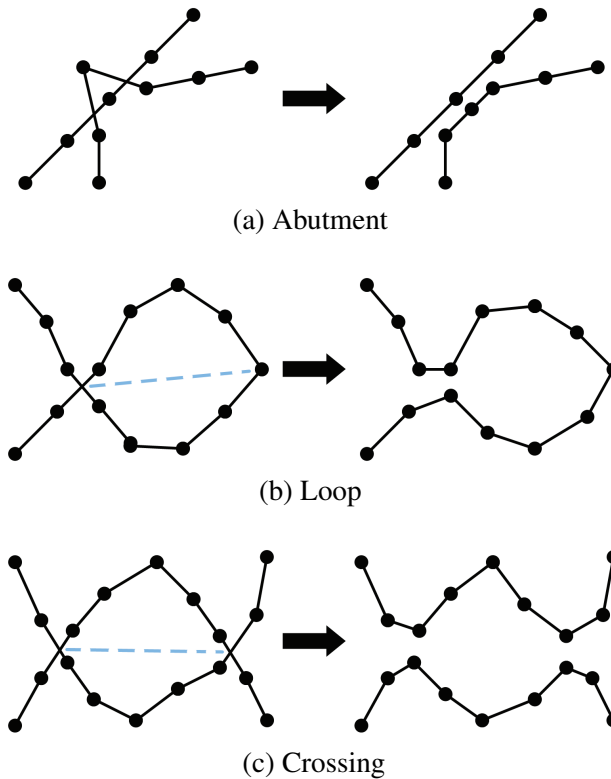
**Figure 4.4:** Skeleton of the colon shown in Figure 4.2. When intersections along the skeleton are present, alternating regions between intersections are colored black and red. (a) Projected 2D skeleton. (b) Lengthened projected 2D skeleton. (c) Untangled skeleton. (d) Close-up of an intersection region from (b). (e) Close-up of the same region from (c). Note that though from a distance the skeleton in (c) still appears to have intersections, the view in (e) shows that they are in actuality just locations where the skeleton sections are very close.

If the original rotation was clockwise instead of counterclockwise in relation to the normal of the projection plane, then the transpose of the rotation matrix is used accordingly. This is performed for all points along the skeleton until the end is reached. After this process, the length of the skeleton from its original 3D shape is preserved, while the geometric structure of the projected 2D skeleton is also preserved. The lengthened 2D skeleton of a colon is shown in Figure 4.4(b).

### 4.2.3 Untangling

Given the lengthened 2D projected skeleton, it is then necessary to untangle it so that no intersections are present. This process, however, should maintain the general shape of the structure (compare Figures 4.4(b)-(c)). This is done by looking for three types of intersections: loops, crossings, and abutments. An example of each type of intersection and the corresponding result of untangling is shown in Figure 4.5. The three types of intersections are as follows:

**Abutment:** An abutment (see Figure 4.5(a)) occurs when a center point among



**Figure 4.5:** Types of intersections and their untangling results.

three consecutive discrete skeleton points lies on the opposite side of another line segment of the skeleton than the other two points. This causes both line segments formed by the three points to intersect the skeleton at a single line segment, and thus contains two points of intersection.

**Loop:** A loop (see Figure 4.5(b)) occurs when a skeleton comes around and crosses over itself without any other intersections within the formed loop. This creates one intersection point.

**Crossing:** A crossing (see Figure 4.5(c)) occurs when two sections of a skeleton intersect each other without any loops or abutments between them. This contains two intersection points.

To remove these intersections, all intersections of the skeleton are first found. Each intersection is actually located twice within the list, once for each time it is encountered when tracing the path. The list of intersections is then searched to locate the three different types listed above. Given that an intersection is made of two intersecting line segments in the discrete case, each intersection is represented by four



points, the first two points representing the endpoints of the first line segment and the second two points representing the endpoints of the second line segment along the skeleton. The points themselves are ordered as their order along the skeleton. Each of the three intersections can be identified as follows:

**Abutment:** An abutment is located by finding two consecutive intersections for the same line segment (intersecting two separate line segments).

**Loop:** A loop is located as two consecutive intersections where the pair of line segments for the two intersections is the same.

**Crossing:** A crossing is located as two consecutive intersections that do not meet either of the above two conditions.

After searching the list of intersections, the first intersection of the highest priority that is found is corrected. The intersections in order from high to low priority are abutment, loop, and crossing. Each type of intersection is corrected as follows:

**Abutment:** An abutment is removed by simply setting the offending point to the average of its two neighbor points.

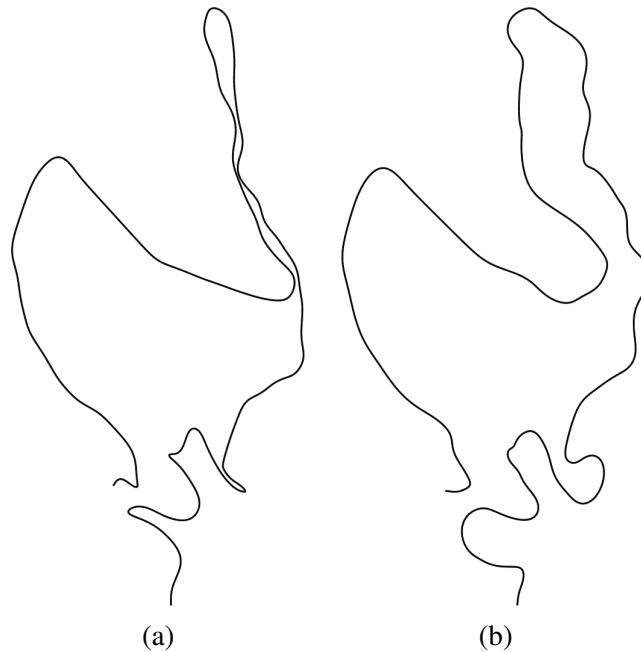
**Loop:** A loop is removed by first creating a line segment from the loop intersection point to the median point of the loop. All points within the loop are then reflected through this line segment in order to untangle it.

**Crossing:** A crossing is removed similarly to a loop. A line segment spanning the two intersection points is found. All points within the crossing, on both segments of the skeleton, are then reflected through this line segment in order to untangle it.

Since performing one correction can cause new intersections to arise, the skeleton is searched again after every correction procedure to find all intersections. This is repeated until no intersections are found along the skeleton. After untangling, the skeleton is iteratively smoothed to remove small sharp features (see Figure 4.6(a)).

#### 4.2.4 Spacing

After the untangling procedure, several segments can lie very close to each other due to the orientation of the projection plane. These close areas must be spaced out so that there is room for the colon surface boundary when flattened. The radius of the structure along the skeleton is recorded, and the 2D skeleton points are translated perpendicular to their direction such that the radius segment from every point



**Figure 4.6:** Refined 2D skeleton of the colon shown in Figure 4.2. (a) The smoothed version of the untangled 2D skeleton. (b) The spaced and smoothed version of the 2D skeleton.

does not intersect other sections of the skeleton. For each offset of the skeleton points, if an intersection is created, the offset is undone. In this way, no new intersections will occur. Neighboring skeleton points are also moved slightly to help avoid sharp bends. At the end, iterative smoothing is applied to the entire skeleton to yield a clean path. An example of the result of this process is shown in Figure 4.6(b).

### 4.3 Boundary Placement

Given the untangled skeleton path of the structure, it is necessary to place the boundary for the flattened surface in relation to it. In addition to preserving the overall shape of the colon, it is also desirable to represent the general size of the colon along the skeleton. This is accomplished by calculating the approximate radius along the skeleton and using it to control the layout of the boundary. This has the effect of condensing the surface perpendicular to the skeleton, but the overall relation of size between regions can still be observed.

### 4.3.1 Slicing

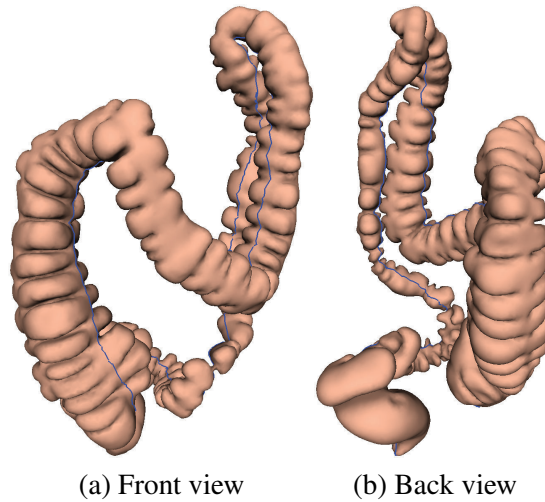
In order to map a tubular structure to a plane, it is necessary for the structure to have a single boundary. The structure is discretely represented by a closed surface triangular mesh (i.e., without boundaries), which is initially topologically equivalent to the sphere in  $\mathbb{R}^3$  (it has an Euler characteristic number  $\chi = 2$ ). To make the surface instead topologically equivalent to a plane in  $\mathbb{R}^3$ , it must be sliced open from one end to the other, creating a single boundary. Although this boundary could technically be located anywhere, it should match the skeleton on the original structure.

To locate the path along which the mesh is to be sliced open, Dijkstra's shortest path algorithm is used. This is advantageous for structures with features such as bumps and valleys, through which the cutting line should not pass, as Dijkstra's algorithm naturally tends to avoid these areas and instead locate a path along more uniform regions. In the case of the colon, this means that the slicing path will generally not go through features such as the haustral folds. For Dijkstra's algorithm, the starting point is taken to be the vertex on the surface mesh nearest to one end of the 3D skeleton. The target point can then be taken as usual as the surface vertex with the greatest distance from the source. Alternatively, the surface vertex located nearest to the other end of the skeleton can be taken as the target.

Given a path along the surface to use for slicing the surface open, two small holes are punched at each end of the path by removing one face for each. This yields a structure that is now topologically equivalent to a cylinder ( $\chi = 0$ ). Slicing along the cutting path is now equivalent to slicing a cylinder from one boundary to another, yielding a surface with a single boundary ( $\chi = 1$ ) which is topologically equivalent to the plane. For every vertex along the slicing path, a new vertex is duplicated with the same surface position. The sliced surface with the boundary in blue is shown in Figure 4.7.

### 4.3.2 Skeletal Correspondence

As the boundary vertices are mapped to a surface around the 2D skeleton, it is necessary for each boundary vertex to have a correspondence to a position along the skeleton. Two corresponding boundary vertices (i.e., an original vertex from the unsliced mesh and a newly created vertex from the slicing at the same position) are both associated with the same position along the skeleton. The four end vertices on the boundary that result from punching holes at each end are treated as corner points and are not associated with the skeleton, as they do not contribute to the mapped shape. For all boundary vertices (other than corresponding pairs as mentioned), no two boundary points can be associated with the same location along the skeleton.



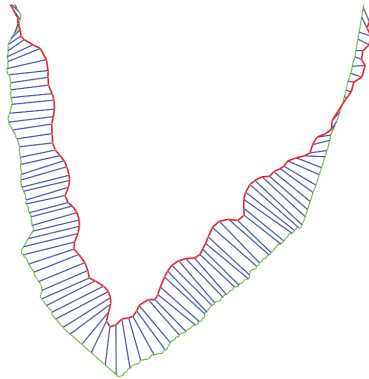
**Figure 4.7:** Slicing path (in blue) on the 3D surface of the colon shown in Figure 4.2.

Since the twists and turns of the colon can often lead to situations where the closest skeletal point for a boundary vertex is actually in a separate section, the location of correspondences between the skeleton and boundary is created based on finding the nearest boundary point for each skeleton point, determining boundary to skeleton correspondences from those, and then interpolating positions for boundary points which were not indicated by a skeleton point. The correspondences are determined for the set of original surface vertices on the slicing path, and then duplicated to the newly created vertices. The first and last vertices along the slicing path are mapped to the first and last position along the skeleton. The algorithm is as follows:

```

for all skeleton points do
  Find nearest boundary vertex
end for
for all vertices on slicing path do
  Find all skeleton points indicated as correspondences
  Take center of those points as true correspondence
end for
for all vertices on slicing path without correspondence do
  Interpolate neighbors
end for
  
```

The correspondences (excluding the two end points) are then smoothed through several iterations by setting the current boundary's correspondence position to the



**Figure 4.8:** Correspondence between boundary vertex points and skeleton points. The skeleton path is shown in green, the slicing path in red, and the connections between a boundary point and the skeleton in blue.

average of its two neighbors. The results of this correspondence are shown in Figure 4.8.

### 4.3.3 Radius Along Skeleton

The radius for each point along the skeleton is determined based on the correspondence with the boundary vertices explained above. The algorithm is as follows:

```

for all skeleton points  $s$  do
  for all boundary vertices  $v$  do
    if  $v$  corresponds to segment containing  $s$  then
      Accumulate weighted distance
      Accumulate weight
    end if
  end for
end for
for all skeleton points do
  if weight is not zero then
    divide distance by weight
  end if
end for
for all skeleton points without radius do
  Interpolate neighbors
end for
  
```

The radii of the skeleton points are then iteratively smoothed (again excluding the endpoints) by averaging each skeleton point's radius with its two neighbors.

#### 4.3.4 Boundary Position

The position of the boundary is determined for every boundary vertex. For every vertex on the slice path (original vertices and newly created vertices), the boundary location on the 2D map for a vertex  $v_i$  is calculated based on the discrete 2D skeleton points preceding and following the vertex's location on the skeleton. The positions of these before and after points are  $p1 = (p1_x, p1_y)$  and  $p2 = (p2_x, p2_y)$ , and the radii for of the two points are  $r1$  and  $r2$ , respectively. The normalized distance from  $p1$  is represented by  $\alpha$ . The new mapped position  $p$  for each vertex on the slice path is calculated as follows:

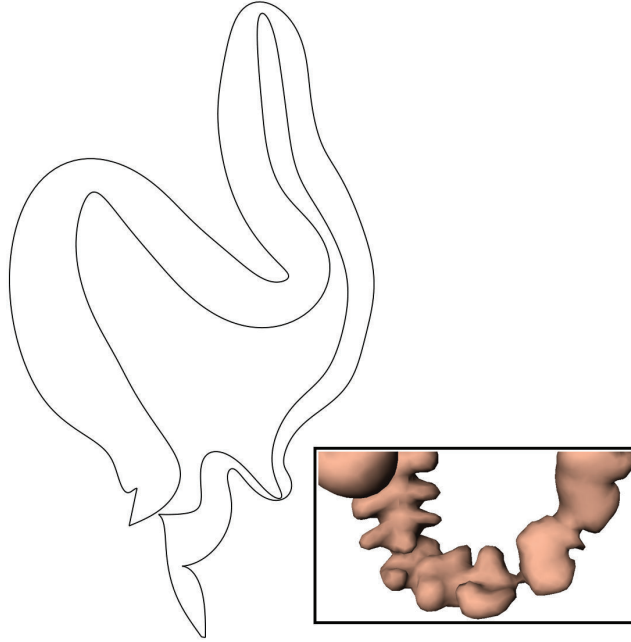
$$\begin{aligned}
 n &= p1 - p2, \\
 d &= \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} n_y \\ -n_x \end{bmatrix}, \\
 p0 &= p1 * (1 - \alpha) + p2 * \alpha, \\
 r &= r1 * (1 - \alpha) + r2 * \alpha, \\
 p &= \begin{cases} p0 + d * r, & \forall v_i \in M_0 \\ p0 - d * r, & \forall v_i \notin M_0 \end{cases},
 \end{aligned} \tag{4.4}$$

where  $M_0$  is the original unsliced mesh. The boundary points not on the slice path (from when the holes were punched) are set to equal intervals between the nearest sliced vertices. A smoothing operation is applied to remove intersections that might occur, while still preserving the overall general geometric structure. An example of the boundary for the mapping is shown in Figure 4.9.

## 4.4 Surface Mapping

Given the projected and spaced 2D skeleton of the structure and the boundary placement generated as the target domain  $S_2$ , the original 3D sliced colon surface  $S_1$  is mapped to the plane. Such mapping preserves the original projected curved information.

Discrete Ricci flow is used to map the open surface  $S_1$  to a rectangle  $D_1$  conformally,  $\phi_1 : S_1 \rightarrow D_1$ . Similarly, the target domain  $S_2$  is mapped to another rectangle  $D_2$  conformally,  $\phi_2 : S_2 \rightarrow D_2$ . Using harmonic mapping, a one-to-one and onto correspondence between  $D_1$  and  $D_2$ ,  $h : D_1 \rightarrow D_2$  is computed. Finally, the 2D correspondence is lifted to the 3D surfaces,  $f : S_1 \rightarrow S_2$ , and the original 3D surface is mapped to the designed target domain. The pipeline is shown in Equation 4.5.



**Figure 4.9:** Placement of the boundary in 2D for the colon in Figure 4.2. The narrow portion is caused by a narrowing in the original mesh (see inset).

The final result can then be rendered either using mesh rendering with vertex normal information stored from the original structure, or using volume rendering through the original volumetric dataset, which is the preferred method.

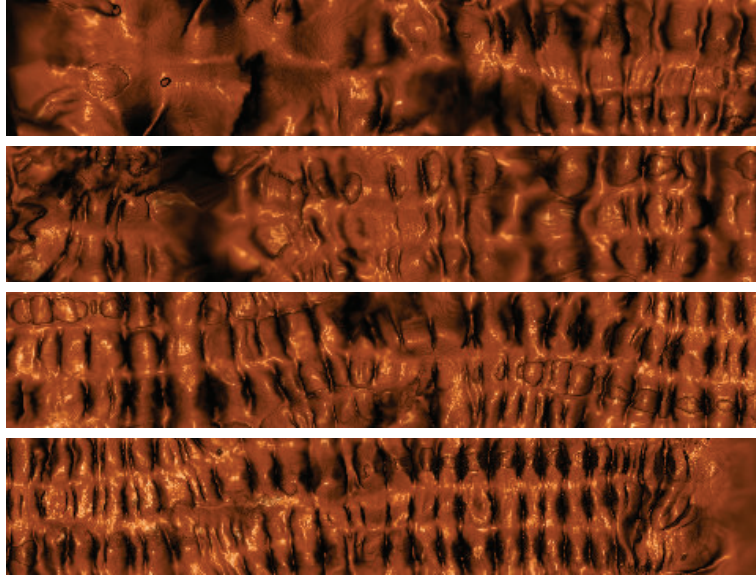
$$\begin{array}{ccc}
 S_1 & \xrightarrow{f} & S_2 \\
 \phi_1 \downarrow & & \downarrow \phi_2 \\
 D_1 & \xrightarrow{h} & D_2
 \end{array} \tag{4.5}$$

#### 4.4.1 Rectangular Mapping

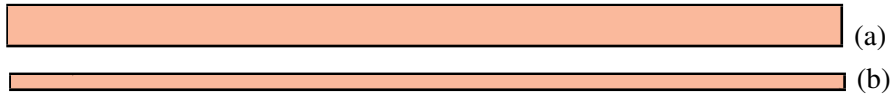
To map the 3D surface to the 2D plane, discrete Ricci flow is used. It behaves like a heat diffusion, deforming one Riemannian metric to another Riemannian metric conformally, as shown in Equation 4.6:

$$\frac{d\mathbf{g}}{dt} = -Kdt \tag{4.6}$$

where  $\mathbf{g}$  is the Riemannian metric and  $K$  is the target curvature.



**Figure 4.10:** Colon surface mapped to a rectangle. The four sections from top to bottom combine to form a single strip from left to right of the entire colon.



**Figure 4.11:** Example conformal modules (the ratios between height and width for the conformal mappings) for (a) the original surface (0.045796) and (b) the target domain (0.015378).

For the sliced colon surface, the boundary sides between each of the two corners are mapped to a straight segment, and the four corners are mapped to be the right corners of a rectangle. At the same time, the interior surface is mapped conformally to the plane. Therefore, for discrete Ricci flow, the target curvature of the four corners is set to be  $\frac{\pi}{2}$  and the curvatures of the other boundary vertices and interior vertices is set to be 0. The result of this is a rectangular mapping, as shown in Figure 4.10. Similarly, for the generated target domain, the corresponding corner vertices and boundary sides are computed, they are set to the same target curvature, and its conformal rectangular mapping is obtained. As a mesh is needed for this, the 2D boundary samples are used as constraints and a constrained Delaunay triangulation is performed to generate the triangular target domain [126].



#### 4.4.2 Deformation

The one-to-one and onto correspondence between two rectangles is computed by minimizing the harmonic energy. The resulting harmonic mapping obtains the smoothest deformation between two rectangles. In general, the original 3D colon surface  $S_1$  and the 2D target domain  $S_2$  are not conformally equivalent. As shown in Figure 4.11, their conformal modules (the ratios between height and width for the conformal mappings  $D_1$  and  $D_2$ ) are quite different.

From the computation of the target boundary placement, the correspondence of boundary vertices is known. The positions of the target boundary vertices are then used as the constraint condition in the harmonic mapping, as shown in the following equations:

$$\text{harmonic energy } E = \sum_{\text{edge}(i,j) \in D_2} \|h(v_i) - h(v_j)\|^2 \quad (4.7)$$

$$\text{harmonic mapping } \Delta h = 0, \quad h|_{\partial D_2} = \phi_1(S_1)|_{\partial D_1} \quad (4.8)$$

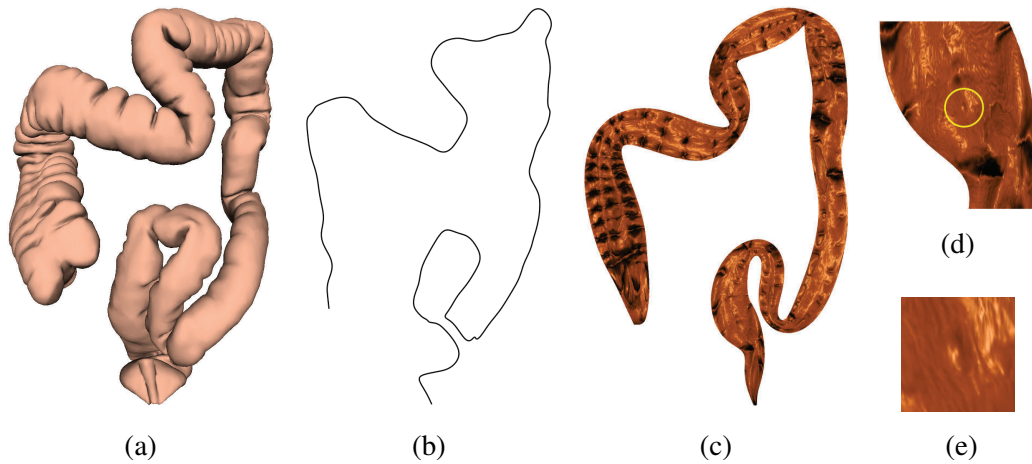
where  $v_i$  is the  $i$ th vertex and  $(i, j)$  is the edge of  $\langle v_i, v_j \rangle$  in the discrete triangular mesh  $D_2$ . The detailed computation can be found elsewhere [162].

The rectangle  $D_2$  of the target curved domain  $S_2$  is mapped to the rectangle  $D_1$  of the 3D sliced colon surface  $S_1$  to get  $D'_2$ . Now  $D'_2$  and  $D_1$  are well aligned. Then, the correspondence  $h : D_1 \rightarrow D_2$  is obtained. Therefore, the correspondence between 3D surfaces can be recovered by  $f = \phi_2^{-1}h \circ \phi_1 : S_1 \rightarrow S_2$ . The 3D colon surface is now mapped to the target boundary domain, as shown in Figure 4.2(c).

#### 4.4.3 Rendering

The flattened map can then be volume rendered similarly to how other flattened surfaces have been rendered [162]. Such volume rendered images are preferred over a mesh rendering. Given the flattened  $(u, v)$  coordinates for each vertex, these can be used to place the map in the 2D plane. Each vertex's original  $(x, y, z)$  coordinates are then used to reference the original location within the volume from which the surface was extracted.

For each vertex, a camera position is also needed; this position is taken to be the nearest skeleton point in relation to the flattened mesh. Since the boundary points are already associated with a skeleton point, the nearest boundary vertex is found for each inner vertex. The boundary vertex's corresponding skeleton point is then taken to be the camera position for that vertex. Using this information, volumetric ray casting is performed to generate the final images (see Figure 4.2(c)).



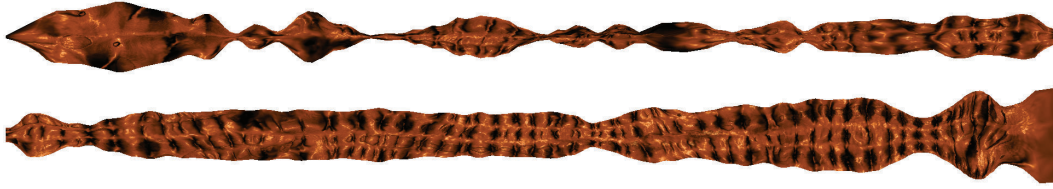
**Figure 4.12:** Results for a colon model. (a) The original colon surface. (b) The shaped skeleton. (c) The flattened and rendered colon map. (d) Zoomed region of the rectum with a small 4mm polyp circled in yellow. (e) Close-up of the polyp.

## 4.5 Results

The pipeline was implemented in C++ in a Windows environment. Matlab was used during the flattening for solving the linear system. The volumetric rendering was performed on the GPU using OpenGL and Cg. The entire pipeline takes approximately 2-3 minutes to run.

The colon data used in this work generally come from volumes with an approximate size of  $512 \times 512 \times 400$  voxels. Preprocessing of the volumes includes electronic cleansing, segmentation, triangular mesh extraction, and skeleton extraction. The meshes obtained are often very large (typically over 1.5 million faces) and are simplified to approximately 5% of their original size. The skeleton is discretized at a spacing of 1mm for use in this work. This method has been used successfully to map colon surfaces, which are notoriously twisty, to a plane in such a way that preserves the overall context.

In observing the results, the major bends of the colon are preserved, and an indication of the colon's radius along the skeleton is also present. In addition to the colon shown in Figure 4.2, a second mapping example is shown in Figure 4.12 with the original mesh, 2D shaped skeleton, and rendered result. Since volume rendering is used to create the final image, even small polyps are easily observable. The colon shown in Figure 4.12 has a small 4mm polyp near the rectum. A zoomed in version of this region, closer to the viewing level one would use if scanning the surface for polyps, is shown in Figure 4.12(d), and the polyp can be readily observed as a small oval bump on the surface. A further close-up of the polyp on the flat colon is shown



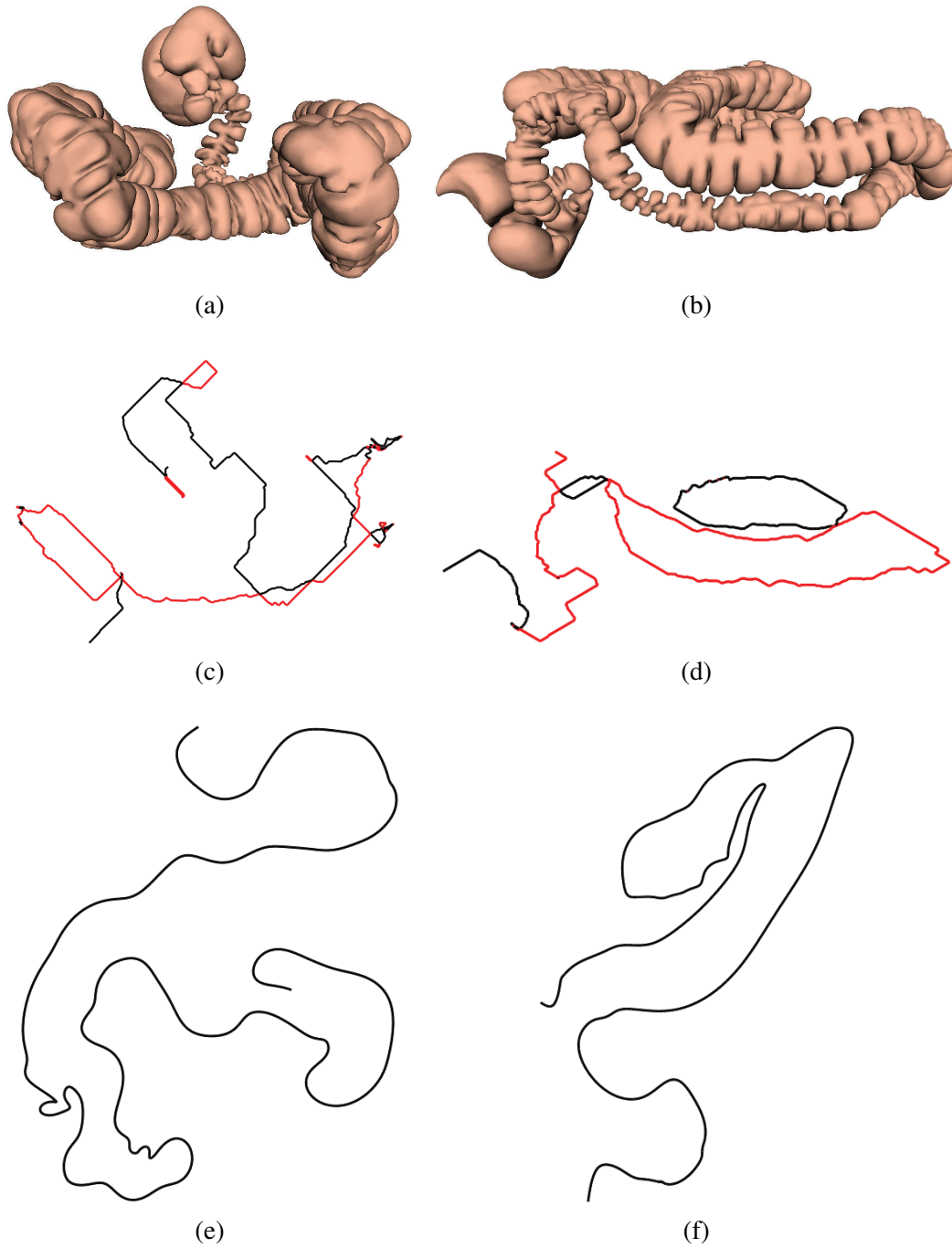
**Figure 4.13:** Straight map adjusted for radius of the colon in Figure 4.2. The top and bottom sections combine to form a single strip from left to right.

in Figure 4.12(e).

Compared with straightened mappings, these context preserving maps allow for a greater understanding of the general geometry of the structure. The colon in Figure 4.2 is mapped to a straight rectangle in Figure 4.10 and is also mapped a straightened representation accounting for radius in Figure 4.13. Compared to both, the general geometry of the colon is well preserved. Due to the several smoothing operations in creating the non-intersecting boundary, some of the radius information is lost compared to Figure 4.13, but the general thickness is preserved and gives a sense of the size.

In addition to better preserving the geometric structure, these maps can also make more intuitive use of a squarish region. This is readily observable when comparing the context preserving map from Figure 4.2(c) to the straight maps in Figures 4.10 and 4.13. The straightened versions require much greater contiguous space along a single axis if they are to be presented without breaks, while the context preserving map makes better use of a squarish block of space.

The selection of a good projection plane is important to the final result, and choosing views far away from the optimal view can lead to unusual results, as shown in Figure 4.14. For the colon, using the diagram view as the basis of the orthogonal projection supplies not only a typical view expected by the user, but also a view which will in general minimize the number of detrimental overlaps.



**Figure 4.14:** Examples of failures for bad viewpoints for the colon in Figure 4.2, viewed from (a, c, e) the top and (b, d, f) the side. Shown are (a, b) the original mesh, (c, d), the projected skeleton, and (e, f) the final untangled and spaced skeleton.

## Chapter 5

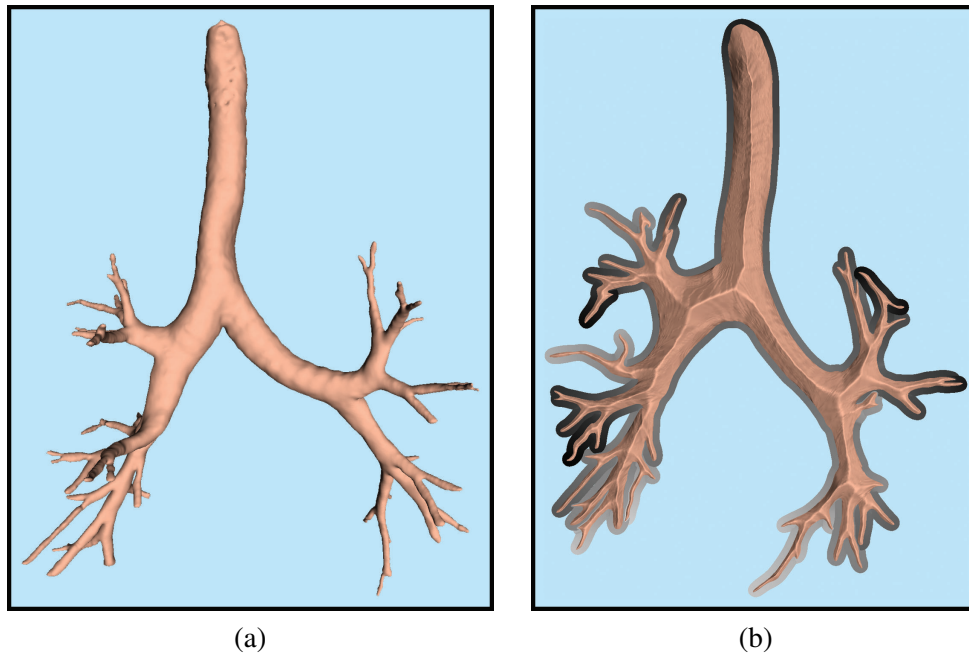
### Contextual Maps of Treelike Structures

Drawing on the work of the previous chapter, this chapter presents a method of creating context preserving parameterizations of treelike objects, such as the bronchi and blood vessels [92]. Though similar to the work in creating 2D context preserving maps of the colon, this technique addresses several new problems that must be overcome, such as the placement order of the branches in the 2D map and the ability to convey a sense of depth to the user. Due to this, this cannot be viewed as a straightforward extension of the previous work and is, in essence, distinct in its manner of method for achieving the final result.

#### 5.1 Motivation

Visualization of treelike objects is a common task for inspection of structures, such as blood vessels and airways. However, observing these objects, whether from an internal or external viewpoint, can be challenging due to occlusions and thus the need to rotate the view. It has been shown that visualizing a 3D object in 2D reduces complexity and makes tasks more efficient and accurate, though it has also been indicated that maintaining the sense of structure for the object is important [11]. This has motivated us to develop a method to create planar visualizations of such objects where there are no occlusions, but the overall geometric context of the object is preserved. This allows the user to succinctly investigate all parts of the object, while being able to easily identify the current region being investigated. The resulting objective of this work is shown in Figure 5.1 using a bronchial tree.

Treelike objects are common occurrences in medicine, and there is often great effort in extracting them due to their importance [80]. Some procedures look at the interior of such tubular structures (e.g., virtual bronchoscopy), while others will look at values mapped to the surface (e.g., simulations for vessel walls). For such applications, there are two use cases where planar visualizations would be advantageous. The first use case is as a guidance map, especially for virtual endoscopic



**Figure 5.1:** Planar visualization for bronchial tree. (a) The original mesh structure viewed from the direction of the projection plane. (b) The planar representation showing two visualization techniques for map use: raised central path (giving an indication of the medial axis) and depth cue border (giving an indication of the depth of the structure along the  $z$ -axis).

procedures. By presenting the entire object to the user in a single view without occlusions, but which still has the shape of the object, one's location within the object can be easily identified without fear of it being occluded, which would necessitate a rotation of the 3D map. The second use case is to utilize such a map for direct inspection. Having a planar version of the object allows for the user to easily observe the entire surface without missing any regions while still being able to understand one's location. A depth-first search of such a flattened object would be an appropriate guidance for inspection, and various views, such as volume rendering the interior of the organ or viewing a color map on the surface, can be provided.

Due to the importance of skeleton curves in computer graphics and visualization, there has been much work dedicated to methods of extracting these curves from objects. For extracting skeletons from volumetric medical data, methods have been proposed which connect centerlines based on a medial voxel axis [168] or which use a distance from boundary field with penalty weights [10]. For obtaining a skeleton directly from a mesh, methods have been presented based on surface con-

traction [63], through the use of advection to move mass from the object's surface to its skeleton [119], and through the use of least squares optimization [148]. For data from incomplete point clouds, curve skeletons can also be approximated based on a generalized rotational symmetry axis for a set of points [135].

The skeleton curve of an object compactly and efficiently represents the structure's general geometric shape, and such curves are employed in a wide variety of applications, including for navigation, animation, and analysis [19], for driving shape deformations [156], and even in graph visualization for edge bundling to yield cleaner views [27]. In the field of volume rendering, skeletons have been used for the creation of shape-based transfer functions [115] and to segment volumes into regions to which different transfer functions can be applied [155]. Virtual endoscopy applications often use paths based on extracted skeletons to automatically guide the user through a navigation of the object [8]. It is important that such paths are accurate and reliable to ensure that the entirety of the organ's internal surface is presented to the user [40]. These paths are especially useful in virtual bronchoscopy systems due to the large number of bifurcations through the airway tree, and extracting and utilizing these skeletons for path planning is an important task [64].

For visualizing graph data, there has been much work on various ways of performing both planar embeddings and 3D visualizations [43], and there is interest in methods to improve the layouts of planar graphs while maintaining planarity [128]. Specific to treelike structures, there has been interest in planar visualization of trees of hierarchical data [97] and contour trees [41]. Radial embedding of graphs is one popular area of graph drawing [5], and this work adapts a method of radial embedding for phylogenetic trees which takes into account a fixed edge length [6]. Unlike other works which are content only to find a pleasing embedding, this work seeks an embedding which is not only planar and with fixed edge length, but which also depicts as accurately as possible the actual shape of the object.

Surface parameterization is a popular topic for research in computer graphics [28, 127]. Conformal geometry for mapping triangular meshes, where local angles are preserved, has been used in the creation of texture maps [130]. Harmonic maps are one method to map between two topological disks, and such maps are conformal under certain circumstances [60]. Discrete Ricci flow is a recent method of computing conformal maps of structures [59], and it has been used to obtain optimal surface parameterizations using inverse curvature maps [158]. However, Ricci flow requires a target Gaussian curvature for each vertex and is thus difficult to control when utilizing a fixed boundary.

Applying mapping techniques for flattening objects to the plane is common in computer graphics applications, but somewhat limited in medical applications of hollow structures. Such unfolding tends to be limited to simple organs such as the

colon [7, 47] and bladder [76]. Similar techniques have been applied in a limited sense to flatten vessels in an angle preserving manner, though a straightening is performed, thus losing the geometric context [169]. In this work, complex objects are flattened to the plane in a way that preserves the overall global shape of the object. Since there is a predefined boundary to which the surface must be mapped, harmonic mapping is used for this task.

In work on coronary artery visualization, users have indicated that accurate anatomical information is important [11]. While that work created tree diagrams to portray this anatomical information, a singular connected surface representing the true shape of the arteries was not provided. In other work which dealt with the idea of straightening objects for comparison, and thus losing the geometric context, it was noted that such views were foreign to physicians, who were accustomed to the original shapes of the objects [3].

Previous work specifically on context preserving flattening of objects has been limited to simple tubular structures without branches [96]. The method used therein is not robust to multi-branched objects. Based on first embedding the skeleton and then removing intersections through a series of reflections which distort the shape, that method can even lead to undesirable results. Presented here is a new method to create the skeleton layout in 2D that first finds a guaranteed planar embedding of the skeleton tree structure, and then adjusts this embedding to recover the original shape. With the shape recovery last, it cannot become distorted due to intersection removal. Rather than using a two-step flattening procedure with Ricci flow and harmonic mapping, a single harmonic mapping step is used which operates on the full high resolution mesh dataset and not on a reduced version. The mapping procedure also takes into account where the mesh is being distorted too much by spreading outside the boundary and adjust the boundary to account for this. This work also explores problems specific to more complicated multi-branched structures, such as the order of placement for branches in the embedding and the proper slicing of the boundary for unfolding, and discusses providing depth cues for the user to assist in understanding the more complicated flattened structures.

This chapter presents a method for creating planar maps of treelike objects without occlusions which maintain the general geometric context of the original structure. These maps are created by first finding a planar layout, without intersections, of the object's skeleton, and then using this to guide the flattening of the object's surface. In some sense, the problem of finding a 2D layout of an object's skeleton with no intersections can be considered as a planar graph embedding problem. Since this work is on treelike objects, and thus there are no cycles in the skeleton, it is guaranteed that a planar embedding exists. Introduced is a modified radial embedding algorithm which takes into account the required order in which branches are to be embedded to allow for the final shape reconstruction to be achieved. Using



this planar embedding, an optimization procedure is developed that reconstructs the original shape as much as possible without creating any intersections.

Given a target embedded skeleton to use as the basis for flattening the object's surface, it is necessary to create a boundary along the object by slicing it open in a manner which is consistent with the order of the embedded branches. Given this surface, which is now topologically equivalent to the plane, harmonic mapping is used to flatten it to the plane. To account for regions where the surface is stretched across gaps near bifurcations, a boundary adjustment procedure is introduced to better fit the data. Using this flattened surface, various modes of visualization can be provided using it as a map for guidance or for direct inspection, and also presented is a grayscale border which gives depth cues to the user, allowing for one to identify which areas are further from the background.

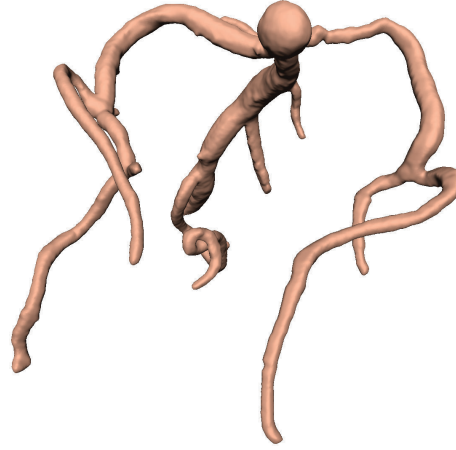
The following new ideas are introduced in this work:

- A method to flatten a treelike structure to the plane such that the general shape of the structure is preserved.
- An algorithm to create a planar embedding of tree graphs which creates an embedding with ordered importance for branches.
- A parameterization method with a movable boundary.
- Depth cue borders to the parameterization to aid in rapid recognition of object depth.

## 5.2 Algorithm

The procedure for creating the planar map of a treelike object is based upon using the object's skeleton to preserve the shape. Using a 2D version of this skeleton and the estimated radius along the skeleton, a radial planar embedding is produced which guarantees no intersections of the 2D skeleton. This embedded skeleton is then deformed such that the angular curvature resembles that of the original projected skeleton. Using this shape restored skeleton, the object surface is sliced open such that the boundaries are consistent for the planar mapping and the flattening is then performed using harmonic mapping.

The skeleton used for this is arranged in a tree form where each node has exactly one parent (except for the root) and any number of children. Since branches along the skeleton typically do not occur at exactly the same point, most nodes have no more than two children. A node at the end of a branch, with no children, is a *leaf* node, a node with more than one child, which therefore causes the split of one or more branches, is termed a *branch* node, and all other nodes, with exactly one child,



**Figure 5.2:** Surface mesh of cranial blood vessels.

are *inner* nodes. Leaf and branch nodes can be collectively referred to as the group of *significant* nodes, and these nodes indicate the general structure of the skeleton. A region of inner nodes between two significant nodes is referred to as a *segment* of the skeleton.

The process of the algorithm is illustrated using the cranial blood vessels in Figure 5.2. Throughout the description, the following terms are used to describe the skeleton and its associated values. The skeleton  $S = (N, E)$  contains the set of  $n \in N$  nodes and  $e \in E$  edges connecting the nodes. Unlike conventional graph theory terminology, an edge  $e_n$  is defined as being the edge which goes from node  $n$  to its parent,  $parent(n)$ . Therefore, the size of  $E$  is one less than the size of  $N$  since the root node  $root(S)$  has no parent and thus no edge. The estimated radius value of a node  $n$  is represented by  $\rho_n$ .

### 5.2.1 Orthographic Projection

The original 3D skeleton of the structure is first projected to a plane. To minimize distortion, this projection is accomplished using the standard orthographic projection matrix:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (5.1)$$

where  $u$ ,  $v$ , and  $w$  are the right, up, and normal vectors of the projection plane, respectively. For each projected point  $(x', y', z')$ ,  $(x', y')$  are used to create the  $uv$  coordinates for the skeleton and  $z'$  is retained for use later in indicating the depth of the object along the view direction. The 2D projected skeleton is shown in Figure 5.3(a).

Based on this 2D projection, an angle  $\theta$  is calculated for each node which is used in reconstructing the shape. For a node  $n$ ,  $\theta_n$  is computed as the angle between  $e_n$  and  $e_{parent(n)}$ . If the 2D edge length of  $e_n$  is zero, then  $\theta_n = \pi$ . If the 2D edge length of  $e_{parent(n)}$  is zero, then continue up the skeleton until a node  $p$  is reached such that the 2D edge length of  $e_p$  is nonzero. If at any node  $parent(n) = root(S)$ , then  $e_{parent(n)}$  is given as the edge between  $root(S)$  and  $root(S) + (1, 0)$ . The angles are stored as the counterclockwise rotation in the range of  $[0, 2\pi)$ .

### 5.2.2 Radial Layout

Once the 2D projection of the skeleton is obtained, the next step is to perform a planar embedding of the skeleton. The primary goal of this embedding is that there be no intersections of the skeleton, which would then become occlusions in the final surface mapping. To achieve this goal of no intersections, a radial planar embedding is first found which guarantees that there are no intersections. This planar embedding loses the shape information, but ensures that there are no intersections of the skeleton with itself. Once this radial layout is achieved, the next step reconstructs the shape based on the angles previously calculated.

When performing the radial layout, it is important that branches are placed in the proper order for the final shape, since branches that are swapped cannot be swapped back in the shape optimization step that comes later. For this, a *placement weight* is calculated for the children of every node. This weight determines which child subtree should be placed first. Since the radial layout is performed in a counterclockwise direction, branches which should be placed left of their siblings must be embedded first and are given a lower weight value. This weight is calculated as the angle between the parent node's edge and the weighted center of the projected positions of nodes in the subtree (up to a maximum distance of the minimum of the lengths of all subtrees of the parent node). The procedure for calculating these weights is given in Algorithm 5.1.

The radial layout itself is based on previous work used in embedding phylogenetic trees, where specified edge lengths are known [6]. This is similar to the situation here, where the distance between two skeleton nodes must be preserved, and thus fixed edge lengths are required. The method is modified to account for the order in which to place subtrees based on the desired location from the projected skeleton using the calculated placement weights. The modified procedure for the radial planar embedding using these weights is given in Algorithm 5.2. Note that the edge lengths  $\varepsilon$  used in this step are the original 3D edge lengths, allowing for the full length of the object to be portrayed. Otherwise, edges closely aligned with the view direction of the projection plane would be greatly reduced. The radially embedded skeleton is shown in Figure 5.3(b).

---

**Algorithm 5.1:** Calculation of Placement Weights

---

**Input:** Skeleton  $S = (N, E)$  with node arrays  $\varepsilon$  (2D edge lengths) and  $x$  (projected 2D coordinates)

**Result:** Placement weights  $\alpha_n$  for all  $n \in N | n \neq \text{root}(s)$

**begin**

**foreach**  $n \in N$  **do**  
    └ CalculateChildWeights( $n$ )

**end**

**function** CalculateChildWeights(*Node*  $n$ )

**foreach**  $c \in \text{children}(n)$  **do**  
    └  $\delta_c \leftarrow \text{SubtreeDistance}(c)$   
   $d \leftarrow \min(\delta_c | c \in \text{children}(n))$   
  **foreach**  $c \in \text{children}(n)$  **do**  
    └  $m_c \leftarrow \text{WeightedCenter}(c, d)$   
    └  $\alpha_c \leftarrow \text{Angle}(m_c, n)$        $\triangleright$  Angle of  $m_c$  w.r.t  $e_n$

**function** SubtreeDistance(*Node*  $n$ )

**foreach**  $c \in \text{children}(n)$  **do**  
    └  $d_c \leftarrow \text{SubtreeDistance}(c)$   
  **return**  $\max(d_c | c \in \text{children}(n)) + \varepsilon_n$

**function** WeightedCenter(*Node*  $n$ , *Float*  $\text{maxDist}$ )

  Create queue  $q$  of pair  $\langle \text{Node}, \text{Float} \rangle$   
  Push pair  $\langle n, \varepsilon_n \rangle$  into  $q$   
   $s \leftarrow (0, 0)$   
   $sw \leftarrow 0$   
  **while**  $q \neq \emptyset$  **do**  
    pair  $\langle cn, cd \rangle \leftarrow$  popped front node of  $q$   
    **if**  $cd \leq \text{maxDist}$  **then**  
       $w \leftarrow 1 - (cd / \text{maxDist})$   
       $s \leftarrow s + w \cdot x_{cn}$   
       $sw \leftarrow sw + w$   
      **foreach**  $c \in \text{children}(cn)$  **do**  
        └ Push pair  $\langle c, cd + \varepsilon_c \rangle$  into  $q$   
  **return**  $s / sw$

---

### 5.2.3 Shape Recovery

Given the radial layout of the skeleton, a 2D embedding is now achieved which is guaranteed to have no intersections. However, the shape of the object has been lost as each branch has been straightened. To restore the shape, each node is bent around its parent, in a breadth-first manner, using the original  $\theta$  angles calculated based on the 2D projection. For this step, the radius values are also used to ensure

---

**Algorithm 5.2:** Radial Layout Using Placement Weights

---

**Input:** Skeleton  $S = (N, E)$  with node arrays  $\epsilon$  (3D edge lengths),  $\theta$  (angles),  $\alpha$  (placement weights) and  $l$  (number of leaves in subtree)

**Result:** Embedded 2D coordinates  $x_n$  for all  $n \in N$

**begin**

```
 $x_{root(S)} \leftarrow (0, 0)$   
 $\omega_{root(S)} \leftarrow 2\pi$   
 $\tau_{root(S)} \leftarrow 0$   
EmbedNode( $root(S)$ )
```

**end**

**function** EmbedNode(Node  $n$ )

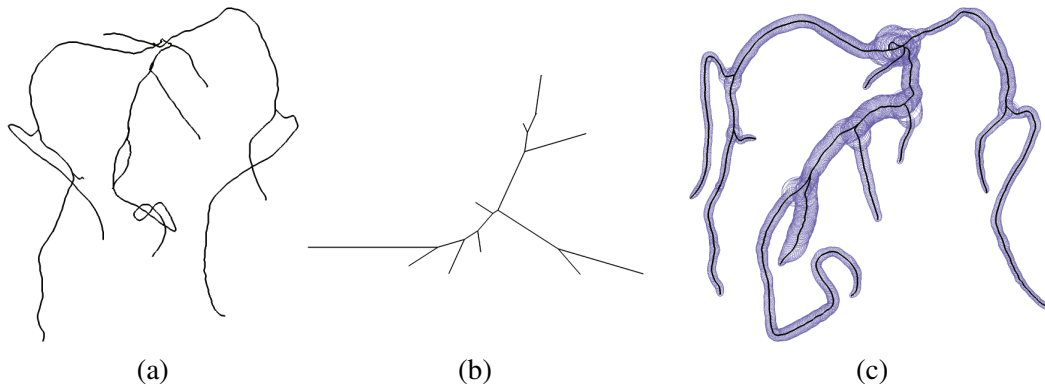
```
if  $n \neq root(S)$  then  
   $x_n \leftarrow x_{parent(n)} + \epsilon_n(\cos(\tau_n + 0.5\omega_n), \sin(\tau_n + 0.5\omega_n))$   
   $\eta \leftarrow \tau_n$   
  forall the children( $n$ ) do  
     $c \leftarrow$  unembedded child with lowest  $\alpha_c$   
     $\omega_c \leftarrow 2\pi(l_c/l_{root(S)})$   
     $\tau_c \leftarrow \eta$   
     $\eta \leftarrow \eta + \omega_c$   
    EmbedNode( $c$ )
```

---

that there is space for the surface to be mapped. The radius values are increased slightly for this step in order to leave space around where the final mapped surface will be placed.

As each node  $n$  is visited, the node is rotated based on its current angle and its target angle  $\theta_n$ . When rotating a node, all of the nodes in its subtree are also rotated accordingly. Given this rotated subtree, it is checked to see if there are any intersections of the skeleton or intersections of circles centered at each node with that node's radius value. If any of these intersections are found, then the rotation is undone along the subtree, the target rotation for  $n$  is slightly reduced, and if it has a single child  $c$ , then  $\theta_c$  is modified by the small amount of the reduction. This process is continued until the maximum amount of possible individual node bending in an iteration is below a specified threshold. A threshold of 0.01 has been found to be sufficient.

When finding intersections of the radius circles, it is apparent that nearby nodes always have intersections. Due to this, for each node, the set of *nearby* nodes is calculated. For a node  $n$ , another node  $m$  is nearby if the walking distance along the skeleton between them is less than or equal to  $\pi(\rho_n + \rho_m)$ , except in the case of leaf nodes. The scaling factor of  $\pi$  allows relatively nearby positions to become merged in the final flattening the better preserve the shape. For leaf nodes, the distance must



**Figure 5.3:** Embedding of the 2D projected skeleton of the cranial blood vessels in Figure 5.2 to the plane. (a) Original 2D projected skeleton. (b) Radial planar embedding layout of the skeleton. (c) Shape restored embedded skeleton with estimated radii along the skeleton indicated by purple circles to show the space for surface mapping.

be less than or equal to  $(\rho_n + \rho_m)$ , or else the tip of the skeleton may come too near to another section. The procedure for this shape recovery step is given in Algorithm 5.3. The result of this process is shown in Figure 5.3(c) with circles illustrating the node radii.

#### 5.2.4 Boundary Creation

In order to map the surface of the object to the plane, it is necessary to slice the mesh surface open to create a boundary, which makes the surface topologically equivalent to the plane. To accomplish this, for each significant node in the skeleton (leaf nodes and branch nodes), a corresponding mesh vertex on the surface is found. For leaf vertices, this is simply the nearest vertex. For branch vertices, the cross product of its children is found as a direction vector, and the negation of it is taken if it lies more in the direction of the normal of the projection plane. A ray from the branch node with this direction is then defined, and the nearest vertex in the nearest face which this ray intersects is taken as the corresponding vertex.

The surface must then be sliced open between these points, with a slice for every segment of the skeleton. In order to allow the surface to be unfolded properly without twisting of the boundaries (see Figure 5.4), the slicing must be accomplished in a specific order. The slicing is performed top down, starting at the root. For the current significant node, paths are sliced between its corresponding mesh vertex and those of its children. To be in the correct order, the children are processed from right to left using the angles in the embedded skeleton (that is, processing from the child with the largest angle to that with the smallest). For the child being processed,

---

**Algorithm 5.3:** Shape Recovery for Skeleton

---

**Input:** Skeleton  $S = (N, E)$  with  $\alpha$  and node arrays  $\theta$  (angles),  $\rho$  (radii), and  $x$  (2D embedded coordinates)

**Result:** Updated 2D coordinates  $x_n$  for all  $n \in N$

**begin**

```
repeat
   $\beta \leftarrow \text{BendNodes}()$ 
until  $\beta < 0.01$ 
```

**end**

**function** BendNodes

```
 $\beta \leftarrow 0$ 
Create queue  $q$  of  $\langle \text{Node} \rangle$ 
foreach  $c \in \text{children}(\text{root}(S))$  do
   $\_ \text{Push} \langle c \rangle$  into  $q$ 
while  $q \neq \emptyset$  do
   $n \leftarrow$  popped front node of  $q$ 
   $\phi \leftarrow \text{CurrentAngle}(n)$   $\triangleright$  Calculates current angle
   $\gamma \leftarrow \alpha(\theta_n - \phi)$ 
  RotateSubtree( $n, \text{parent}(n), \gamma$ )
  if FindIntersections( $n$ ) then
    RotateSubtree( $n, \text{parent}(n), -\gamma$ )
     $\theta_n \leftarrow \theta_n - \gamma$ 
    if  $|\text{children}(n)| = 1$  then
       $\_ \theta_{\text{child}(n)} \leftarrow \theta_{\text{child}(n)} + \gamma$ 
  if  $|\gamma| > \beta$  then  $\beta \leftarrow |\gamma|$ 
  foreach  $c \in \text{children}(n)$  do
     $\_ \text{Push} \langle c \rangle$  into  $q$ 
return  $\beta$ 
```

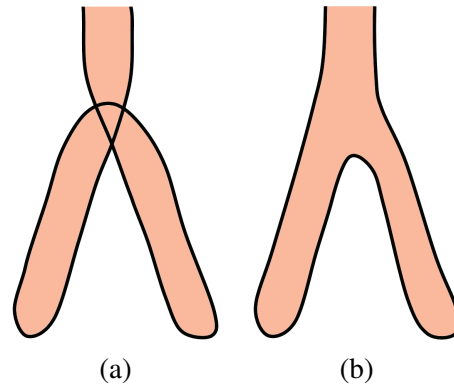
**function** RotateSubtree(Node  $n$ , Node  $p$ , Radians  $\gamma$ )

```
 $x_n \leftarrow \text{Rotate}(x_n, x_p, \gamma)$   $\triangleright$  Rotates  $x_n$  around  $x_p$  by  $\gamma$ 
foreach  $c \in \text{children}(n)$  do
   $\_ \text{RotateSubtree}(c, p, \gamma)$ 
```

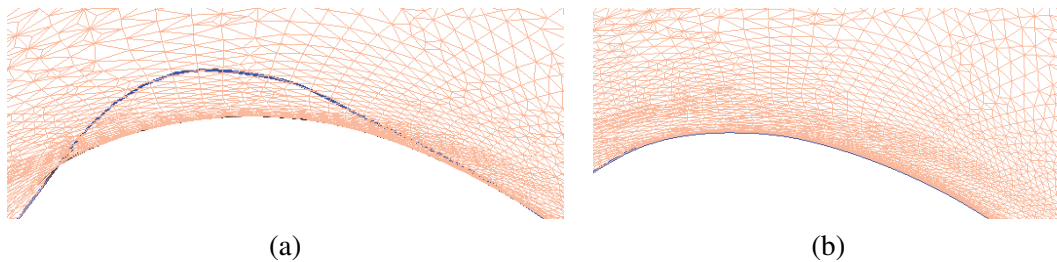
**function** FindIntersections(Node  $n$ )

```
foreach  $e \in E \mid e \neq e_n$  do
   $\_ \text{if}$  line segments  $e$  and  $e_n$  intersect then return true
foreach  $m \in N \mid m \notin \text{nearby}(n)$  do
   $\_ \text{if}$  circles  $(x_n, \rho_n)$  and  $(x_m, \rho_m)$  intersect then return true
foreach  $c \in \text{children}(n)$  do
   $\_ \text{if}$  FindIntersections( $c$ ) then return true
return false
```

---



**Figure 5.4:** Diagram showing the importance of slicing the mesh consistent with the skeleton position. (a) A bad slice, where boundary edges cross over each other. (b) A good slice with consistent boundaries yielding a proper view.



**Figure 5.5:** The use of boundary adjustment (boundary shown in dark blue). (a) Without boundary adjustment, a large portion of the mesh is stretched outside the boundary. (b) With boundary adjustment, the mesh is properly contained and fits better.

the shortest path along the mesh surface from its vertex to the parent vertex is found. This path is then sliced along, creating new vertices in the mesh. The vertices on the right side of the slice are the newly created ones, while the vertices on the left remain as the old ones. This is important, so that when the shortest path is found for the next child (which is left of the current child), it is be traced back to the original vertex point, which is on the left of the slice boundary.

### 5.2.5 Surface Mapping

In mapping the surface of the object to the plane, the planar positions of all boundary vertices in the mesh must be found. Every boundary vertex is first associated with the nearest skeleton node that corresponds to that segment of the skeleton. Since the boundaries were created in a consistent manner, boundary vertices from the original mesh can be placed at  $90^\circ$  counterclockwise from the skeleton node, while the newly created vertices are placed at  $90^\circ$  clockwise. A smoothing oper-



ation removes overlaps when multiple boundary vertices are associated with the same embedded skeleton node.

Given these initial planar boundary positions, there are often looped intersections along the boundary in narrow regions. To remove these loops, which also improves the area for which a large amount of the mesh must be mapped to, localized smoothing operations are performed within the loop and its nearby neighbors. Since the smoothing is restricted only to the looped section and its close neighbors, this does not affect the general shape of the flattened object.

Harmonic mapping is used to map the interior vertices to the plane based on the fixed boundary. The edge weights that are used for this are the Mean Value Coordinates which are guaranteed to be positive [29]. For a vertex  $v_i$ , its edge weight  $k_{ij}$  with respect to vertex  $v_j$  is given as follows:

$$k_{ij} = \frac{w_{ij}}{\sum_{v_k \in N(v_i)} w_{ik}}, \quad w_{ij} = \frac{\tan(\frac{\alpha_{ija}}{2}) + \tan(\frac{\alpha_{ijb}}{2})}{\|v_i - v_j\|} \quad (5.2)$$

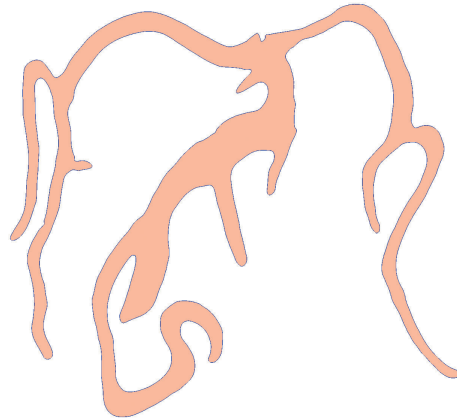
where  $N(v_i)$  is the set of all neighboring vertices to  $v_i$  and  $\alpha_{ija}$  and  $\alpha_{ijb}$  are the two angles at  $v_i$  formed by  $e_{ij}$ . Note that the edge weights are not symmetric, and thus  $k_{ij} \neq k_{ji}$ . Since the boundary is fixed during the parameterization, weights are calculated only for interior edges.

The interior vertices are initially flattened to the plane by setting their positions equal to that of the boundary vertex to which they are nearest. Then the harmonic energy is optimized, where the total harmonic energy  $H$  is given as:

$$H = \sum_{[i,j] \in M} |h(v_i) - h(v_j)|^2 \quad (5.3)$$

For this, the goal is to reach the state such that  $\Delta h_v = 0$ , where  $v \notin \partial M$ . The harmonic function is updated until the total distance moved by vertices between iterations is less than a specified threshold (a threshold of 0.1 has been found to be sufficient).

Due to the unfolding, some regions of the mesh can become stretched over a gap near a bifurcation. This causes the triangles in the mesh to go outside of the desired boundary and some to invert. To account for this, the affected boundary vertices are adjusted as necessary where this inversion is occurring. This is accomplished by translating the inverted vertex  $v$  orthogonal to the mesh boundary at that position by a value of one sixteenth the sum of the edge lengths of its two neighbor boundary edges. A local smoothing operation is then applied to all translated boundary vertices. Periodically, the entire boundary is checked for loops which may occur from this, and they are duly removed. The final result of this flattening is shown in Figure 5.6.



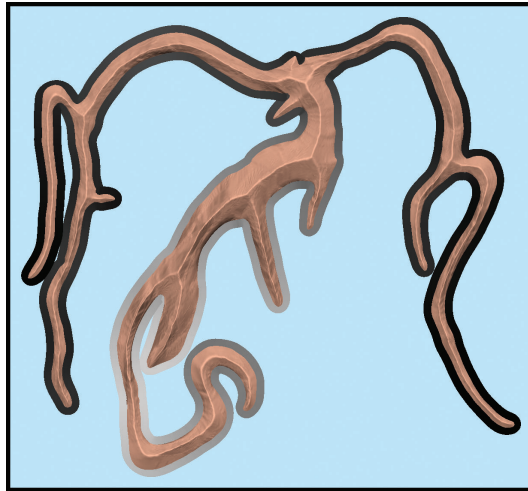
**Figure 5.6:** Flattened surface of cranial blood vessels.

## 5.3 Visualization

Given a planar representation of an object's surface, there are several ways in which it can be visualized, depending on need. These include being used as a guidance map or for direct investigation using volume rendering of the interior of the object or showing color maps on the surface. An indication of the spatial relation in the  $z$ -direction of the view can also be provided by depth cues around the border of the object. For direct inspection of the object, a depth first traversal can be used to ensure complete coverage.

### 5.3.1 Map Surface View

One use of the planar map is as an actual map to use as guidance during a virtual endoscopy. It is common for the physician to need to note their position within the structure, and a 3D map is often used for this task. However, since a 3D map may contain occlusions, the current position might be hidden, requiring the user to disengage from their current task to rotate the view and identify their location. Using a planar map eliminates this problem. For such a use, a view similar to an exterior view of the normal surface mesh would be appreciated. Since the mesh has been completely flattened to the plane, it would look completely flat when rendered (see Figure 5.6). To give the mesh a more natural appearance, height can be added back to the mesh's  $z$ -coordinates. This is done by finding the shortest distance from each interior vertex to the boundary. This distance is then set to the  $z$  coordinate of the vertex, raising it out of the plane. Now when rendered with lighting, a peaked appearance is given rather than a simple flat appearance (see Figure 5.7). Although not smooth and rounded, this is actually advantageous, as the peak through the

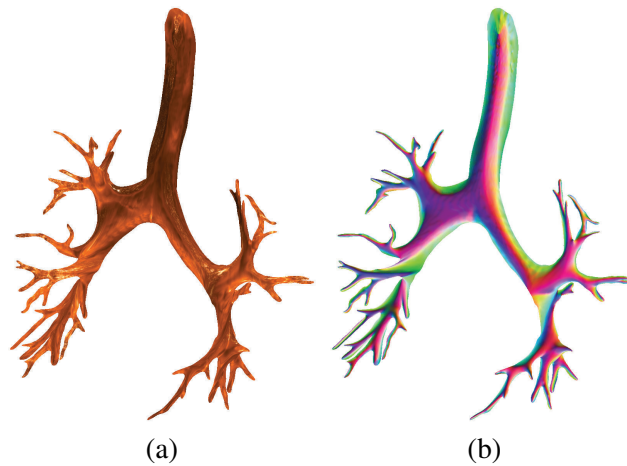


**Figure 5.7:** Flattened map of the cranial vessels with raised central path and depth cue as grayscale border.

structure forms a central path through the object which can serve as a visual cue to the approximate location of the skeleton within.

### 5.3.2 Depth Cues

An indication of the  $z$  depth of the mapped object can also be provided, and is a useful tool for more complicated branching structures. A border is first created around the boundary of the object. Based on the  $z$  coordinates of the projected skeleton, the corresponding vertices in the boundary can be colored in grayscale with black for vertices closer to the viewer and white for those further away. In this sense, the areas with the black border pop more out of the background, while the others fade into it. For this, a light blue color is used for the background, or else the whitish sections of the border would completely disappear. An example of this is shown in combination with the map view in Figure 5.7. When regions of the border may overlap the mesh surface in tight areas, the mesh surface is always given priority in being viewed. When the border may overlap itself in some areas, the darker border (indicating closer to the viewer) is always shown on top of the lighter border. A very small border is always maintained around the surface boundary with the corresponding correct depth indication which is not obscured. The bronchial tree in Figure 5.1 contains examples of these overlapping borders and the resulting view. The depth border can be turned on or off as desired by the user, and the width of the border can also be adjusted as needed. Note that for the images presented here, the border is kept substantially large so as to appear clearly in the images.



**Figure 5.8:** Examples of visualization for direct inspection. (a) Volume rendering. (b) Color map of surface normals.

### 5.3.3 Direct Inspection

For virtual endoscopy applications, it is possible to perform volume rendering of the interior surface on the flattened map, allowing it to be inspected directly in an efficient manner. To perform this volumetric ray casting, the original  $xyz$  coordinates of the mesh are used to calculate the starting locations of the ray for each vertex within the volume's local coordinate system. The vertex's normal value can also be used to define the ray direction. This normal is slightly smoothed with those of the neighboring vertices to account for the discrete nature of the triangular mesh. With these values set at every vertex, the volume positions and ray directions are then interpolated over each triangle in the mesh and used for volume rendering. An example of this is shown in Figure 5.8(a).

For other applications, it might be necessary to visualize a color map on the object's surface which represents data calculated from or simulated on the mesh. Since the full resolution mesh was mapped to the plane, and not a downsampled version, the color map can be rendered at full quality on the flattened surface without loss of data resolution. An example of this is shown in Figure 5.8(b).

## 5.4 Results

The inputs to this algorithm are the original triangular mesh, extracted skeleton, and estimated radii for each node along the skeleton. If these radius values are not known, they can be estimated from the skeleton curve and triangular mesh. For each skeleton point in 3D, rays are cast around it and the closest intersections with the

**Table 5.1:** Running times (in minutes) for the illustrated datasets. The Skeleton Time covers the orthographic projection, radial layout, and shape recovery. The Mesh Time covers the boundary creation and surface mapping.

<i>Dataset</i>	<i>Skeleton Nodes</i>	<i>Mesh Vertices</i>	<i>Skeleton Time</i>	<i>Mesh Time</i>
Bronchial Tree I	1,948	92,047	1.83 min.	1.25 min.
Bronchial Tree II	1,031	59,230	0.26 min.	0.99 min.
Cranial Blood Vessels	1,763	134,589	3.58 min.	1.25 min.
Lower Limb Vessels	1,531	93,875	0.59 min.	0.53 min.
Aorta	2,670	331,200	2.02 min.	25.3 min.

triangular mesh are found and averaged to use as the radius. Using steps of fifteen degree increments for  $360^\circ$  around the point is sufficient. Though a technical unfolding of the surface should provide a distance of one half of the circumference on each side instead of the radius, this significantly larger value would cause problems with greater amounts of regions intersecting each other. Using the radius shows the item as narrower in relation to its length, but the width along the object is relationally consistent along the entire skeleton. In some senses, this is actually better for guidance maps, as the view remains similar to that of the original 3D mesh view. Figure 5.9 shows this technique applied to the lower limb vessels and the similarity to the original mesh, albeit without occlusions.

The projection plane used in the orthographic projection is chosen as a typical view for the object. For most types of objects, especially in medical applications, a canonical view is known which provides the best view and is familiar to the user. If such a view is not known a priori, a plane can be chosen at the discretion of the user, such as one which minimizes structural overlap and skeletal intersections.

Due to the boundary adjustment step in the harmonic mapping, inverted vertices are virtually eliminated. Only an extremely small amount remain, which do not affect the final result (less than 0.05% remain inverted). Due to the fixed boundary, a parameterization that is fully shape preserving (i.e., conformal) is not possible, but the resulting mapping is quasi-conformal (i.e., circles are mapped to ellipses).

The running time for this technique is shown in Table 5.1 (in minutes) for the five datasets illustrated herein. Also given for each dataset are the number of discrete nodes in the skeleton that was processed and the number of vertices in the final flattened mesh (that is, after slicing). Since it is imperative to ensure that the boundary adjustment grows slowly and thus does not greatly change the shape, Newton's method is not used to accelerate the harmonic mapping. However, as shown, the running times are generally sufficient on a modern processor. These times were acquired using an Intel Core i7-3930K processor.

**Table 5.2:** Average angular distortions (in radians).

<i>Dataset</i>	<i>Avg. Angular Distortion</i>
Bronchial Tree I	0.48
Bronchial Tree II	0.46
Cranial Blood Vessels	0.48
Lower Limb Vessels	0.47
Aorta	0.49

Though global shape recovery is the goal, this is not always best when the object would be used for direct inspection. As shown in the close-up Figure 5.10(a), some regions may become sharply turned. There are no actual overlaps here (see Figure 5.10(c), which shows the border only of the mesh), though such a large twisting of the surface can be undesirable. This effect can be mitigated in some instances by using a different placement weight based only on the location of the first child node instead of the entire subtree (see Figure 5.10(b)).

This method has been tested on real world datasets of common, naturally occurring, biological structures which are frequently encountered in real world situations. One can dream of further situations where much more complex structures require such a mapping. Since the planar embedding is based on a radial embedding which is guaranteed to exist, any structure can be mapped. However, extremely large structures might need to make concessions in either the amount of shape recovery possible, the size of resulting mapping, or both.

## 5.5 Evaluation

The average angular distortion on the parameterized surface was measured. The angle distortion  $\varepsilon_{f,\theta}$  for a face  $f$  is given as:

$$\varepsilon_{f,\theta} = |\alpha_{f,p} - \alpha_{f,o}| + |\beta_{f,p} - \beta_{f,o}| + |\gamma_{f,p} - \gamma_{f,o}| \quad (5.4)$$

where  $(\alpha_{f,p}, \beta_{f,p}, \gamma_{f,p})$  are the three interior angles in the parameterized face  $f$  and  $(\alpha_{f,o}, \beta_{f,o}, \gamma_{f,o})$  are the original interior angles. The average angular distortion  $\bar{\varepsilon}_\theta$  is given as:

$$\bar{\varepsilon}_\theta = \frac{1}{3|f \in M|} \sum_{f \in M} \varepsilon_{f,\theta} \quad (5.5)$$

for a mesh  $M$ . The average and error values for the meshes are given in Table 5.2. Due to the requirement of a fixed boundary, the result is a parameterization that is quasi-conformal. This can be seen when using the parameterization for texture

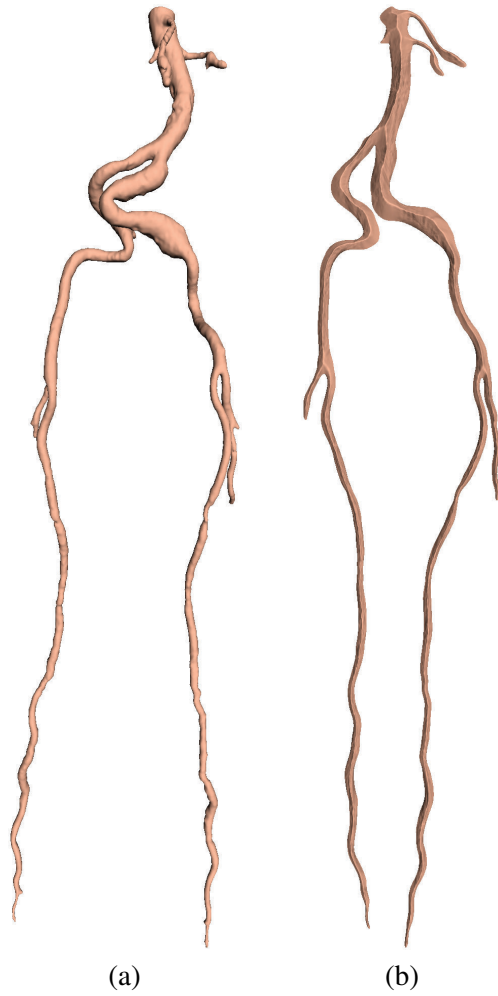
**Table 5.3:** User study results and improvement.

<i>Task</i>	<i>in 3D</i>	<i>in 2D</i>	<i>Improvement</i>
Search Test (time)	33.84 sec.	12.74 sec.	62.4 %
Search Test (accuracy)	88.0 %	96.3 %	9.4 %
<i>Task</i>	<i>w/o Borders</i>	<i>w/ Borders</i>	<i>Improvement</i>
Depth Test (time)	15.95 sec.	13.24 sec.	17.0 %

mapping; circles in the 2D texture image become ovals when mapped to areas which are distorted on the 3D model (see Figure 5.11).

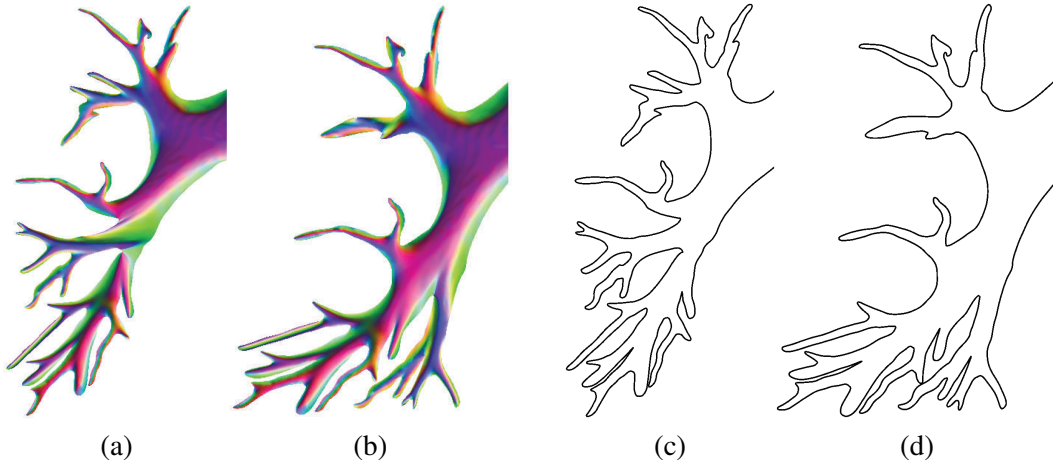
A small informal user study was performed with six subjects to assess the results from this technique. The user study consisted of two tasks. In the first task, the subject was asked to identify red spots placed randomly on the object, both in 3D and 2D. The purpose of this test was to evaluate the user's accuracy and the amount of time taken to perform the task. Accuracy improved from 88% to over 96% when using the 2D images since the users were able to plainly see all sections. The time required for the test also dropped drastically from an average of 33.84 seconds per dataset to 12.74 seconds per dataset. Again, the ability to efficiently view the entire structure without tumbling the view contributed to this improvement.

The second task was to identify which of two points was closer to him/her (that is, which is further away from the background). The purpose of this task was to evaluate the usefulness of the depth borders by seeing the time it took users to make the decision with and without depth borders. When there were no depth borders, the user was required to mentally connect between the 2D and 3D models. Using the depth borders allowed the average time for the task to drop from 15.95 seconds to 13.24 seconds. Users also noted that while they were generally able to mentally understand the relationships between the 2D maps and the original 3D structures, the depth cues indeed helped in making their decisions and making them more quickly. However, many subjects still spent at least some time going between the 2D and 3D models even when the depth borders were present. The accuracy was the same in both cases. The results of these two tasks are recorded in Table 5.3. The two additional mapped datasets (a second bronchi tree and the upper aorta) are shown in Figure 5.12.

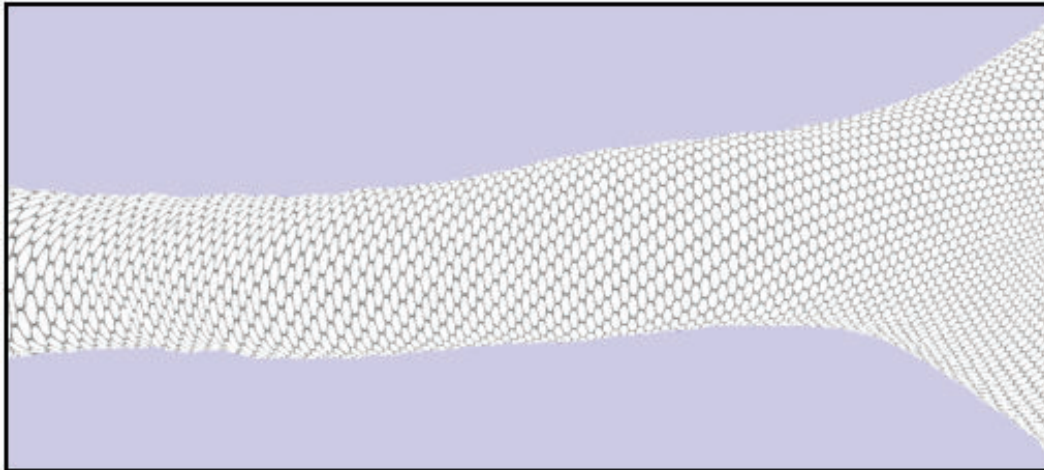


**Figure 5.9:** Planar map of lower limb blood vessels. (a) The original surface mesh. (b) The planar map with raised central path.

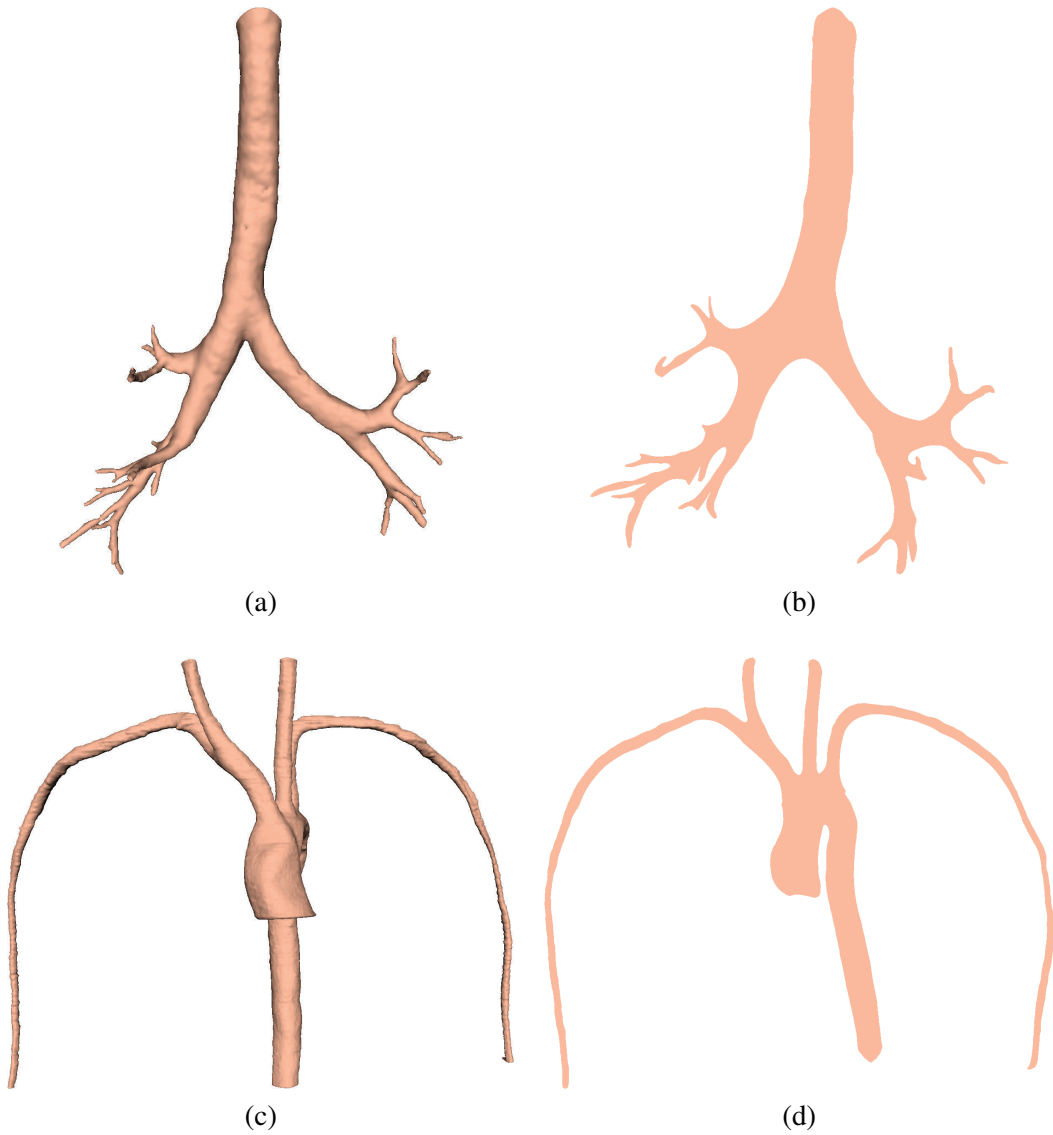




**Figure 5.10:** Modifying placement weights for better direct inspection. (a) Region using original placement weights. (b) Region using only the first node for placement weights. (c) Border of map in (a). (d) Border of map in (b).



**Figure 5.11:** Texture mapping of circles onto the original bronchial surface using parameterized map.



**Figure 5.12:** Further results of context preserving mapping. (a) Mesh model of a second bronchi model. (b) 2D map of second bronchi model. (c) Mesh model of upper aorta. (d) 2D map of upper aorta

## Chapter 6

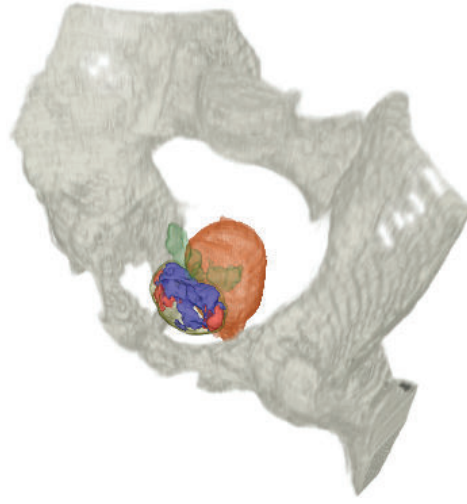
### Prostate Visualization

Combining multiple imaging modalities into a single view has been a staple of visualization research, and this chapter presents techniques for combining multiple MR modes from prostate imaging [91]. In addition to simply rendering the multi-modal data, techniques are also developed to aid in the exploration and interaction of the data within a multi-volume ray casting framework.

#### 6.1 Motivation

Prostate cancer (CaP) is the second most incident cancer for males worldwide [105]. However, diagnostic methods remain primitive and inexact, relying primarily on a non-specific blood test and a digital rectal examination. Non-cancerous prostate diseases can yield positive results, requiring biopsies to be performed. Though guided by transrectal ultrasound (US), these biopsies are inexact, and many are often necessary to successfully sample a cancerous region [81]. Recently, it has been noted that MRI can be used for detecting CaP [44], though multiple MR images obtained with different settings are necessary. Most commonly used is a combination of  $T_2$ -weighted ( $T_2$ -w) and  $T_1$ -weighted ( $T_1$ -w) image sequences, and the use of MRSI has also been suggested. This work is focused on using these three common acquisitions for visualizing the prostate, CaP, and the surrounding anatomy, as shown in Figure 6.1.

Radiologists will examine the MR data by viewing the slices and trying to correlate matching positions between scans in various orientations (e.g., axial and coronal). An automated analysis combined with a 3D rendering system, which would allow for viewing of the entire gland at once and include the data from each scan, would be more intuitive and efficient. This need for improved visualization techniques for treatment planning has been noted [123]. For physicians it is an additional tool to augment slice viewing, and can be used to assist in localizing the disease and for treatment planning. For the patient, a 3D visualization allows for



**Figure 6.1:** Visualization of the prostate gland and surrounding anatomy. The prostate is between the rectum (shown in burnt orange) and the pelvic bone. Suspected areas of cancer and hemorrhages in the gland are indicated by blue and red, respectively, and the seminal vesicles are in green.

a more intuitive explanation of their disease and can be helpful in discussing what course of action to pursue.

Although the prostate has been a subject of research for some time, most work in 3D has focused on shape reconstruction and the use of geometric models for simulation purposes [170]. Volume rendering has typically been used for illustration of the gland's size and location. The prostate and surrounding anatomy have been rendered from high resolution anatomical images of the Chinese visible human [154]. Simple rendering of MR and CT volumes to see the shape of the prostate for CT-based radiotherapy treatment planning has been proposed [56], and US has been used for brachytherapy planning [85]. Research has also been done to merge single photon emission computed tomography (SPECT) imaging, where the cancer is very apparent, with anatomical data for treatment planning [70].

As more imaging techniques become popular, multi-modal rendering has become common for dealing with the increased data. Though technically not multi-modal since they are all MR scans, the use of different acquisition types leads to similar considerations. Early discussions centered on how to combine multiple volumes for display [16]. In medical applications, multi-modal rendering is often used to combine functional data with anatomic data, such as positron emission tomography (PET) and CT [124], as well as for surgery planning [9]. In such works, values from different modes can be combined during rendering to achieve a desired look. Other work has performed multimodal fusion on two datasets, including two MRI

modes, using mutual information to decide at each voxel which dataset contains the more important information [13]. However, that work takes only a single value at each sample point to display. In the work presented here, it is necessary to view the data from multiple sources at the same time.

The  $T_2$ -w,  $T_1$ -w, and MRSI data are acquired at different levels of resolution, being approximately  $0.5 \times 0.5 \times 3$  mm,  $1 \times 1 \times 6$  mm, and  $7 \times 7 \times 3$  mm, respectively. Since it is desirable to visualize both the immediate area of the prostate (gland and seminal vesicles) as well as the general pelvic region (bladder, rectum, and pelvic bone), multi-volume rendering is necessary. For ray casting, it is common to use depth peeling and perform the ray casting in multiple passes [113, 125] or to do only certain chunks at one time [14]. Methods where the ray casting occurs in a single pass normally require the volume datasets to be preprocessed such that they are registered and resampled to a single grid [9].

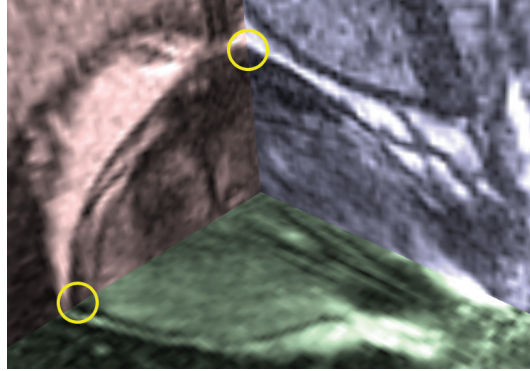
This chapter discusses work towards a virtual prostate visualization system [91]. A volume rendering framework is presented that allows the user to observe the prostate and its surrounding anatomy in an intuitive 3D environment. The concept of a *score volume* is introduced for visualizing the disease and methods are presented to observe all three types of multi-modal MR data in a single 3D view. This score volume can also be integrated into 2D slice-based views. User-driven importance rendering allows for different information to be emphasized based on the user's desires. A method of *visibility persistence* is presented, where a score of interest can automatically maintain visibility when it would be occluded, and overlaps are indicated. To render the surrounding anatomy, a single pass multi-volume ray caster accelerated on the GPU is used. A method of *interactive region painting* is also presented and integrated into the multi-volume ray caster.

## 6.2 Score Volumes

This work is based on creating *score volumes* to contain the detection results, and then rendering these score volumes rather than the original MR image data for the prostate region. The current system has been developed around a specific set of datasets, though it is generalizable to further imaging modalities and types of MR acquisition.

### 6.2.1 Standard Patient Dataset

The system has been built around the use of five image sequences: three approximately orthogonal  $T_2$ -w sequences (axial, sagittal, and coronal), one  $T_1$ -w axial sequence, and one MRSI sequence. For the  $T_2$ -w sequences, the images are focused



**Figure 6.2:** Natural alignment of three  $T_2$ -w MR slices, with axial tinted red, sagittal blue, and coronal green. The prostatic capsule aligns well between slices (yellow circles).

on the region of the prostate. The  $T_1$ -w images, acquired with lower resolution, encompass the entire pelvic region. As these were acquired using an endorectal coil, artifacts from it affect areas near the rectum. To overcome this, specialized software to negate this effect is provided with the MR machine, and it is these post-processed images that are utilized.

These scans were acquired during a single session, and as such, are naturally registered in world space. The image position, orientation, and pixel spacing information from the DICOM data can be used to align the volumes properly with respect to each other (see Figure 6.2). Due to this natural alignment, it is possible to calculate the corresponding positions in each volume with respect to another volume. In general, the alignment between orthogonal slices was observed to be within 0 - 1 mm of the correct position. Because of this, no formal registration methods are used. In cases where the datasets are not naturally aligned, MR registration algorithms have been proposed [170].

Segmentation of the prostate and surrounding anatomy is necessary since nearby structures are often within a similar range of values. Various methods have been proposed for the segmentation of the prostate data [170] as well as for the pelvic anatomy [106]. However, here efforts have been focused on scoring and visualization techniques for this data, and thus segmentation is outside the scope of this work. For accuracy, the data has been manually segmented (considered the current gold standard method) with guidance from an experienced radiologist. For this system, the label map segmentations for each of the four image volumes are taken. For the three  $T_2$ -w volumes, the PZ, CG, and SVs are segmented. For the  $T_1$ -w volume, the whole prostate gland, bladder, rectum, and bone are segmented.

## 6.2.2 Label Map Upsampling

Due to the low  $z$ -resolution of the MR image sequences, a shape based interpolation method is necessary to recreate intermediate slices and make the resolution near isotropic. One method of upsampling is to perform grey-level interpolation to create new images slices [33]. However, this increases the amount of segmentation work that would need to be performed. Rather, a method for interpolating slices of segmented volumes is required. Common methods for this take as input a binary segmentation [118] or a threshold that allows for a binary segmentation of the data [42]. However, for this work it is necessary to maintain a ternary segmentation, and thus a ternary shape based segmentation interpolation method is presented in which known regions are dilated into unknown regions to form the final shape in an interpolated slice. Further taken into account is the information from multiple labelmaps to create the final upsampled labelmap. The method is fast, easy to implement, and suitable for CaP visualization needs [90].

The first part of the upsampling method is to upsample each  $T_2$ -w ternary labelmap volume separately along its  $z$ -axis by interpolating new slices in order to reduce the interslice spacing to the level of the intraslice spacing. The ternary segmentation is encoded into the voxels as follows: voxels not belonging to the prostate are assigned a value of 0, voxels belonging to the PZ are assigned a value of 10, and voxels belonging to the CG are assigned a value of 30. The result of the upsampling will likewise contain only these three values when completed. A simple method based on iterative erosions and dilations has been developed which will take this ternary data into account, preserving the shape of the entire gland as well as of the individual zonal regions.

An interpolated slice is created midway between each pair of neighboring slices in the original labelmap volume. The algorithm contains four steps which are performed on the interpolated slice that is to be created. These four steps can be repeated as needed to reduce the interslice spacing of the volume to the level of the intraslice spacing. For the discussion that follows, the use of the term neighboring voxels refers to the two neighboring voxels from the two neighboring slices. That is, given two slices  $A$  and  $B$ , for the interpolated slice  $AB$  between  $A$  and  $B$ , a voxel  $v_{AB}$  with position  $(x,y)$  in the interpolated slice contains two neighbor voxels  $v_A$  and  $v_B$  with position  $(x,y)$  in slices  $A$  and  $B$ , respectively.

The first step in this algorithm is an initial labeling of all voxels in the interpolated slice. For every voxel  $v_{AB}$  in the interpolated slice, its value is set to be the mean of the two neighboring voxels,  $v_A$  and  $v_B$ . If both neighboring voxels are labeled as non-prostate, then the corresponding interpolated voxel must also be non-prostate and is correctly labeled 0. If both neighboring voxels are either PZ or CG, then the corresponding interpolated voxel must also be PZ or CG, and it is

correctly labeled as 10 or 30, respectively. If one neighboring voxel is PZ and the other is CG, then the interpolated voxel will certainly be in the prostate, but it is as yet undetermined as to whether it should be labeled as PZ or CG (its current value is set to 20). If the interpolated voxel is between a prostate voxel and a non-prostate voxel, then it will be labeled as uncertain (value of 5 or 15) and will be further processed.

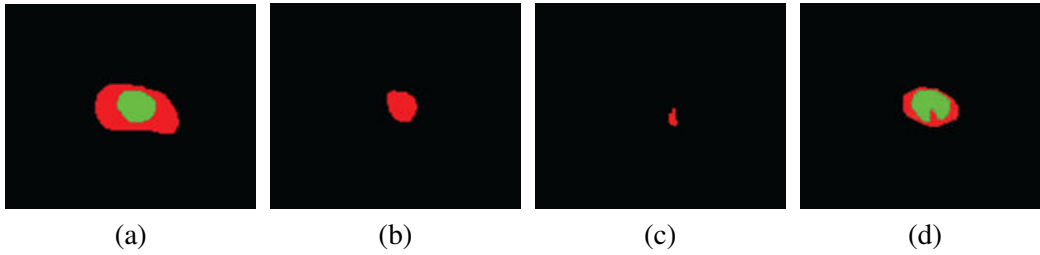
The second step is an erosion of the areas that have been labeled as uncertain (labeled as 5 or 15); that is, areas that could be inside or outside of the prostate. If an area is known to belong to the prostate, it is referred to as certain (note that voxels which must belong to the prostate but can be either PZ or CG are referred to as certain but undetermined). The uncertain regions are eroded by performing iterative dilations on the certain regions into the uncertain regions. After this step, all voxels in the interpolated slice will be labeled as one of the four certain types. Note that the undetermined voxels (labeled as 20) are also dilated, such that they grow outwards from their initial locations, as this is important for the next step.

The third step is to re-label voxels as belonging to the PZ or CG. For this, a decision is made for all of the undetermined voxels (labeled as 20). Since this region was grown during the previous step, some of these undetermined voxels will now have a prostate label in one neighboring slice and a non-prostate label in the other neighboring slice. Since these voxels are certain to be contained within the prostate, they are labeled with the PZ or CG label from its corresponding prostate neighbor (value 10 or 30).

The final step for the  $z$ -resolution upsampling is a further erosion of the remaining undetermined voxels (labeled as 20), which are guaranteed to belong to the prostate but are not yet labeled as PZ or CG. These voxels are eroded similarly to the second step above, though only the PZ labels (value of 10) and CG labels (value of 20) are allowed to grow into them, as it is known that the voxel must belong to the prostate and thus the non-prostate voxels (value of 0) are not allowed to grow into them. After this step, all voxels in the prostate will be labeled as belonging to either the PZ or CG.

After these four steps, every voxel will be labeled as either non-prostate (value of 0), PZ region (value of 10), or CG region (value of 30), preserving the ternary state of the labelmap. This method is necessary instead of a conventional binary shape-based interpolation approach in order to avoid gaps. If each prostate region (PZ and CG) is interpolated separately, gaps can occur in the resulting interpolated labelmap that should be covered by the prostate. An example of this problem is shown in Figure 6.3. This interpolation is repeated until the  $z$ -resolution of each labelmap volume is no worse than twice its corresponding  $x$ - and  $y$ -resolutions.





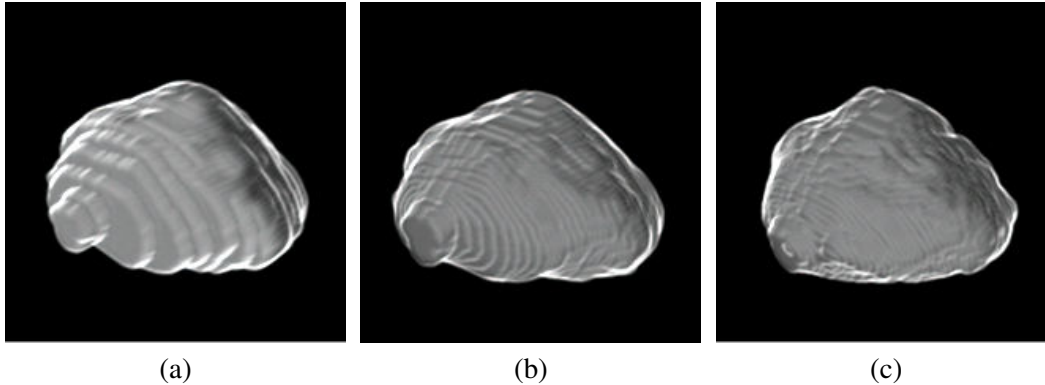
**Figure 6.3:** Example of the ternary labelmap interpolation. The PZ label is indicated by red and the CG by green. (a) and (b) Two neighboring slices from the original labelmap volume. (c) Interpolated slice when the PZ and CG regions are interpolated separately, resulting in a large missing area where the prostate should be. (d) Result from the ternary method, where the shape of the overall prostate segmentation has been preserved and no prostate area is missing.

### 6.2.3 Composite Label Map

The second part of the upsampling algorithm is to create a composite upsampled labelmap volume. All three upsampled labelmap volumes from the  $T_2$ -w data are used in creating this composite volume, capitalizing on the good intraslice resolution of the generally orthogonal datasets. That is, if the axial volume is taken as the canonical orientation for  $xyz$ , then it will have good resolution in  $x$  and  $y$ , but poor in  $z$ , and thus the segmentation might be off slightly in that dimension. However, the coronal volume will have good resolution in  $x$  and  $z$ , while the sagittal volume will have good resolution in  $y$  and  $z$ . In this way, each dimension will be encompassed by the good intraslice resolution data from two volumes.

For this composite labelmap, the axial  $T_2$ -w upsampled labelmap is taken as the coordinate system that will be used. For each voxel in the composite volume, an average labelmap is computed using the labelmap values from the three upsampled labelmaps. Only areas where either two or all three segmentations agree are preserved. That is, at least two of the three upsampled labelmaps must agree that a voxel is in the prostate in order for it to be labeled as such, helping to remove outliers. This composite labelmap results in a more accurate and visually pleasing representation of the prostatic volume.

The results of the simple prostate upsampling are shown for one dataset in Figure 6.4. The voxel spacing from the original image sequences is  $0.55 \times 0.55 \times 3$  mm for each of the sequences (axial, sagittal, and coronal). For this resolution of data, the upsampling interpolation will be performed twice for each of the three  $T_2$ -w image volumes, yielding an inter-slice resolution of 0.75 mm. Figure 6.4(a) shows an isosurface rendering of the original segmented prostate from the  $T_2$ -w axial image sequence. Due to the large interslice gap, very obvious plateaus are visible where



**Figure 6.4:** Results of the composite segmentation upsampling procedure. (a) Original axial segmentation of the prostate. (b) Upsampled result for the axial segmentation. (c) Composite result from a combination of the three shape interpolated upsampled segmentation volumes (axial, coronal, and sagittal). The composite segmentation is significantly less faceted and better illustrates the true shape of the prostate.

each slice was segmented. Figure 6.4(b) shows a rendering of the result from up-sampling the axial sequence alone. The plateau artifacts have been greatly reduced, though are still somewhat disturbing. Figure 6.4(c) shows a rendering of the final composite segmentation based on combining the upsampling of the axial, sagittal, and coronal sequences. The faceted artifacts have been further reduced, and the entire shape of the prostate is more full and accurate due to the contributions to the shape from the sagittal and coronal views.

#### 6.2.4 Score Volume Creation

This work in visualizing the prostate gland and suspicious regions is based on creating a score volume, where every voxel contains three values which are scores corresponding to each of the three types of MR acquisitions. The  $T_2$  score is based on detecting cancer from the  $T_2$ -w data. The  $T_1$  score is based on detecting regions of post-biopsy hemorrhage from the  $T_1$ -w data. The MRSI score is based on detecting areas of increased chemical ratios indicating the possibility of cancer existence. The  $T_1$  and MRSI scores are binary and the  $T_2$  score is quaternary. The inputs for the creation of the score volume are the five image sequences, the four upsampled segmentation label maps, and the composite label map.

Empirically determined thresholds are used to decide a score for each of the modes. These thresholds were defined by using a group of three datasets and manually observing the typical signal intensities for regions believed to be normal and abnormal in the PZ (decreased for  $T_2$ , increased for  $T_1$ , elevated spectra in MRSI).

The pathology results were used to ensure that samples were taken from sextants known to contain either cancer or hemorrhage. The scores are created as follows, with the default values being zero.

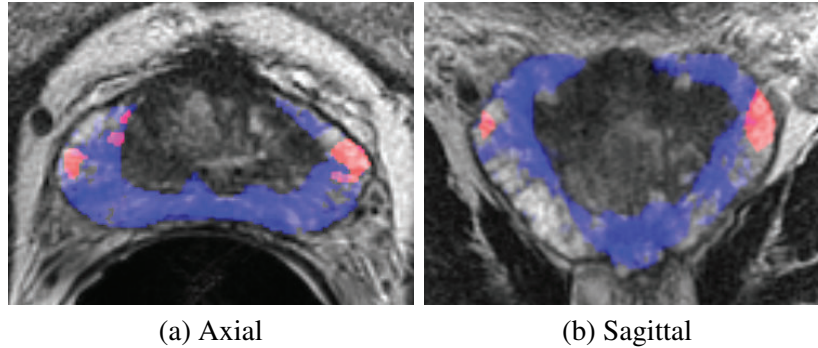
**$T_2$  Score (PZ):** Decreased  $T_2$ -w intensity in the PZ is indicative of cancer, so voxels which are below the  $T_2$  threshold are of interest. Since there are three volumes of  $T_2$ -w data, all of them are sampled to take advantage of each volume's high intraslice resolution. Each volume's score contributes one third towards the final score value, which can therefore contain four values (0, 0.33, 0.67, and 1).

**$T_1$  Score (PZ):** Increased  $T_1$ -w intensity in the prostate is indicative of post-biopsy hemorrhage, so voxels which are above the  $T_1$  threshold are of interest. The single  $T_1$ -w volume contributes fully to the final score value.

**MRSI Score (PZ and CG):** An increase in one or both of the spectroscopic ratios in the MRSI data is indicative of prostate cancer. If either of the two ratios are above the MRSI threshold, then the voxel is scored as being of interest. This scoring system, unlike for the  $T_2$  and  $T_1$  scores, is applied to both the PZ and CG.

**$T_2$  Score (SVs):** Similar to the  $T_2$  scoring for the PZ, decreased  $T_2$ -w image intensity in the SVs is indicative of cancer. However, the SVs pose a hurdle in that their walls (both interior and exterior) also appear with decreased  $T_2$ -w intensity. To account for this, a three part scoring process is used. First, each  $T_2$ -w sequence (axial, sagittal, and coronal) is scored individually. Their individual score volumes are then eroded by two voxels to remove thin boundaries. The final SV score is created with each of the individual scores contributing one third to the final score.

The neighboring regions of the PZ (prostatic capsule and CG) are generally dark, and thus could yield false positive results if included accidentally as part of the PZ. To account for this, the border voxels are not scored. To ensure that samples are from the correct region for each of the three  $T_2$ -w volumes, the upsampled label maps are taken into account for each volume, and that volume is sampled only if its label map indicates the region is correct. Likewise, the upsampled label map of the  $T_1$ -w volume is provided to ensure samples are from within the prostate. Since areas immediately outside of the prostate are often of high intensity in  $T_1$ -w data, they could be mistaken as indicators of a hemorrhage if improperly sampled. Trilinear interpolation is used when sampling from the upsampled label maps and tricubic interpolation is used when sampling from the original MR data.



**Figure 6.5:** Integrating the score volume with slice-based viewing. The score information is overlaid on the MR slices, with blue indicating  $T_2$  scores and red indicating  $T_1$  scores.

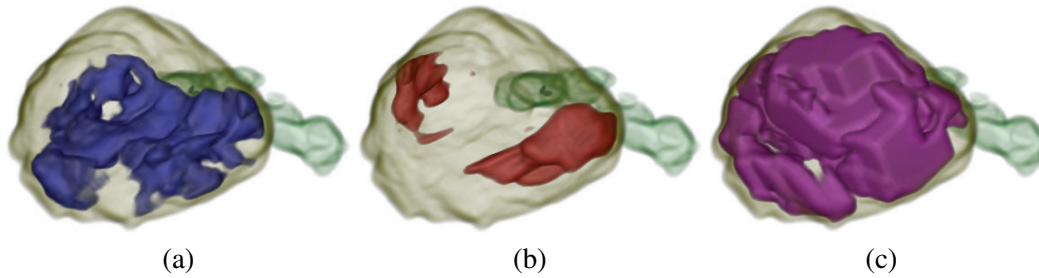
### 6.2.5 2D Slice-Based Visualization

The created score volume can be integrated into a 2D slice-based viewing system to provide guidance for the radiologist in viewing the slices by presenting information from other slices on the slice being viewed. For each voxel in a slice being viewed, the score values from the score volume can be found and overlaid on the grayscale image. Though the score volume is aligned with the axial  $T_2$ -w image sequence, it can be interpolated to obtain values for the corresponding pixels in the other image sequences. Examples of this are shown in Figure 6.5, where the  $T_2$  and  $T_1$  scores are shown as blue and red overlays, respectively. The user can adjust the opacity of the overlays as desired.

## 6.3 3D Visualization

A 3D volume rendered view of medical imagery is an intuitive method of visualizing the data and obtaining a good sense of the relationship between objects. The system allows the user to visualize the prostate region and the surrounding anatomy of the pelvic region. For the prostate region, using the score volume allows the user to visualize tumor and hemorrhage locations. The inputs for the volume rendering framework are the following four volume files:

1. Score volume (prostate region)
2. Composite label map volume (prostate region)
3. Upsampled label map of  $T_1$ -w volume (pelvic region)
4.  $T_1$ -w pelvic MR volume (pelvic region)



**Figure 6.6:** Rendering examples of individual score values. (a) The  $T_2$  score values with blue indicating cancer. (b) The  $T_1$  score values with red indicating hemorrhages. (c) The MRSI score values with purple indicating elevated ratios.

The two prostate region volumes occupy the same volumetric space, and the two pelvic region volumes likewise occupy the same volumetric space. For rendering the surrounding anatomy, especially the bones, it is necessary to make use of the pelvic region volumes, which encompass a much greater area than the prostate region volumes. Since it is undesirable to scale or resample the data, multi-volume rendering is performed through these two volumetric spaces. If only the prostate region is to be viewed, this multi-volume rendering is not necessary.

Two preprocessing operations are performed before rendering. The prostate region volumes are both trimmed so that surrounding area is removed where no anatomy is labeled. This trimming is done such that a 3 mm border remains around the cuboid region of interest and will typically reduce its size to 15% of the original. Since all of the data has been based on binary segmentations with no smooth gradients between labeled and non-labeled regions, the score volume and both label map volumes are filtered with a  $3 \times 3 \times 3$  mean filter to improve the rendering results.

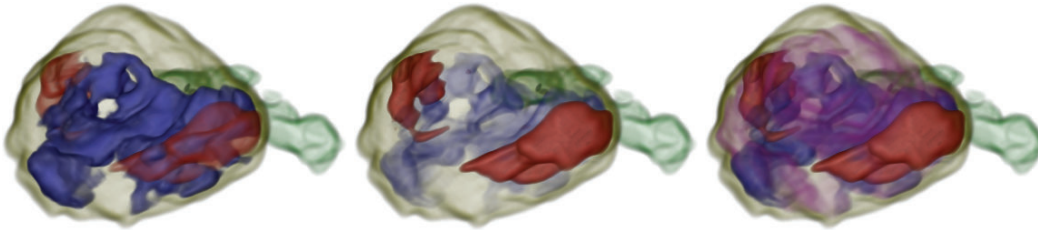
### 6.3.1 Standard Prostate Region

Visualization of the prostate region is based on using the composite label map volume and the score volume. For rendering the interior areas of the gland and SVs, volume rendering is performed on the score volume. The score volume contains three values per voxel, corresponding to the  $T_2$  score,  $T_1$  score, and MRSI score. The user can view each of the values individually, or combined as desired. For the surface of the gland, direct semi-transparent isosurface rendering of the composite label map is used.

The following color scheme has been developed for the score values. For a high  $T_1$  score, indicating hemorrhage, red is the obvious choice. For regions with a high  $T_2$  score, blue is used to represent the location of suspect cancerous areas. For the MRSI score, purple is used to indicate increased ratios. The prostate gland



**Figure 6.7:** Seminal vesicles indicating bilateral invasion.



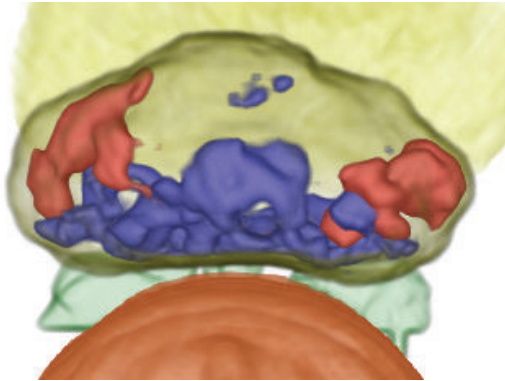
**Figure 6.8:** Rendering examples of the score values with various levels of transparency.

itself is rendered as a semi-transparent tan color and the seminal vesicles as a semi-transparent green color. The transfer functions controlling the gland colors (prostate and SVs) are applied to the label map volume, while the transfer functions for the score colors are applied to the score volume. The  $T_2$ -w data itself is not used in the volume rendering. The user is presented with two standard options for rendering the prostate score data:

**Isosurface Score View:** The solid isosurface renderings of each of the score values are viewed. This mode is typically done with a single score value at a time. Examples of the three scores rendered individually can be seen in Figure 6.6. Since the  $T_2$  score is a quaternary value, the isosurface is set so that a score  $\geq 0.66$  is required (i.e., at least two of the  $T_2$ -w volumes indicated decreased signal).

Since only  $T_2$  scores exist within the seminal vesicles, its coloring is tied to that of the  $T_2$  score for the prostate and uses the same blue color. An example of the seminal vesicles with SV invasion indicated is shown in Figure 6.7.

**Transparent Score View:** When viewing multiple scores together, user-defined transparency per score is typically used. This is most useful if the user wants to see relationships between different scores (e.g., between a cancerous  $T_2$  score and a hemorrhage  $T_1$  score). Examples of combinations of multiple score renderings with transparency are shown in Figure 6.8.



**Figure 6.9:** Viewing the angle between the prostate and the rectum (endorectal coil), looking towards the patient's head.

### 6.3.2 Surrounding Anatomy

Single-pass multi-volume rendering is used when including the surrounding anatomy. For each ray, its starting position and direction are calculated for both the prostate region volume and the pelvic region volume. The steps along each ray are adjusted to be the same length, such that steps along one ray are colocated with steps along the other ray, and the number of steps to enter and exit each of the volumes is calculated. Since the pelvic region is larger and fully encompasses the prostate region, it is guaranteed that a sample position in the prostate region will also be within the pelvic region, though most sample points within the pelvic region will not be within the prostate region. Separately calculated for the rays are the number of steps within the pelvic region before reaching the prostate region, the number of steps within both, and the number of steps after the prostate region until exiting the pelvic region. Using these values, the rays are cast through the volumes, and the prostate region is sampled only when the current position is within the correct range.

The pelvis and other nearby bones account for the majority of the area in the pelvic region volumes. When the bones are not being rendered, the minimum and maximum extent of the remaining anatomy (bladder and rectum) are calculated and the sampling rays are cast only through this bounding box, reducing the amount of area being traversed to approximately 10% of the full size. The prostate region is between these two objects and thus will not be missed.

The rectum is rendered as an orange semi-transparent isosurface. The user is able to easily observe the angle between the rectum and the prostate surface (see Figure 6.9). Similar to the rectum, the bladder is rendered as a yellow semi-transparent isosurface. The bones are rendered as slightly off-white isosurfaces.

The user is not able to modify the translucency of these objects, though the visibility of each of these three objects can be toggled on or off as desired.

Unlike for the prostate region, the transfer functions for the pelvic region are applied to the  $T_1$ -w MR data, allowing for a less precise segmentation. Due to this, these objects can take on a stepped appearance in the superior-inferior direction because of the low interslice resolution. However, the shape of these objects is not as important as that of the prostate, which is fully smoothed due to the rendering of the upsampled label map and score data directly.

## 6.4 Exploration and Interaction

### 6.4.1 Visibility Persistence

A technique for viewing occluded structures by peeling back outer layers of the object has been presented [120]. Methods for importance-driven rendering, allowing for a region of interest to be preserved, have also been proposed [144]. In that work, they perform composition in order to present the important structures when they may otherwise be obstructed. This is expanded on to provide contextual feedback of overlapping regions. Since multiple score values are being used at the same position which can have a relationship to each other, a tinting function is introduced to color regions that overlap. This allows the user to further distinguish within the view whether the score of interest overlaps another score, or whether it is simply occluded.

In addition to standard rendering of the prostate score volumes, a score rendering method called visibility persistence is provided. This mode will assist in keeping a score of interest (i.e., the persistent score) visible when other scores may occlude it. This mode will also indicate whether the occluded score is inside of the occluding score (i.e., they overlap). This is accomplished by tinting it green, a color which is otherwise not used within the prostate.

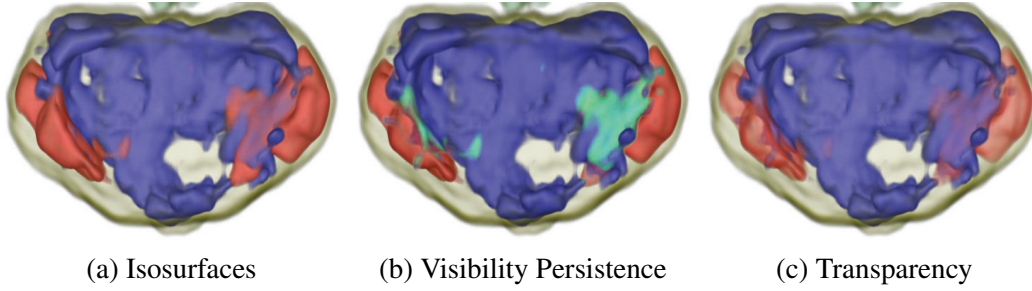
The visibility persistence is accomplished using a modified compositing strategy. A second volume rendering integral is accumulated with reduced color and opacity values for the non-persistent scores. The discretized volume rendering integral will then contain the standard front-to-back compositing as such:

$$\begin{aligned} C_{dst} &\leftarrow C_{src} \times (1 - \alpha_{dst}) + C_{dst} \\ \alpha_{dst} &\leftarrow \alpha_{src} \times (1 - \alpha_{dst}) + \alpha_{dst} \end{aligned}$$

where:

$$\begin{aligned} C_{src} &\leftarrow C_{gland} + C_{PersistentScore} + C_{OtherScores} \\ \alpha_{src} &\leftarrow \alpha_{gland} + \alpha_{PersistentScore} + \alpha_{OtherScores} \end{aligned}$$





**Figure 6.10:** Rendering example of  $T_2$  (blue) and  $T_1$  (red) scores with visibility persistence for the  $T_2$  score (green).

and will also contain:

$$\begin{aligned}
 C_{dst2} &\leftarrow C_{src2} \times (1 - \alpha_{dst2}) + C_{dst2} \\
 \alpha_{dst2} &\leftarrow \alpha_{src2} \times (1 - \alpha_{dst2}) + \alpha_{dst2} \\
 \alpha_{score} &\leftarrow \alpha_{PersistentScore} \times (1 - \alpha_{score}) + \alpha_{score}
 \end{aligned}$$

where:

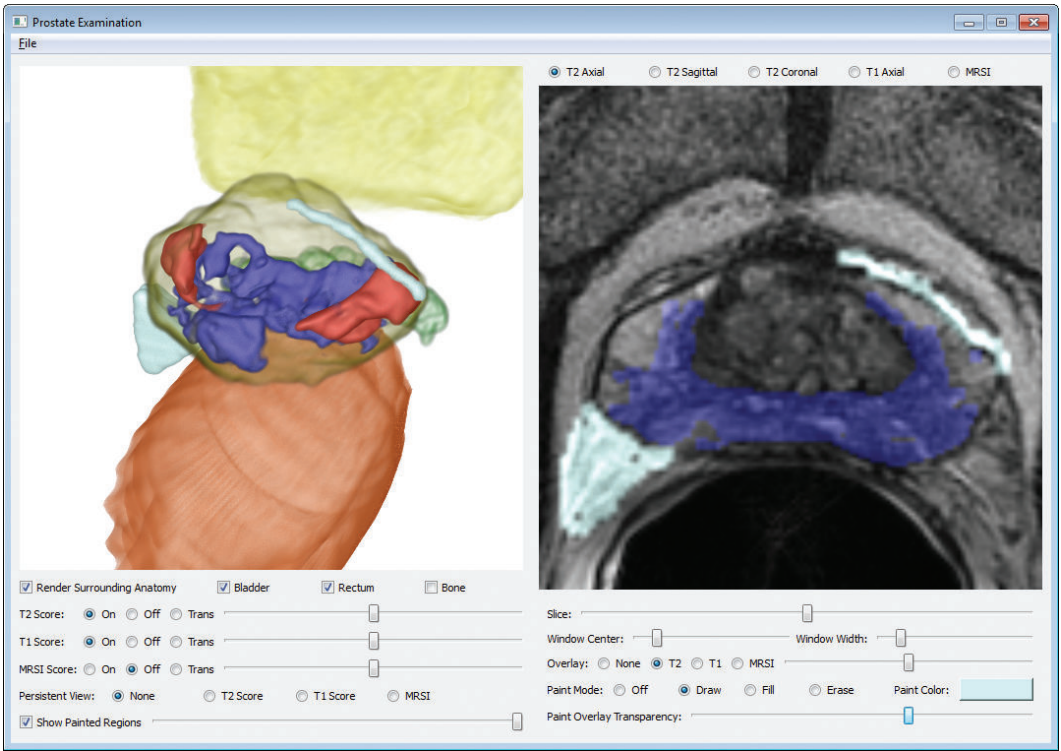
$$\begin{aligned}
 C_{Tint} &\leftarrow Tinter(C_{OtherScores}) \\
 C_{Tint} &\leftarrow C_{PersistentScore} + C_{Tint} \times \alpha_{PersistentScore} \\
 C_{src2} &\leftarrow C_{gland} + C_{Tint} + 0.1 \times C_{OtherScores} \\
 \alpha_{src2} &\leftarrow \alpha_{gland} + \alpha_{PersistentScore} + 0.1 \times \alpha_{OtherScores}
 \end{aligned}$$

The *Tinter* function accumulates the intensities of the non-persistent score colors into the green channel of  $C_{Tint}$ , which is then modulated by  $\alpha_{PersistentScore}$  and added to  $C_{PersistentScore}$ . At the end of the volume rendering integral, the final output color and opacity are composited as such:

$$\begin{aligned}
 C_{dst} &\leftarrow C_{dst2} \times \alpha_{score} + C_{dst} \times (1 - \alpha_{score}) \\
 \alpha_{dst} &\leftarrow \alpha_{dst2} \times \alpha_{score} + \alpha_{dst} \times (1 - \alpha_{score})
 \end{aligned}$$

where the  $\alpha_{score}$  value for blending is used to prevent a jagged halo effect around the persistent score.

As shown in Figure 6.10, the spatial relationships of the scores and gland are fully maintained by this method. In contrast to making one score transparent to view another occluded behind it, this mode allows for an occluded score to automatically be visible, while the other scores will maintain their full opacity unless they are occluding the score of interest. This method also gives context by showing if one score is inside of another, or simply behind it, by tinting the persistent score green when the scores overlap.



**Figure 6.11:** Screenshot of the user interface with examples of light blue user painted regions in both 3D and 2D views.

**6.4.2 Interactive Region Painting**

The ability of the user to interact with the underlying volume itself has also been explored. Earlier work on the editing of volumes has focused on various ways of displaying the existing data, such as painting on surfaces and making annotations, cutting windows within structures, or selecting regions to render differently [15]. In contrast to this, instead of being able to edit the already existing data, the user is allowed to draw new structures on-the-fly within the data which are then rendered as 3D objects in a desired color along with the existing data.

The ability for the user to manually denote regions of interest on the 2D slices is provided, and these regions are then updated in the 3D view. Within the user interface (Figure 6.11), the user can quickly switch between slice views and can view a slice directly next to the 3D rendering. Demarcations within one slice can be viewed on the other slices for correlation, and the user can modify as required by looking at all views. The user is also able to select their own custom colors when painting lines or filling regions so that separate entities can be easily distinguished and correlated between the 2D slice views and the 3D rendering view.

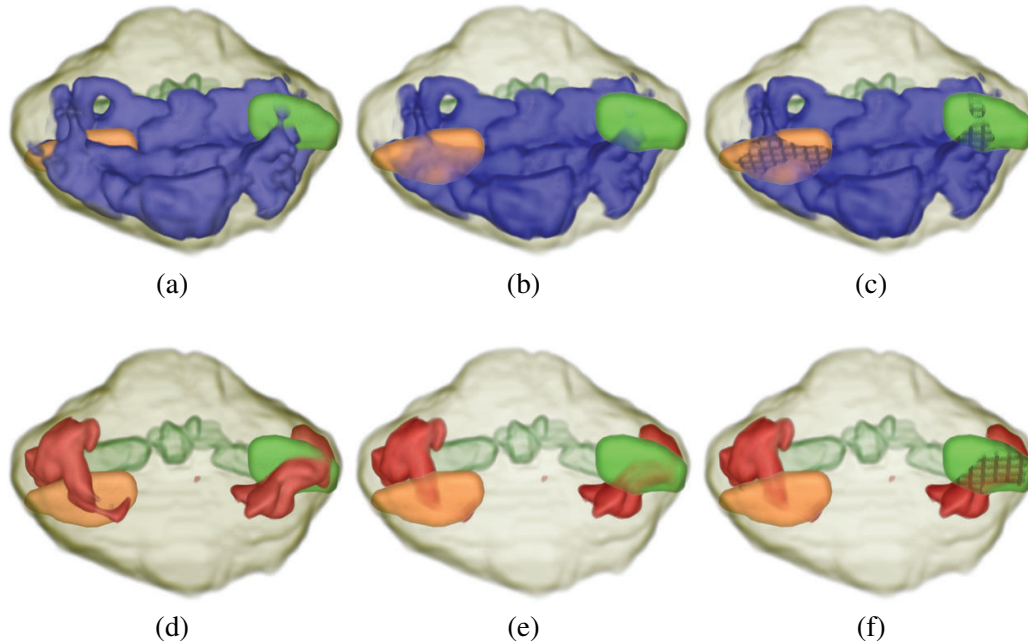
To accomplish this, a new paint volume based on the upsampled  $T_1$ -w label map is created with four channels to allow for various colors to be held. One difference between the dimensions of this volume and that of the upsampled  $T_1$  label map is that the paint volume has twice the resolution in the  $z$ -dimension (0.75 mm) to allow for greater accuracy when denoting areas on the  $T_2$ -w slices. Though coarser than the  $T_2$ -w images, this resolution is sufficient for marking areas; since the paint volume must cover the entire region (prostate and pelvis), maintaining a higher resolution would require more memory and processing power with little advantage.

When the user draws out a region, a custom color of choice can be selected to be used as both the overlay in the slice view and for the 3D rendering. As with the normal score volume, these painted regions can be turned on, off, or transparent as desired by the user. For rendering, the paint volume is smoothed with a  $3 \times 3 \times 3$  mean filter to yield a more natural look. The minimum and maximum extent of these regions are used in defining a bounding box for ray casting so empty areas do not need to be traversed. No transfer function is used, as the color and opacity information is supplied directly in the paint volume.

### 6.4.3 Visibility Persistence in Painted Regions

While the previously presented method of visibility persistence works well for such a closed system with predetermined colors, it does not translate well to the open system of allowing the user to paint custom regions within the prostate. In such a case, this default tinting of the custom painted regions is not applicable, as the user can choose any color to paint the regions in. While it would be possible to restrict the user's choice of colors, this was not deemed a prudent course of action.

For general occlusions, the same technique is used so that structures in front of the painted regions will appear with reduced opacity, allowing the painted region to be visible. For regions of overlap, it was decided to place a striped pattern on the painted region to indicate the overlap. Since this is not color dependent, the user is free to choose any colors for the painted regions (which do not conflict with the default  $T_2$  and  $T_1$  score colors). The presence of this striped pattern is based on the alpha value of the  $T_1$  and  $T_2$  scores at the sampling location. The stripes are created by halving the RGB color values for the painted region, which results in more of the overlapping color being present. The striped pattern itself is determined by the sample position within the 3D volume, so the pattern remains consistent from varying viewpoints. Examples of this updated method are shown in Figure 6.12.



**Figure 6.12:** Examples of persistence visibility for custom painted regions.  $T_2$  scores are rendered in blue,  $T_1$  scores in red, and two custom regions are present in orange and green. (a) and (d) Normal isosurface rendering. (b) and (e) Visibility persistence with no overlap indications. (c) and (f) Visibility persistence with overlap indications as striped patterns. In (e), for example, it can be seen how it might be ambivalent as to whether or not the painted orange and green regions are inside of the  $T_1$  score data when simply revealing occluded structures. However, in (f), it can be clearly seen that the green region does overlap with portions of the  $T_1$  score data, while the orange region does not.

## 6.5 Results

The data used in this work comes from the ACRIN 6659 study [150] and the current system has been built around their standard clinical protocol for MR imaging of the prostate. This ACRIN study was a large, multi-institutional study, and one goal was to be able to establish a protocol for MR acquisition that would be followed at multiple hospitals in order to obtain consistent data. The properties of a typical dataset, including original and processed data, are shown in Table 6.1.

### 6.5.1 Performance

There are four basic computational processes in the pipeline. First, the four label map segmentations are upsampled. Then, the three upsampled  $T_2$ -w label maps are combined to form a composite label map. Third, the score volume is created.

**Table 6.1:** Typical dataset properties. In the dataset volume descriptions, the Composite Label Map is represented by CLM. The number of data channels for each voxel is indicated in the last column (Ch.).

<i>Dataset Volume</i>	<i>Volume Size (voxels)</i>	<i>Voxel Spacing (mm)</i>	<i>Data Type</i>	<i>Ch.</i>
$T_2$ -w Axial, Sagit., Coron.	$256 \times 256 \times 20$ -24	$0.55 \times 0.55 \times 3.00$	16-bit int	1
$T_1$ -w Axial	$256 \times 256 \times 34$	$0.93 \times 0.93 \times 6.00$	16-bit int	1
MRSI	$7 \times 3 \times 12$	$6.88 \times 7.31 \times 3.00$	32-bit float	2
Score Volume, CLM	$256 \times 256 \times 89$	$0.55 \times 0.55 \times 0.75$	8-bit uint	3
Trim Score Vol, CLM	$115 \times 89 \times 78$	$0.55 \times 0.55 \times 0.75$	8-bit uint	3
Upsample $T_1$ Label Map	$256 \times 256 \times 133$	$0.94 \times 0.94 \times 1.50$	8-bit uint	4

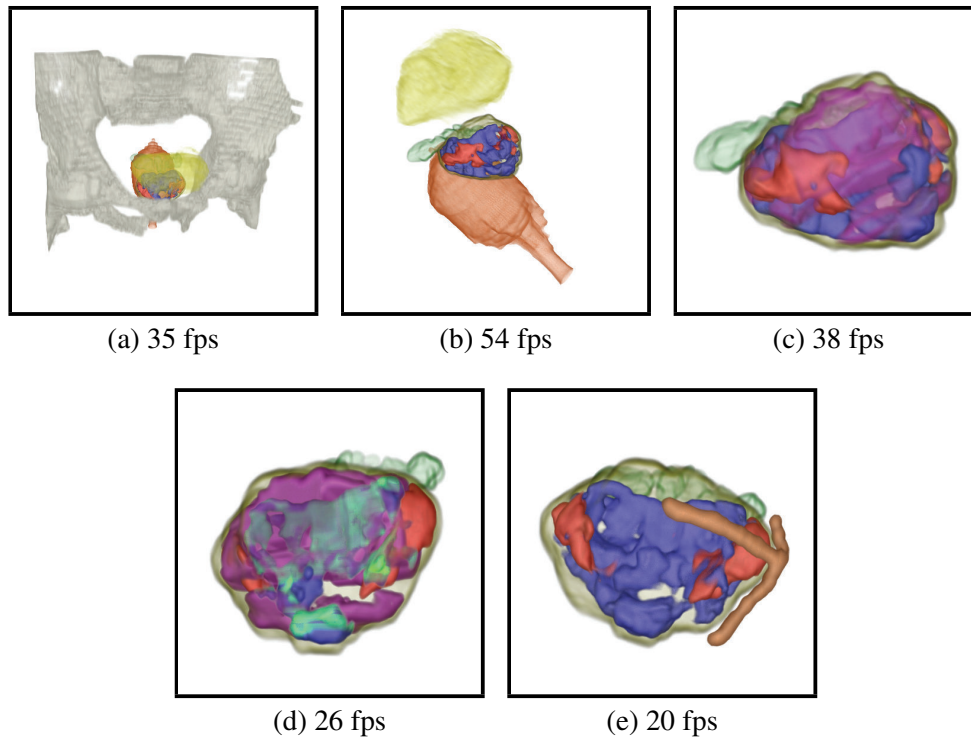
Finally, the score volume and composite label map volume are trimmed and these two along with the upsampled  $T_1$ -w label map are mean filtered to improve the rendering results. The average running time for these steps is 14 seconds.

The performance of the volume rendering varies depending on what regions and objects are being viewed. All renderings maintain interactive performance. Figure 6.13 shows the five basic types of rendering with differing performance (all objects, without bone, prostate only, persistence, and painted region), and their performance in frames per second (fps). These have a render target of  $512 \times 512$  pixels.

### 6.5.2 Score Evaluation

Although the focus in this system is on developing the visualization techniques for a CaP detection system, the scoring system has been evaluated by comparing against the results of the ACRIN 6659 study. For this study, the MR acquisitions were performed on patients 4-6 weeks after needle biopsy and before radical prostatectomy. The determinations of both radiologists and pathologists were denoted for each sextant of the prostate. The six sextants are formed by splitting the gland into left and right halves and further splitting them into apex, midgland, and base sections from top to bottom. Since the results from the MRSI are very broad and non-specific, only the  $T_2$  and  $T_1$  scores are considered in the evaluation.

For each patient dataset, a total of eight radiologists reviewed the MR data and made determinations as to the presence of cancer and hemorrhage on a per-sextant basis. For the cancer determinations, a ranking on a scale of one to five was used, with one indicating definitely no cancer and five indicating definitely cancer. To compare against the system's results, the minimum and maximum scores are dropped (to eliminate outliers), the remaining six are averaged, and an average of three or greater is considered to indicate cancer. For determining hemorrhage, the radiologists' results are taken as the ground truth since this is not indicated from the



**Figure 6.13:** Performance examples for the five main types of rendering. (a) All objects in the scene. (b) All objects except for the bones. (c) Prostate region only. (d) Visibility persistence. (e) User painted custom regions.

pathology.

Given the excised prostate, a pathological analysis was performed on it, and results were reported for cancer in the prostate, again on a per-sextant basis. The pathology report also indicates whether or not there was invasion of the seminal vesicles. The results from the pathology (both cancer determination in the prostate and seminal vesicle invasion) are taken as the ground truth in evaluating the system.

The results of testing the system on three datasets (different from those used in determining the thresholds) are summarized in Tables 6.2-6.4, where [GT] indicates the ground truth column. For the determination of cancer in a sextant (Table 6.2), it is noted that the results from the system were better than from the radiologists. However, further work is needed to see how accurate the specific regions are (this information is not provided with the ACRIN data). This method was less successful for detecting SV invasion (Table 6.3), though this problem is more difficult (the radiologists in the ACRIN study were not asked to determine this). For the detection of hemorrhages (Table 6.4), the simple threshold method proves to be very effective, no doubt due to the large disparity between normal and hemorrhaged tissue.

**Table 6.2:** Sextant evaluation for cancer in the prostate.

<i>Cancer Diagnosis</i>	<i>Pathology [GT] (# Sextants)</i>	<i>T<sub>2</sub> Score (# Correct)</i>	<i>Radiologists (# Correct)</i>
cancer	15	14	7
no cancer	3	3	2

**Table 6.3:** Seminal vesicle evaluation for cancer invasion.

<i>Seminal Vesicle Invasion</i>	<i>Pathology [GT] (# Cases)</i>	<i>T<sub>2</sub> Score (# Correct)</i>
yes	2	1
no	1	0

The extent of consensus among the radiologists has also been calculated when their average opinion was incorrect. For this, the radiologists' scores were ternarized into negative (1-2), undecided (3), and positive (4-5). First removing the undecided scores so as to focus on the deterministic scores, the percentage of the scores which yielded the incorrect average opinion was calculated; a larger percentage indicates greater agreement amongst the radiologists. For the nine incorrect sextants in Table 6.2, three had a consensus in the range of 85% - 100%, three were in the range of 75% - 84%, and the remaining were in the range of 60% - 74%.

### 6.5.3 Physician Evaluation

The current system was shown to and discussed with a radiologist and positive feedback was received from him. He noted the possible utility of the system for surgery planning, and observed the utility of being able to manually indicate regions and visualize shapes. Regarding the use of MRSI, he noted that it was most often not of use, and was focused more on the  $T_2$ -w and  $T_1$ -w data. It was also observed that the color scheme chosen was intuitive and easy to understand.

**Table 6.4:** Sextant evaluation for hemorrhage.

<i>Post-Biopsy Hemorrhage</i>	<i>Radiologists [GT] (# Sextants)</i>	<i>T<sub>1</sub> Score (# Correct)</i>
hemorrhage	6	6
no hemorrhage	12	12



## Chapter 7

### Conclusions and Future Work

#### 7.1 Conclusions

Medical visualization is an important area of research. There is a high emphasis on using imaging modalities to assist in diagnosing cancer, and visualization techniques help to improve the usability. The research presented here has been developed to aid in improving the diagnosis of three of the most prevalent cancers: colon, lung, and prostate cancer.

Visualization is needed to make the large amounts of data more accessible to the user. Research on virtual colonoscopy and virtual bronchoscopy has been ongoing for many years, and thus the research presented here can be considered as extensions. In contrast, there has been little research on visualization for prostate cancer detection, and the research here was performed as a basis for such a system to develop from.

The integration of VC techniques into the typical OC is an interesting field of research, and herein were provided two possible ideas towards such a larger system. The first is a method of removing the radial distortion from the endoscopic images so that they appear with normal perspective distortion instead. Also, a method of creating a correlation between the VC centerline and the OC hugging corners path was proposed.

Supine-prone colon registration allows for matching regions from two separately scanned colons of the same patient to be presented to the user in an intuitive manner so that they can be easily compared. Presented herein was a method for performing this work based on quasi-conformal mapping. The results from this are rectangular maps that are one-to-one and onto, allowing for corresponding visualizations in both the 2D and 3D domains.

Using a 3D mesh model of a colon as a map presents problems during navigation, as occlusions can cause questions to be raised as to one's precise location. To mitigate this problem, a map with no occlusions would be useful, and a 2D map

would be preferred so that the entire colon surface can be viewed at once. For this, a method was presented which will create a 2D boundary based on the centerline of the colon and then the colon surface is mapped to this 2D boundary.

The idea of a context preserving parameterization of a mesh model can also be applied to treelike structures, such as the bronchi or blood vessels. However, this is not a straight extension of the technique. In treelike structures, new items must be taken into account, such as the order of branch placement and preservation of depth to display to the end user. In accounting for these, a technique was developed which was shown to yield better results for searching and identifying regions.

For the prostate, the idea of a 3D visualization system for exploring MRI scans for diagnosis is new. Presented was a method to create score volumes from the MR data, which can then be rendered as desired by the user. Also included in the rendering are the surrounding anatomy (rectum, pelvis, and bladder). Special rendering modes, such as visibility persistence and interactive region painting, further allow the user to experiment with the data.

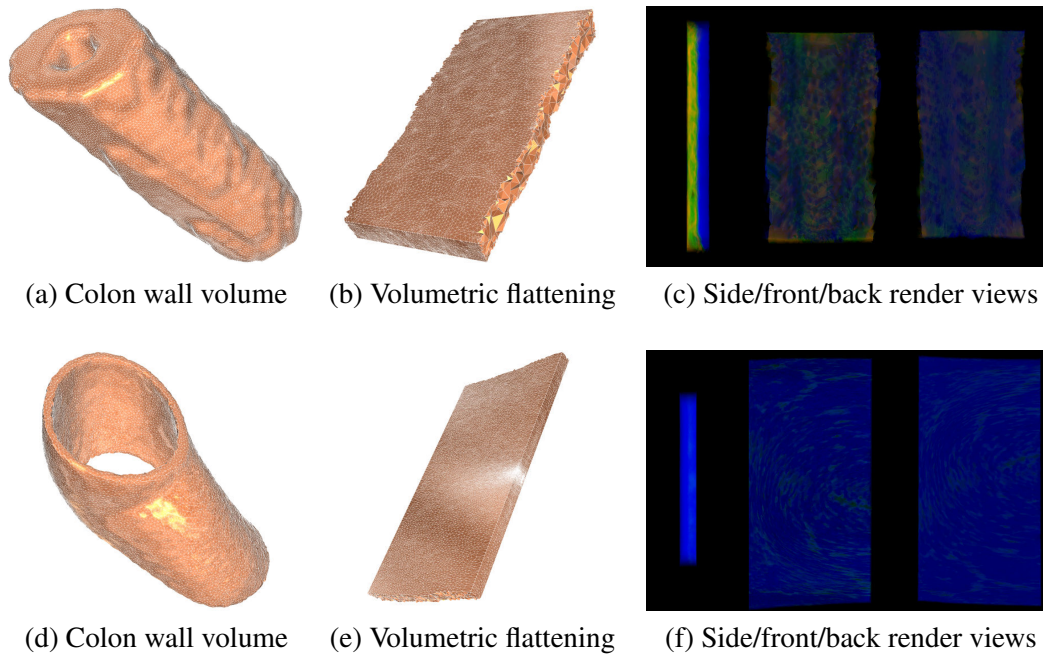
## **7.2 Future Work**

There are several available directions for future work for the colon, the prostate, and other related organs. For the colon, there is the possibility of improving registration quality through the use of thickly segmented colon walls. There is also interest in developing better tools for analyzing the structure of polyps. For the prostate, there are several further MR acquisition modes which can be analyzed and integrated into the visualization system, as well as possibilities for treatment simulations. These future directions can be divided into short term and long term goals.

### **7.2.1 Short Term**

Using an outer wall colon segmentation, it is possible to map a *thick* colon surface to a volumetric cuboid [163]. This has possibilities for use in improving the registration quality. By mapping both supine and prone to volumetric cuboids, a volumetric rather than simply surface registration can be achieved. Challenges for this further research include identifying additional features both on the outer wall boundary as well as within the colon wall to use in performing the registration. Though not discussed in this work, such thick colons would also have the possibility of assisting in improving CAD results [49].

Similar to the proposed work for thick volumetric registration of tetrahedral colons, registration of prostate models can be similarly performed. The prostate



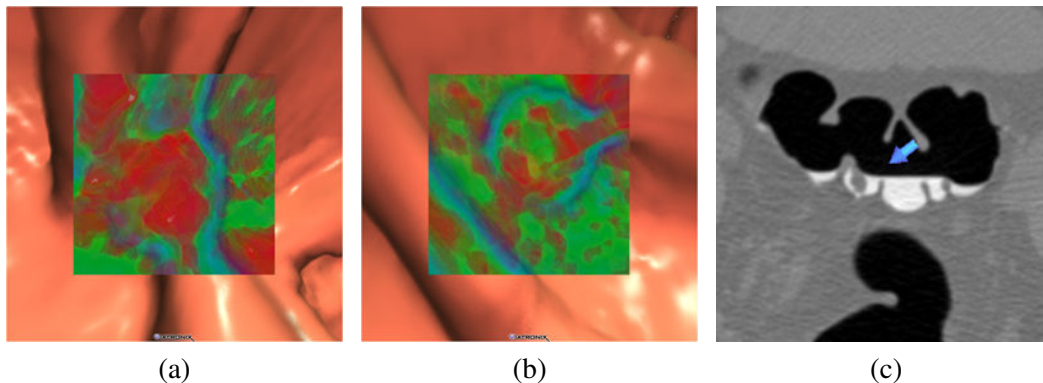
**Figure 7.1:** Rendering of a flattened volumetric colon wall. (a) and (d) Colon wall volumes with triangulation. (b) and (e) Flattening results. (c) and (f) Volumetric rendering results as electronic biopsy images, with side, front, and back views. The tissue density is color encoded, the yellow and red colors represent the higher density, the blue color the lower.

models can be mapped to canonical spheres, which can then be aligned through the use of feature points. The deformed prostate model can then be restored to the original shape of its match.

In addition to the three basic MR modes explored in this work, there are a number of further modes which are being explored in the medical domain for the identification of prostate cancer. Diffusion-weighted imaging can be used to create apparent diffusion maps, in which cancer will show lower values than normal tissue [36]. Dynamic contrast-enhanced MRI has also been used, where the tumor will have a faster wash-in and wash-out [55]. Further work on preserving the hues of various colors might also be applicable to the custom user painting ability and the 2D slice overlays as more color choices are used [18].

## 7.2.2 Long Term

It would be very helpful for doctors to have the ability to provide diagnostic information from VC to identify cases where OC can be avoided or to identify which polyps are necessary to be removed. Although the electronic biopsy method



**Figure 7.2:** Example of a discrepancy between prone and supine views of a polyp. (a) The prone electronic biopsy view shows significant high density throughout the polyp (red coloration). (b) The supine electronic biopsy view shows much lower density within the center of the polyp (green coloration) and less prevalent high density around the periphery. (c) The polyp is located within a pool of tagged material in the supine, leading the radiologist to believe that the observed discrepancy is the result of subtraction artifacts from the digital cleansing.

has been developed as a method of allowing the user to gain insight into the structure of a polyp, it is not yet ready to act as a primary mode of diagnosing polyp pathology [88]. In looking at the ability of a radiologist to determine whether a polyp was benign or an adenoma, the overall accuracy when using the combined score was only 77%. When looking at the sensitivity and specificity values, much of the error came from false positive results. Indeed, the overall sensitivity is quite good for the total score at 93%. However, the specificity of 43% is extremely low.

Digital cleansing can cause problems with the electronic biopsy function, such as in the case of the polyp shown in Figure 7.2. For this polyp, a score of *definitely an adenoma* was assigned for the prone electronic biopsy and a score of *probably not an adenoma* was assigned for the supine electronic biopsy. The polyp was indeed a tubulovillous adenoma. The radiologist made a note on this case that he did not believe the supine image and that the erroneous electronic biopsy view was probably the result of a subtraction artifact due to the electronic cleansing. While electronic cleansing methods for tagged materials are popular, they can obviously have effects on the electronic biopsy technique, which relies solely on the intensity values from the CT data. More investigation is needed to mitigate this problem.

Knowledge from this and future studies can be used to improve the abilities of a virtual colonoscopy system to provide accurate diagnostic information on the pathology of colonic polyps. In addition to modifying the translucent transfer function used in generating the electronic biopsy images, another possibility is to use a

segmentation of the colon outer boundary [8]. With this, the ray integration through the colon would be limited so as to not reach any structures beyond the colon. The hope is that at a future date the electronic biopsy can be applied as a primary means of diagnosis, allowing for a reduction in the number of patients that are referred for the invasive OC procedure based on VC findings.

Further research work on bringing doctor interaction more into the visualization domain can be looked at for visualization for assistance in treatment planning for prostate cancer. There has been work performed on simulating laser therapy for BPH [166]. Radiotherapy simulations have been performed for the liver [121], and similar techniques would be applicable to the prostate. Simulation of safe methods of entry into the body to reach the prostate could also be of interest [62]. Applying these various methods to the prostate is a further possibility.

## Bibliography

- [1] B. Acar, S. Napel, D. S. Paik, P. Li, J. Yee, R. B. Jeffrey, Jr., and C. Beaulieu. Medial axis registration of supine and prone CT colonography data. *Proc. of Engineering in Medicine and Biology Society (EMBS)*, pages 2433–2436, Oct. 2001.
- [2] M. Ament, D. Weiskopf, and H. Carr. Direct interval volume visualization. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1505–1514, Nov. 2010.
- [3] P. Angelelli and H. Hauser. Straightening tubular flow for side-by-side visualization. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2063–2070, Dec. 2011.
- [4] K. V. Asari, S. Kumar, and D. Radhakrishnan. A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation. *IEEE Trans. on Medical Imaging*, 18(4):345–354, Apr. 1999.
- [5] C. Bachmaier. A radial adaptation of the Sugiyama framework for visualizing hierarchical information. *IEEE Trans. on Visualization and Computer Graphics*, 13(3):583–594, May-June 2007.
- [6] C. Bachmaier, U. Brandes, and B. Schlieper. Drawing phylogenetic trees. *Proc. of International Symposium on Algorithms and Computation*, pages 1110–1121, 2005.
- [7] A. V. Bartrolí, R. Wegenkittl, A. König, and E. Gröller. Nonlinear virtual colon unfolding. *Proc. of IEEE Visualization*, pages 411–420, 2001.
- [8] D. Bartz. Virtual endoscopy in research and clinical practice. *Computer Graphics Forum*, 24(1):111–126, March 2005.
- [9] J. Beyer, M. Hadwiger, S. Wolfsberger, and K. Bühler. High-quality multi-modal volume rendering for preoperative planning of neurosurgical interven-

- tions. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1696–1703, November 2007.
- [10] I. Bitter, A. E. Kaufman, and M. Sato. Penalized-distance volumetric skeleton algorithm. *IEEE Trans. on Visualization and Computer Graphics*, 7(3):195–206, July 2001.
- [11] M. A. Borkin, K. Z. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. J. Rybicki, C. L. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2479–2488, Dec. 2011.
- [12] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sept. 2004.
- [13] R. Bramon, I. Boada, A. Bardera, J. Rodríguez, M. Feixas, J. Puig, and M. Sbert. Multimodal data fusion based on mutual information. *IEEE Trans. on Visualization and Computer Graphics*, 18(9):1574–1587, Sept. 2012.
- [14] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1077–1084, September 2006.
- [15] K. Bürger, J. Krüger, and R. Westermann. Direct volume editing. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1388–1395, Nov. 2008.
- [16] W. Cai and G. Sakas. Data intermixing and multi-volume rendering. *Computer Graphics Forum*, 18(3):359–368, September 1999.
- [17] A. S. Chowdhury, J. Yao, R. Vanuitert, M. Linguraru, and R. Summers. Detection of anatomical landmarks in human colon from computed tomographic colonography images. *Proc. of the 19th International Conference on Pattern Recognition (ICPR)*, pages 1–4, Dec. 2008.
- [18] J. Chuang, D. Weiskopf, and T. Möller. Hue-preserving color blending. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1275–1282, November/December 2009.
- [19] N. D. Cornea, D. Sivler, and P. Min. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. on Visualization and Computer Graphics*, 13(3):530–548, May 2007.

- [20] C. D. Correa and K.-L. Ma. The occlusion spectrum for volume classification and visualization. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1465–1472, November/December 2009.
- [21] CUDA. [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html).
- [22] A. H. de Vries, R. Truyen, J. van der Peijl, J. Florie, R. E. van Gelder, F. Gerritsen, and J. Stoker. Feasibility of automated matching of supine and prone CT-colonography examinations. *British Journal of Radiology*, 79:740–744, Sept. 2006.
- [23] W. De Wever, V. Vandecaveye, S. Lanciotti, and J. Verschakelen. Multidetector CT-generated virtual bronchoscopy: An illustrated review of the potential clinical indications. *European Respiratory Journal*, 23(5):776–782, May 2004.
- [24] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *Proc. of ACM SIGGRAPH*, pages 65–74, 1988.
- [25] K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-Time Volume Graphics*. AK Peters, 2006.
- [26] K. Engel, M. Kraus, and T. Ertl. Interactive high-quality pre-integrated volume rendering using hardware-accelerated graphics hardware. *Proc. of ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 9–16, 2001.
- [27] O. Ersoy, C. Hurter, F. V. Paulovich, G. Cantareira, and A. Telea. Skeleton-based edge bundling for graph visualization. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2364–2373, Dec. 2011.
- [28] M. Floater and K. Hormann. Surface parameterization: A tutorial and survey. *Advances in Multiresolution for Geometric Modelling*, pages 157–186, 2005.
- [29] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.
- [30] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Professional, 1995.
- [31] N. M. Gatto, H. Frucht, V. Sundararajan, J. S. Jacobson, V. R. Grann, and A. I. Neugut. Risk of perforation after colonoscopy and sigmoidoscopy: A population-based study. *Journal of the National Cancer Institute*, 95(3):230–236, 2003.



- [32] S. F. F. Gibson. Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. *Proc. of MICCAI*, pages 888–898, 1998.
- [33] G. J. Grevera and J. K. Udupa. Shape-based interpolation of multi-dimensional grey-level images. *IEEE Trans. on Medical Imaging*, 15(6):881–892, Dec. 1996.
- [34] X. Gu and S.-T. Yau. Global conformal parameterization. *Proc. of Symposium on Geometry Processing*, pages 127–137, 2003.
- [35] X. D. Gu and S.-T. Yau. *Computational Conformal Geometry*. International Press of Boston, 2008.
- [36] M. A. Haider, T. H. van der Kwast, J. Tanguay, A. J. Evans, A. Hashmi, G. Lockwood, and J. Trachtenberg. Combined T2-weighted and diffusion-weighted MRI for localization of prostate cancer. *American Journal of Roentgenology*, 189(2):323–328, Aug. 2007.
- [37] S. Haker, S. Angenent, A. Tannenbaum, and R. Kikinis. Nondistorting flattening maps and the 3-D visualization of colon CT images. *IEEE Trans. on Medical Imaging*, 19(7):665–670, July 2000.
- [38] A. Haritsis, D. Gillies, and C. Williams. Realistic generation and real time animation of images of the human colon. *Computer Graphics Forum*, 11(3):367–379, May 1992.
- [39] R. I. Hartley and S. B. Kang. Parameter-free radial distortion correction with centre of distortion estimation. *Proc. Tenth IEEE International Conference on Computer Vision (ICCV)*, 2:1834–1841, 2005.
- [40] T. He, L. Hong, D. Chen, and Z. Liang. Reliable path for virtual endoscopy: Ensuring complete examination of human organs. *IEEE Trans. on Visualization and Computer Graphics*, 7(4):333–342, Oct.-Dec. 2001.
- [41] C. Heine, D. Schneider, H. Carr, and G. Scheuermann. Drawing contour trees in the plane. *IEEE Trans. on Visualization and Computer Graphics*, 17(11):1599–1611, Nov. 2011.
- [42] G. T. Herman, J. Zheng, and C. A. Bucholtz. Shape-based interpolation. *IEEE Computer Graphics and Applications*, 12(3):69–79, May 1992.
- [43] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. on Visualization and Computer Graphics*, 6(1):24–43, Jan.-March 2000.

- [44] M. R. Hersh, E. L. Knapp, and J. Choi. Newer imaging modalities to assess tumor in the prostate. *Cancer Control*, 11(6):353–357, Nov. 2004.
- [45] W. E. Higgins, K. Ramaswamy, R. D. Swift, G. McLennan, and E. A. Hoffman. Virtual bronchoscopy for three-dimensional pulmonary image assessment: State of the art and future needs. *RadioGraphics*, 18(3):761–778, May 1998.
- [46] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual voyage: Interactive navigation in the human colon. *Proc. of SIGGRAPH*, pages 27–34, 1997.
- [47] W. Hong, X. Gu, F. Qiu, M. Jin, and A. Kaufman. Conformal virtual colon flattening. *Proc. ACM Symposium on Solid and Physical Modeling*, pages 85–93, 2006.
- [48] W. Hong, F. Qiu, and A. Kaufman. A pipeline for computer aided polyp detection. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):861–868, Sept. 2006.
- [49] W. Hong, F. Qiu, J. Marino, and A. Kaufman. Computer-aided detection of colonic polyps using volume rendering. *Proc. of SPIE Medical Imaging*, 6514:651406, Feb. 2007.
- [50] W. Hong, J. Wang, F. Qiu, A. Kaufman, and J. Anderson. Colonoscopy simulation. *Proc. of SPIE Medical Imaging*, 6511:65110R, Feb. 2007.
- [51] Z. Hossain, U. R. Alim, and T. Moller. Toward high-quality gradient estimation on regular lattices. *IEEE Trans. on Visualization and Computer Graphics*, 17(4):426–439, April 2011.
- [52] N. Howlader, A. M. Noone, M. Krapcho, N. Neyman, R. Aminou, S. F. Altekruse, C. L. Kosary, J. Ruhl, Z. Tatalovich, H. Cho, A. Mariotto, M. P. Eisner, D. R. Lewis, H. S. Chen, E. J. Feuer, and K. A. Cronin (eds). SEER cancer statistics review, 1975-2009 (vintage 2009 populations), National Cancer Institute. Bethesda, MD, based on November 2011 SEER data submission, posted to the SEER web site, April 2012. [http://seer.cancer.gov/csr/1975\\_2009\\_pops09/](http://seer.cancer.gov/csr/1975_2009_pops09/).
- [53] A. Huang, D. Roy, M. Franaszek, and R. M. Summers. Teniae coli guided navigation and registration for virtual colonoscopy. *Proc. of IEEE Visualization*, pages 279–285, Oct. 2005.

- [54] A. Huang, D. A. Roy, R. M. Summers, M. Franaszek, N. Petrick, J. R. Choi, and P. J. Pickhardt. Teniae coli-based circumferential localization system for CT colonography: Feasibility study. *Radiology*, 243(2):551–560, May 2007.
- [55] A. S. N. Jackson, S. A. Reinsberg, S. A. Sohaib, E. M. Charles-Edwards, S. Jhavar, T. J. Christmas, A. C. Thompson, M. J. Bailey, C. M. Corbishley, C. Fisher, M. O. Leach, and D. P. Dearnaley. Dynamic contrast-enhanced MRI for prostate cancer localization. *British Journal of Radiology*, 82:148 – 156, 2009.
- [56] A. B. Jani, P. A. S. Johnstone, T. Fox, and C. Pelizzari. Optimization of opacity function for computed tomography volume rendered images of the prostate using magnetic resonance reference volumes. *Intl. Journal of Computer Assisted Radiology and Surgery*, 1(5):285–293, February 2007.
- [57] R. Jiang, J. Marino, W. Zeng, X. Zhao, X. Gu, and A. Kaufman. Registration of volumetric prostate scans using curvature flow. *Technical Report, Computer Science Department*, March 2011.
- [58] M. Jin, J. Kim, and X. D. Gu. Discrete surface Ricci flow: Theory and applications. *Proc. of IMA Conf. on the Mathematics of Surfaces*, pages 209–232, 2007.
- [59] M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface Ricci flow. *IEEE Trans. on Visualization and Computer Graphics*, 14(5):1030–1043, Sept. 2008.
- [60] M. Jin, Y. Wang, S.-T. Yau, and X. Gu. Optimal global conformal surface parameterization. *Proc. of IEEE Visualization*, pages 267–274, 2005.
- [61] L. Kettner. Designing a data structure for polyhedral surfaces. *Proc. of Symposium on Computational Geometry*, pages 146–154, 1998.
- [62] R. Khlebnikov, B. Kainz, J. Muehl, and D. Schmalstieg. Crepuscular rays for tumor accessibility planning. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2163–2172, Dec. 2011.
- [63] O. Kin-Chung Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee. Skeleton extraction by mesh contraction. *ACM Trans. on Graphics*, 27(3):44:1–44:10, August 2008.
- [64] A. P. Kiraly, J. P. Helferty, E. A. Hoffman, G. McLennan, and W. E. Higgins. Three-dimensional path planning for virtual bronchoscopy. *IEEE Trans. on Medical Imaging*, 23(9):1365–1379, Sept. 2004.

- [65] A. Knoll, Y. Hijazi, R. Westerteiger, M. Schott, C. Hansen, and H. Hagen. Volume ray casting with peak finding and differential sampling. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1571–1578, November/December 2009.
- [66] A. Krüger, C. Kubisch, G. Strauß, and B. Preim. Sinus endoscopy - application of advanced GPU volume rendering for virtual endoscopy. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1491–1498, November/December 2008.
- [67] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Proc. of ACM SIGGRAPH*, pages 451–458, July 1994.
- [68] S. Lakare, M. Wan, M. Sato, and A. Kaufman. 3D digital cleansing using segmentation rays. *Proc. of IEEE Visualization*, pages 37–44, Oct. 2000.
- [69] J. Lamy and R. M. Summers. Teniae coli detection from colon surface: Extraction of anatomical markers for virtual colonoscopy. *Proc. of the Third Annual Symposium on Visual Computing*, pages 199–207, 2007.
- [70] Z. Lee, D. B. Sodee, M. Resnick, and G. T. Maclennan. Multimodal and three-dimensional imaging of prostate cancer. *Computerized Medical Imaging and Graphics*, 29:477–486, 2005.
- [71] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.
- [72] M. Levoy. Efficient ray tracing of volume data. *ACM Trans. on Graphics*, 9(3):245–261, July 1990.
- [73] P. Li, S. Napel, B. Acar, D. S. Paik, R. B. Jeffrey, Jr., and C. F. Beaulieu. Registration of central paths and colonic polyps between supine and prone scans in computed tomography colonography: Pilot study. *Medical Physics*, 31(10):2912–23, Oct. 2004.
- [74] W. Li, K. Mueller, and A. Kaufman. Empty space skipping and occlusion clipping for texture-based volume rendering. *Proc. of IEEE Visualization*, pages 317–324, 2003.
- [75] X. Li, Y. Bao, X. Guo, M. Jin, X. Gu, and H. Qin. Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):805–819, 2008.

- [76] Z. Liang, C. Duan, D. Gu, and H. Lu. Extracting the inner and outer borders of bladder wall and flattening the extracted wall for MR cystography. *Proc. of World Congress on Medical Physics and Biomedical Engineering*, 25:322–325, 2009.
- [77] B. Lichtenbelt, R. Crane, and S. Naqvi. *Introduction to Volume Rendering*. Hewlett-Packard Professional Books, Prentice Hall, 1998.
- [78] S. Lim, H.-J. Lee, and B.-S. Shin. Surface reconstruction for efficient colon unfolding. *Proc. of Geometric Modeling and Processing*, pages 623–629, 2006.
- [79] P. Ljung, C. Winskog, A. Persson, C. Lundström, and A. Ynnerman. Full body virtual autopsies using a state-of-the-art volume rendering pipeline. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):869–876, September/October 2006.
- [80] P. Lo, B. van Ginneken, J. Reinhardt, and M. de Bruijne. Extraction of airways from CT (EXACT’09). *Proc. of International Workshop on Pulmonary Image Analysis*, pages 175–189, 2009.
- [81] T. Loch. Urologic imaging for localized prostate cancer in 2007. *World Journal of Urology*, 25:121–129, 2007.
- [82] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, August 1987.
- [83] C. Lundström, T. Rydell, C. Forsell, A. Persson, and A. Ynnerman. Multi-touch table system for medical visualization: Application to orthopedic surgery planning. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1775–1784, December 2011.
- [84] L. Ma, Y. Chen, and K. L. Moore. Flexible camera calibration using a new analytical radial undistortion formula with application to mobile robot localization. *Proc. IEEE International Symposium on Intelligent Control*, pages 799–804, Oct. 2003.
- [85] P. J. MacMahon, A.-M. Kennedy, D. T. Murphy, M. Maher, and M. M. McNicholas. Modified prostate volume algorithm improves transrectal US volume estimation in men presenting for prostate brachytherapy. *Radiology*, 250(1):273–280, January 2009.

- [86] A. Madabhushi, J. Shi, M. D. Feldman, M. Rosen, and J. Tomaszewski. Comparing ensembles of learners: Detection prostate cancer from high resolution MRI. *Proc. of Computer Vision Approaches to Medical Image Analysis*, pages 25–26, 2006.
- [87] J. Mallon and P. F. Whelan. Precise radial un-distortion of images. *Proc. 17th International Conference on Pattern Recognition (ICPR)*, pages 18–21, Aug. 2004.
- [88] J. Marino, W. Du, M. Barish, E. Li, W. Zhu, and A. Kaufman. Evaluation of electronic biopsy for clinical diagnosis in virtual colonoscopy. *Proc. of SPIE Medical Imaging*, 7964:796419, Feb. 2011.
- [89] J. Marino and A. Kaufman. Colon visualization using shape preserving flattening. *Proc. of MICCAI Workshop on Virtual Colonoscopy & Abdominal Imaging*, pages 155–159, Sept. 2010.
- [90] J. Marino and A. Kaufman. Upsampling of prostate volumes. *Proc. of CE-WIT International Conference*, Sept. 2010.
- [91] J. Marino and A. Kaufman. Prostate cancer visualization from MR imagery and MR spectroscopy. *Computer Graphics Forum*, 30(3):1051–1060, June 2011.
- [92] J. Marino and A. Kaufman. Planar visualization of treelike objects. *IEEE Trans. on Visualization and Computer Graphics*, Submitted for publication, 2012.
- [93] J. Marino and A. Kaufman. Prostate visualization with overlapping regions of interest. *IEEE Pacific Visualization*, Poster, Feb. 2012.
- [94] J. Marino, F. Qiu, and A. Kaufman. Co-registration of virtual and optical colonoscopy views. *Proc. of MICCAI Workshop on Virtual Colonoscopy*, pages 122–126, Sept. 2008.
- [95] J. Marino, F. Qiu, and A. Kaufman. Virtually assisted optical colonoscopy. *Proc. of SPIE Medical Imaging*, 6916:69160J, Feb. 2008.
- [96] J. Marino, W. Zeng, X. Gu, and A. Kaufman. Context preserving maps of tubular structures. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1997–2004, Dec. 2011.

- [97] K. Marriott, P. Sbarski, T. van Gelder, D. Prager, and A. Bulka. Hi-trees and their layout. *IEEE Trans. on Visualization and Computer Graphics*, 17(3):290–304, March 2011.
- [98] N. Max. Optical models for direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [99] M. Mirzargar and A. Entezari. Quasi interpolation with Voronoi splines. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1832–1841, Dec. 2011.
- [100] L. Mroz, H. Hauser, and E. Gröller. Interactive high-quality maximum intensity projection. *Computer Graphics Forum*, 19(3):341–350, 2000.
- [101] D. Nain, S. Haker, W. E. L. Grimson, E. Cosman Jr., W. W. Wells, H. Ji, R. Kikinis, and C. F. Westin. Intra-patient prone to supine colon registration for synchronized virtual colonoscopy. *Proc. of MICCAI*, pages 573–580, 2002.
- [102] J. Näppi, A. Okamura, H. Frimmel, A. Dachman, and H. Yoshida. Region-based supine-prone correspondence of false-positive CAD polyp candidates in CT colonography. *Academic Radiology*, 12(6):695–707, 2005.
- [103] A. Neubauer, S. Wolfsberger, M.-T. Forster, L. Mroz, A. Wegenkittl, and K. Bühler. Advanced virtual endoscopic pituitary surgery. *IEEE Trans. on Visualization and Computer Graphics*, 11(5):497–507, September/October 2005.
- [104] OpenCL. <http://www.khronos.org/ocl/>.
- [105] D. M. Parkin, F. Bray, J. Ferlay, and P. Pisani. Global cancer statistics, 2002. *CA: A Cancer Journal for Clinicians*, 55(2):74–108, March/April 2005.
- [106] D. Pasquier, T. Lacornerie, M. Vermandel, J. Rousseau, E. Lartigau, and N. Betrouni. Automatic segmentation of pelvic structures from magnetic resonance images for prostate cancer radiotherapy. *International Journal of Radiation Oncology, Biology, Physics*, 68(2):592–600, June 2007.
- [107] D. Perchet, C. I. Fetita, and F. J. Preteux. Advanced navigation tools for virtual bronchoscopy. *Proc. of SPIE Image Processing*, pages 147–158, May 2004.

- [108] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The VolumePro real-time ray-casting system. *Proc. of ACM SIGGRAPH*, pages 251–260, 1999.
- [109] H. Pfister and A. Kaufman. Cube-4 - A scalable architecture for real-time volume rendering. *Proc. of Symposium on Volume Visualization*, pages 47–54, 1996.
- [110] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. Avila, K. Martin, R. Machiraju, and J. Lee. The transfer function bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–22, May 2001.
- [111] P. J. Pickhardt, J. R. Choi, I. Hwang, J. A. Butler, M. L. Puckett, H. A. Hildebrandt, R. K. Wong, P. A. Nugent, P. A. Mysliwiec, and W. R. Schindler. Computed tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults. *The New England Journal of Medicine*, 349(23):2191–2200, Dec. 2003.
- [112] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer, 1996.
- [113] J. Plate, T. Holtkaemper, and B. Froehlich. A flexible multi-volume shader framework for arbitrarily intersecting multi-resolution datasets. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1584–1591, November 2007.
- [114] W. Plishker and R. Shekhar. Virtual colonoscopy registration regularization with global chainmil. *Proc. of MICCAI Workshop on Virtual Colonoscopy*, pages 116–121, 2008.
- [115] J.-S. Praßni, T. Ropinski, J. Mensmann, and K. Hinrichs. Shape-based transfer functions for volume visualization. *Proc. of Pacific Visualization Symposium*, pages 9–16, 2010.
- [116] F. Qiu, J. Marino, , and A. Kaufman. Computer aided polyp detection with texture analysis. *Proc. of MICCAI Workshop on Virtual Colonoscopy*, pages 148–152, Sept. 2008.
- [117] L. Raghupathi, L. Grisoni, F. Faure, D. Marchal, M.-P. Cani, and C. Chailou. An intestinal surgery simulator: Real-time collision processing and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):708–718, Nov. 2004.



- [118] S. P. Raya and J. K. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Trans. on Medical Imaging*, 9(1):32–42, March 1990.
- [119] D. Reniers, J. J. van Wijk, and A. Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Trans. on Visualization and Computer Graphics*, 14(2):355–368, March 2008.
- [120] C. Rezk-Salama and A. Kolb. Opacity peeling for direct volume rendering. *Computer Graphics Forum*, 25(3):597–606, 2006.
- [121] C. Rieder, T. Krüger, C. Schumann, and H. K. Hahn. GPU-based real-time approximation of the ablation zone for radiofrequency ablation. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1812–1821, Dec. 2011.
- [122] C. Rieder, A. Weihusen, C. Schumann, S. Zidowitz, and H.-O. Peitgen. Visual support for interactive post-interventional assessment of radiofrequency ablation therapy. *Computer Graphics Forum*, 29(3):1093–1102, 2010.
- [123] R. A. Robb. Three-dimensional visualization and analysis in prostate cancer. *Drugs of Today*, 38(3):153–165, 2002.
- [124] T. Ropinski, S. Hermann, R. Reich, M. Schafers, and K. Hinrichs. Multimodal vessel visualization of mouse aorta PET/CT scans. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1515–1522, November/December 2009.
- [125] F. Röbler, R. P. Botchen, and T. Ertl. Dynamic shader generation for GPU-based multi-volume ray casting. *IEEE Computer Graphics and Applications*, 28(5):66–77, September 2008.
- [126] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, 1995.
- [127] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171, Jan. 2006.
- [128] P. Simonetto, D. Archambault, D. Auber, and R. Bourqui. ImPrEd: An improved force-directed algorithm that prevents nodes from crossing edges. *Computer Graphics Forum*, 30(3):1071–1080, June 2011.

- [129] M. Smelyanskiy, D. Holmes, J. Chhugani, A. Larson, D. M. Carmean, D. Hanson, P. Dubey, K. Augustine, D. Kim, A. Kyker, V. W. Lee, A. D. Nguyen, L. Seiler, and R. Robb. Mapping high-fidelity volume rendering for medical imaging to CPU, GPU and many-core architectures. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1563–1570, November/December 2009.
- [130] B. Springborn, P. Schröder, and U. Pinkall. Conformal equivalence of triangle meshes. *ACM Trans. on Graphics*, 27(3):77:1–77:11, August 2008.
- [131] S. Stegmaier, M. Strengert, T. Klein, and T. Ertl. A simple and flexible volume rendering framework for graphics-hardware-based raycasting. *Proc. Volume Graphics*, pages 187–195, 2005.
- [132] J. W. Suh and C. L. Wyatt. Deformable registration of supine and prone colons for computed tomographic colonography. *Journal of Computer Assisted Tomography*, 33(6):902–911, Nov. 2009.
- [133] R. M. Summers. Navigation aids for real-time virtual bronchoscopy. *American Journal of Roentgenology*, 168(5):1165–1170, May 1997.
- [134] R. M. Summers, C. D. Johnson, L. M. Pusanik, J. D. Malley, A. M. Youssef, and J. E. Reed. Automated polyp detection at CT colonography: Feasibility assessment in a human population. *Radiology*, 219(1):51–59, 2001.
- [135] A. Tagliasacchi, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. *ACM Trans. on Graphics*, 28(3):71:1–71:9, August 2009.
- [136] P. Tiwari, A. Madabhushi, and M. Rosen. A hierarchical unsupervised spectral clustering scheme for detection of prostate cancer from magnetic resonance spectroscopy (MRS). *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention*, 10 (Pt 2):278–286, 2007.
- [137] J. Torres and J. M. Menéndez. A practical algorithm to correct geometrical distortion of image acquisition cameras. *International Conference on Image Processing (ICIP)*, pages 2451–2454, 2004.
- [138] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. *Proc. of European Conference on Computer Vision (ECCV)*, pages 596–609, 2008.

- [139] R. V. Uitert, I. Bitter, and R. Summers. Detection of colon wall outer boundary and segmentation of the colon wall based on level set methods. *Proc. of IEEE Engineering in Medicine and Biology Society*, pages 3017–3020, 2006.
- [140] Y. Umemoto, M. Oda, T. Kitasaka, K. Mori, Y. Hayashi, Y. Suenaga, T. Takayama, and H. Natori. Extraction of teniae coli from CT volumes for assisting virtual colonoscopy. *Proc. of SPIE Medical Imaging*, 6916:69160D, 2008.
- [141] U.S. Preventive Services Task Force. Screening for colorectal cancer: U.S. preventive services task force recommendation statement. *Annals of Internal Medicine*, 149(9):627–637, Nov. 2008.
- [142] R. L. Van Uitert and R. M. Summers. Automatic correction of level set based subvoxel precise centerlines for virtual colonoscopy using the colon outer wall. *IEEE Trans. on Medical Imaging*, 26(8):1069–1078, Aug. 2007.
- [143] F. P. Vidal, F. Bello, K. W. Brodlie, N. W. John, D. Gould, R. Phillips, and N. J. Avis. Principles and applications of computer graphics in medicine. *Computer Graphics Forum*, 25(1):113–137, 2006.
- [144] I. Viola, A. Kanitsar, and M. E. Gröller. Importance-driven volume rendering. *Proc. IEEE Visualization*, pages 139–145, 2004.
- [145] M. Wan, F. Dachilles, K. Kreeger, S. Lakare, M. Sato, A. Kaufman, M. Wax, and Z. Liang. Interactive electronic biopsy for 3D virtual colonoscopy. *Proc. SPIE Medical Imaging*, pages 483–488, 2001.
- [146] M. Wan, Z. Liang, Q. Ke, L. Hong, I. Bitter, and A. Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE Trans. on Medical Imaging*, 21(12):1450–1460, Dec. 2002.
- [147] S. Wang, H. Cohen, S. Mankes, L. Li, J. J. Chen, and Z. Liang. An EM approach to MAP solution of segmenting tissue mixture percentages with application to CT-based virtual colonoscopy. *Medical Physics*, 35(12):5787–5798, 2008.
- [148] Y.-S. Wang and T.-Y. Lee. Curve-skeleton extraction using iterative least squares optimization. *IEEE Trans. on Visualization and Computer Graphics*, 14(4):926–936, July 2008.
- [149] Z. Wang, Z. Liang, L. Li, B. Li, D. Eremina, and H. Lu. An improved electronic colon cleansing method for detection of colonic polyps by virtual

- colonoscopy. *IEEE Transactions on Biomedical Engineering*, 53(8):1635–46, Aug. 2006.
- [150] J. C. Weinreb, J. D. Blume, F. V. Coakley, T. M. Wheeler, J. B. Cormack, C. K. Sotito, H. Cho, A. Kawashima, C. M. Tempany-Afdhal, K. J. Macura, M. Rosen, S. R. Gerst, and J. Kurhanewicz. Sextant localization at MR imaging and MR spectroscopic imaging before prostatectomy - results of ACRIN prospective multi-institutional clinicopathologic study. *Radiology*, 251(1):122–133, April 2009.
- [151] L. Westover. Footprint evaluation for volume rendering. *Proc. of ACM SIGGRAPH*, 24:367–376, Sept. 1990.
- [152] A. C. Westphalen, F. V. Coakley, A. Qayyum, M. Swanson, J. P. Simko, Y. Lu, S. Zhao, P. R. Carroll, B. M. Yeh, and J. Kurhanewicz. Peripheral zone prostate cancer: Accuracy of different interpretative approaches with MR and MR spectroscopic imaging. *Radiology*, 246(1):177–184, January 2008.
- [153] D. Williams, S. Grimm, E. Coto, A. Roudsari, and H. Hatzakis. Volumetric curved planar reformation for virtual endoscopy. *IEEE Trans. on Visualization and Computer Graphics*, 14(1):109–119, Jan. 2008.
- [154] Y. Wu, S.-X. Zhang, N. Luo, M.-G. Qiu, L.-W. Tan, Q.-Y. Li, G.-J. Liu, and K. Li. Creation of the digital three-dimensional model of the prostate and its adjacent structures based on Chinese visible human. *Surgical and Radiologic Anatomy*, 32:629–635, 2010.
- [155] D. Xiang, J. Tian, F. Yang, Q. Yang, X. Zhang, Q. Li, and X. Liu. Skeleton cuts - an efficient segmentation method for volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 17(9):1295–1306, Sept. 2011.
- [156] H.-B. Yan, S.-M. Hu, R. R. Martin, and Y.-L. Yang. Shape deformation using a skeleton to drive simplex transformations. *IEEE Trans. on Visualization and Computer Graphics*, 14(3):693–706, May 2008.
- [157] Y.-L. Yang, R. Guo, F. Luo, S.-M. Hu, and X. Gu. Generalized discrete Ricci flow. *Computer Graphics Forum*, 28(7):2005–2014, Oct. 2009.
- [158] Y.-L. Yang, J. Kim, F. Luo, S.-M. Hu, and X. Gu. Optimal surface parameterization using inverse curvature map. *IEEE Trans. on Visualization and Computer Graphics*, 14(5):1054–1066, Sept. 2008.

- [159] H. Yoshida and J. Näppi. Three-dimensional computer-aided diagnosis scheme for detection of colonic polyps. *IEEE Trans. on Medical Imaging*, 20(12):1261–1274, 2001.
- [160] W. Zeng, F. Luo, S.-T. Yau, and X. D. Gu. Surface quasi-conformal mapping by solving beltrami equations. *Proc. of the 13th IMA International Conference on Mathematics of Surfaces*, pages 391–408, 2009.
- [161] W. Zeng, J. Marino, X. Gu, and A. Kaufman. Conformal geometry based supine and prone colon registration. *Proc. of MICCAI Workshop on Virtual Colonoscopy & Abdominal Imaging*, pages 149–154, Sept. 2010.
- [162] W. Zeng, J. Marino, K. C. Gurijala, X. Gu, and A. Kaufman. Supine and prone colon registration using quasi-conformal mapping. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1348–1357, Nov. 2010.
- [163] W. Zeng, J. Marino, A. Kaufman, and X. Gu. Volumetric colon wall flattening using harmonic differentials. *Computers & Graphics*, 35(3):726–732, June 2011.
- [164] W. Zeng, D. Samaras, and X. D. Gu. Ricci flow for 3D shape analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(4):662–677, April 2010.
- [165] N. Zhang, W. Hong, and A. Kaufman. Dual contouring with topology-preserving simplification using enhanced cell representation. *Proc. of IEEE Visualization*, page 505512, Oct. 2004.
- [166] N. Zhang, X. Zhou, Y. Shen, and R. Sweet. Volumetric modeling in laser BPH therapy simulation. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1405–1412, Nov. 2010.
- [167] L. Zhao, C. P. Botha, J. O. Bescos, R. Truyen, F. M. Vos, and F. H. Post. Lines of curvature for polyp detection in virtual colonoscopy. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):885–892, September/October 2006.
- [168] Y. Zhou and A. W. Toga. Efficient skeletonization of volumetric objects. *IEEE Trans. on Visualization and Computer Graphics*, 5(3):196–209, July-September 1999.
- [169] L. Zhu, S. Haker, and A. Tannenbaum. Flattening maps for the visualization of multibranched vessels. *IEEE Trans. on Medical Imaging*, 24(2):191–198, Feb. 2005.

- [170] Y. Zhu, S. Williams, and R. Zwiggelaar. Computer technology in detection and staging of prostate carcinoma: A review. *Medical Image Analysis*, 10(2):178–199, April 2006.
- [171] Y. Zhu, R. Zwiggelaar, and S. Williams. Prostate segmentation: A comparative study. *Proc. of Medical Image Understanding and Analysis*, pages 129–132, 2003.
- [172] R. Zwiggelaar, Y. Zhu, and S. Williams. Semi-automatic segmentation of the prostate. *Proc. of Iberian Conference on Pattern Recognition and Image*, pages 1108–1116, 2003.