

**VARIABLE SCALE STATISTICS FOR CARDIAC
SEGMENTATION AND SHAPE ANALYSIS**

by

Constantine Aaron Cois

Bachelor of Science, University of Pittsburgh, 2003

Master of Science, University of Pittsburgh, 2006

Submitted to the Graduate Faculty of
The School of Engineering in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2007

UNIVERSITY OF PITTSBURGH
SCHOOL OF ENGINEERING

This dissertation was presented

by

Constantine Aaron Cois

It was defended on

October 12th 2007

and approved by

George Stetten, Professor, Department of Bioengineering

J. R. Boston, Professor, Department of Bioengineering

Mei Chen, Research Scientist, Intel Research, Pittsburgh

C. C. Li, Professor, Department of Electrical and Computer Engineering

Michael Sacks, Professor, Department of Bioengineering

Dissertation Director: George Stetten, Professor, Department of Bioengineering

Copyright © by Constantine Aaron Cois
2007

VARIABLE SCALE STATISTICS FOR CARDIAC SEGMENTATION AND SHAPE ANALYSIS

Constantine Aaron Cois, PhD

University of Pittsburgh, 2007

A novel framework for medical image analysis, known as *Shells and Spheres*, has been developed by our research lab. This framework utilizes spherical operators of variable radius, centered at each image pixel and sized to reach, but not cross, the nearest boundary. Statistical population tests are performed on the populations of pixels within adjacent spheres to compare image regions across boundaries, delineating both independent image objects and the boundaries between them. This research has focused on developing the Shells and Spheres framework and applying it to the problem of segmentation of anatomical objects. Furthermore, we have rigorously studied the framework and its applications to clinical segmentation, validating and improving our n -dimensional segmentation algorithm. To this end, we have enhanced the original Shells and Spheres segmentation algorithm by adding *a priori* information, developing techniques for optimizing algorithm parameters, implementing a software platform for experimentation, and performing validation experiments using real 3D ovine cardiac MRI data. The system developed provides automated 3D segmentation given *a priori* information in the form of a trivial 2D manual training procedure, which involves tracing a single 2D contour from which 3D algorithm parameters are then automatically derived. We apply this system to segmentation of the Right Ventricular Outflow Tract (RVOT) to aid in research toward the creation of a Tissue Engineered Pulmonary Valve (TEPV). Experimental methods are presented for the development and validation of the system, as well as a detailed description of the Shells and Spheres framework, our segmentation algorithm, and the clinical significance of this work.

Keywords: Image Analysis, Medical Image Analysis, Image Segmentation, Medical Imaging, Computer Vision, Cardiovascular Imaging, Shape Modeling.

TABLE OF CONTENTS

1.0 INTRODUCTION	1
1.1 Goals and Motivation	3
1.2 Description of Chapters	7
2.0 BACKGROUND AND SIGNIFICANCE	9
2.1 Tissue Engineered Pulmonary Valves	9
2.2 Medical Image Segmentation	10
2.2.1 Fuzzy-Connectedness	13
2.3 Clinical Image Analysis	15
3.0 THE SHELLS AND SPHERES FRAMEWORK	17
3.1 Framework	18
3.1.1 Shells and Spheres	18
3.1.2 Variable Scale Statistics	21
3.1.2.1 Primary Statistics.	23
3.1.2.2 Secondary Statistics.	24
3.1.2.3 Validation of Population Testing	26
3.2 Noiseless Sphere Map Optimization Algorithm	33
3.3 Real Image Segmentation Algorithm	36
3.3.1 Step 1: VSS Gradient-Based Radius Approximation	37
3.3.2 Step 2: Variance-Constrained Radius Reduction	38
3.3.3 Step 3: Outpost Selection and Exclusion	40
3.3.4 Step 4: Variance-Constrained Scale Growth	45
3.3.5 Step 5: Medial Pixel Identification	46

3.3.6	Step 6: Medial Flood-Fill Segmentation	48
3.4	Discussion	49
4.0	ALGORITHM PARAMETER OPTIMIZATION	51
4.1	2D Optimization for 3D Segmentation	52
4.1.1	Preliminary Experimental Validation	55
4.1.2	Automating Parameter Optimization	60
4.1.2.1	Determining Optimization Ranges	60
4.1.2.2	Seed Point Extraction	65
4.2	2D Slice Selection	67
4.2.1	Experimental Setup	69
4.2.2	Validation	70
4.3	Alternative Parameter Calculation Techniques	74
4.3.1	Alternative Method A	75
4.3.2	Alternative Method B	76
4.3.3	Alternative Method C	78
4.3.4	Alternative Method D	78
4.4	Discussion	80
5.0	VALIDATION RESULTS	81
5.1	3D Manual Segmentations	81
5.1.1	Standard 2D Parameter Optimization Technique	86
5.1.2	Alternative Parameter Calculation Techniques	92
5.2	Error Visualization	100
5.3	Validation Against Other Techniques	106
5.3.1	Supervision	106
5.3.2	Experimental Methods	108
5.3.3	Results	109
6.0	CONCLUSIONS AND FUTURE WORK	112
6.1	Clinically Accessible Systems	112
6.2	Validation of the Shells and Spheres Framework for Image Segmentation	114

6.3	Using Segmentations of the Right Ventricular Outflow Tract for Mechanical Modeling	115
6.4	Future Work	116
6.4.1	Algorithm Improvements	116
6.4.2	Further Research Studies	119
6.4.3	Design of a New Segmentation Algorithm	120
BIBLIOGRAPHY	132

LIST OF TABLES

1	Table of DSC values comparing segmentations produced by 3 independent subjects and our Shells and Spheres (S&S) algorithm.	59
2	Parameters necessary to perform algorithm parameter optimization, along with descriptions.	61
3	Percentage of data sets observing the optimal value of α_s as the value of δ is increased.	65
4	Optimal threshold values α_s and α_g for each data set, calculated using full parameter optimization method.	77

LIST OF FIGURES

1	The Blum medial manifold (dotted line) of a rectangle, with selected medial spheres shown.	18
2	A: 3D contrast-enhanced CT scan of the aortic arch (top left), and the same scan with an overlaid 3D segmentation (bottom left) achieved using the Shells and Spheres framework. B: Surface model of a Shells and Spheres segmentation of the right heart with labeled Right Ventricle (RV) and Right Ventricular Outflow Tract (RVOT), shown from three different perspectives.	19
3	Each pixel is shown as a number indicating its integer distance from the central pixel. If we denote the central pixel as \mathbf{x} , then pixels labeled n are members of the set $H_n(\mathbf{x})$. For example, the pixels labeled “3” (shown in bold) comprise the set $H_3(\mathbf{x})$	21
4	Noiseless image with boundary between two objects. Correctly scaled spheres $S_r(\mathbf{x})$ with $r(\mathbf{x}) = 3$ and $S_r(\mathbf{y})$ with $r(\mathbf{y}) = 2$ touch, but do not cross, the boundary. Numbers indicate pixel intensity.	22
5	Correctly optimized sphere map of the image in Fig. 4. Numbers indicate the integer radius of the sphere at each pixel. Pixels \mathbf{x} and \mathbf{y} are labeled as before.	22
6	A: A synthetic noiseless image. B: A corresponding vertical gradient image, calculated using the VSS gradient. Note that there are no edge effects, such as those seen with conventional gradient calculations.	24
7	The $S^{-1}(\mathbf{x})$ set of spheres that contain pixel \mathbf{x} , adjacent to the boundary between two noiseless objects with respective intensities of 1 and 9.	25

8	Image with noise. Pixel \mathbf{x} is deterred from extending its sphere across the boundary because its mean is an outlier in the population $S^{-1}(\mathbf{y})$	26
9	Left: 2D CT image of the Aortic Arch (AA) used in preliminary testing. Right: 2D AA image with manual tracing highlighted (pink).	27
10	Plot of the normality, measured as the p-value of a SW test, of two randomly selected spheres for radii 5-15.	29
11	Histogram of the radius of maximum normality of all spheres with a gold standard radius of 11 in our test image.	30
12	Histogram of the radius of maximum normality, normalized by statistical significance of the normality measurement, of all spheres with a gold standard radius of 11 in our test image.	31
13	Plot of the mean SW normality of all spheres at each radius, from 5 to 15, with the maximum mean normality marked in red.	32
14	A: An example noiseless image. B: The sphere map created by the Noiseless Algorithm, represented as an intensity image where each pixel \mathbf{x} has an intensity value equivalent to the radius of its sphere $S(\mathbf{x})$	34
15	A: A 2D slice from one of our 3D ovine MRI data sets. B: The same 2D slice with a manual tracing overlaid (pink). C: The manual tracing shown as a binary mask. D: The gold standard sphere map produced by running the Noiseless Algorithm on the segment mask, growing spheres only at pixels within the segment.	35
16	Our synthetic test image from Figure 14A with gaussian noise added.	37
17	A colored height map of the internal variance of all spheres from Fig. 16 after algorithm Step 1, with the variance of each sphere assigned to its central pixel.	39
18	A colored height map of the internal variance of all spheres from Fig. 16 after algorithm Step 2, with the variance of each sphere assigned to its central pixel.	41
19	Illustration of $K^{-1}(\mathbf{x})$ containing 7 pixels (bold), each of whose sphere would place its reflector across the boundary at \mathbf{x}	42

20	Left: Diagram of an object with intensity 1 between two regions of intensity 9, showing set $S^{-1}(\mathbf{b})$ of pixels (bold) whose spheres contain pixel \mathbf{b} . This set produces an $\mathbf{s}(\mathbf{b})$ vector (see text) along which the furthest bold pixel \mathbf{m} is the center of a medial sphere (circle in bold) touching both boundaries (dashed lines). Right: Image of a 2D slice through a CT scan of the aorta with contrast showing an actual $S^{-1}(\mathbf{b})$ set (purple/grey), the resulting $\mathbf{s}(\mathbf{b})$ vector, and the medial manifold (dashed curve) on which the furthest sphere along $\mathbf{s}(\mathbf{b})$ must lie.	47
21	A graph showing the normalized DSC, Average Symmetric Surface Distance, and Percent Volumetric Overlap for ninety automated segmentations compared to manual gold standards.	53
22	Example DSC curve produced by a 2D parameter optimization with $\beta_s \in 0.0 : 6.0$, $\beta_g \in 0.0 : 1.9$, $i = 0.1$. The optimal DSC value 0.96, produced by the parameter set ($\beta_s = 2.1$, $\beta_g = 0.7$), is indicated with an arrow.	54
23	(a) 2D slice of MR cardiac data and (b) manual segmentation (red or dark) used for parameter optimization in the 3D segmentation study.	56
24	DSC curve produced by a 2D parameter optimization with $\beta_s \in 1.0 : 3.5$, $\beta_g \in -0.5 : 1.0$, $i = 0.1$. The optimal DSC, 0.97, is indicated with an arrow at values $\beta_s = 1.9$, $\beta_g = -0.2$)	57
25	Surface model of the automated segmentation of the right heart with the RV and RVOT labeled, shown from three different perspectives.	58
26	Top row: The MR data set with one plane moved incrementally backwards through the RV. Bottom row: Images corresponding to those above, with the segmentation surface model overlaid on the source data.	59
27	Example 2D slice extracted from a 3D MRI data set. This slice, along with a manual 2D tracing of the RVOT within it, is used to optimize parameters and provide seed points for a 3D segmentation of the RVOT in the full volume.	66
28	Left: Example 2D slice with manual tracing (red) overlaid. Right: Example 2D slice with manual tracing (red) and medial sphere used to extract seed point (green) overlaid.	67

29	An example of the three slice orientations used for initialization in the study presented. The RVOT (red, outlined) has been manually traced in each slice.	68
30	A bar graph showing the mean agreement to the 2D initialization tracing achieved using each of the three slice orientations for parameter optimization, indicating how well the system was able to match the training data. Error bars indicate the standard error of the mean.	71
31	The mean agreement of 3D automated segmentations to our expert 3D segmentations produced using the optimal parameter values calculated by brute force parameter optimization. Error bars indicate the standard error of the mean.)	72
32	A box plot of the 3D segmentation accuracy yielded by each of the initialization slice orientations over all data sets. Inter-quartile range (IQR) is bounded by the first and third quartiles, and the range from minimum to maximum recorded DSC value is indicated by the vertical lines, with the minimum and maximum values themselves indicated by pink and blue squares, respectively.	73
33	An example of optimization parameter space.	75
34	An example axial bounding plane, taken from Data Set 1. A: Bounding plane slice. B: Subsequent slice, moving outside our defined RVOT, where the RV merges with the RA. (See text)	82
35	An example sagittal bounding plane, taken from Data Set 1. A: Bounding plane slice. B: Subsequent slice, moving outside our defined RVOT, where the PA branches.	83
36	A surface rendering of a bounded RVOT segmentation, with the axial and sagittal bounding planes labeled.	84
37	Bar graph of the Inter-rater reliability for our expert manual segmentations, calculated as the mean DSC between manual segmentations for each data set.	85
38	Example DSC curve produced by a 2D parameter optimization The optimal DSC value 0.96 found during this optimization procedure is indicated with an arrow.	86

39	Bar graph of mean DSC of S&S automated 3D segmentations created using brute force 2D parameter optimization with the axial initialization slice, when compared to our three expert manual segmentations, for each of our MRI data sets.	87
40	Bar graph showing the inter-rater reliability of sets containing only the expert manual segmentations (yellow) and the expert manual segmentations along with the automated S&S segmentation (blue) for comparison.	88
41	Bar graph showing the difference between the inter-rater reliability measurements of the set of expert manual segmentations with and without the addition of the automated S&S segmentation.	89
42	Bar graph of the average (blue) and minimum (maroon) times of 3D segmentation, including parameter optimization for each data set. Average and minimum values are in the context of the three initialization slices tested. . .	90
43	Bar graph of the average time, split into parameter optimization (dark) and algorithm run time (light) for each data set.	91
44	Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-A, using each of the three 2D initialization slices.	93
45	Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-B, using each of the three 2D initialization slices.	94
46	Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-C, using each of the three 2D initialization slices.	95
47	Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-D, using each of the three 2D initialization slices.	96
48	Box plot of the 3D segmentation accuracy, measured by DSC, for each of the parameter calculation methods. Data includes automated 3D segmentations of all ten MRI data sets, using each of the three initialization slices, compared to each of the three expert manual segmentations. Q1 and Q3 represent the first and third quartiles of the DSC data, respectively.	97

49	Box plot of the computation time of 3D segmentation, measured in minutes, for each of the parameter calculation methods. Data includes automated 3D segmentations of all ten MRI data sets, using each of the three initialization slices, compared to each of the three expert manual segmentations. Q1 and Q3 represent the first and third quartiles of the time data, respectively. . . .	98
50	Screenshot of ITK-SNAP displaying an Error Sphere Map (ESM) overlaid on the original image data. The object segmented was an RVOT, bounded at pre-determined cutoff planes. the lower left quadrant displays a surface rendering of the ESM.	102
51	A: Surface rendering of the ESM of an RVOT. B: The same RVOT ESM, with spheres within 1 radius of the gold standard removed. C: The same RVOT ESM, with spheres within 2 radii of the gold standard removed. D: The same RVOT ESM, with spheres within 3 radii of the gold standard removed. . . .	104
52	An example feature image $g(I)$ created via manual parameter adjustment using ITK-SNAP. The RVOT is labeled. (red)	108
53	A slice of the grayscale MRI image used to create the feature image in Figure 52, with the seed bubble (green) for the active contour placed in the RVOT.	109
54	Bar graph showing the mean 3D segmentation agreement to all expert segmentations over all data sets for both the active contour method (blue) and our S&S method (maroon). Error bars show the standard error of the mean (SEM).	110
55	Bar graph showing the individual 3D segmentation agreement to expert segmentations for each data set for both the active contour method (blue) and our S&S method (maroon).	111
56	A truncated wedge (light) extending from the surface of a sphere (dark) to the center.	122

PREFACE

‘Education is not the filling of a pail, but the lighting of a fire.’ William Butler Yeats

The research in this dissertation marks a fundamentally new approach to statistical medical image analysis, and presents a vision of intelligent systems truly designed for clinical use. I offer this work to fuel the further development of clinically accessible automated image analysis systems, allowing state of the art technology to be routinely employed to improve patient care.

I’d like to thank my family, friends, and colleagues for all of their support throughout my education. A special thanks goes to the members of VIALab, past and present, that were involved in the development and testing of Shells and Spheres: Robert Tamburo, John Galeotti, Ken Rockot, David Weiser, and especially my advisor and mentor George Stetten. Dr. Stetten, George, thank you for the support, motivation, and guidance through the years.

Most of all, I’d like to dedicate this work to my wife, Kimberly, on whose love and support I can always depend.

1.0 INTRODUCTION

Medical images provide clinicians with a wide variety of in-situ structural and functional data in a non-invasive manner, and have thus become an extremely powerful clinical tool. The ability to visualize the shape and location of anatomical structures non-invasively is invaluable in diagnosis, surgical planning, and post-treatment evaluation, as well as many avenues of medical research. However, interpreting imaging data can be difficult, as delineating specific anatomical structures and quantifying their geometric values requires highly trained radiologists and technicians.

Decades of research have produced numerous computational systems capable of performing various analytical operations on medical images. Among the most desired medical image analysis operations is *image segmentation*, defined as the delineation of anatomical objects within images. Computational systems have been designed to segment structures within medical images, with various levels of user interaction, or *supervision*. Designing effective segmentation systems requires researchers to address complex technical challenges, adapting cutting-edge methods from machine learning, computational shape modeling, mathematics and statistics, and a host of other specialized fields to the problem of image segmentation. This complexity stems from the intricate nature of analyzing medical images. Effective human analysis of medical images, at the level necessary for clinical use, requires years of training in anatomy, pathology, imaging, and other specialized topics on the part of radiologists. Image analysis researchers aim to encode as much of this information as possible into automated systems, reducing the amount of supervision required by human users and focusing manual effort to the most difficult and subjective aspects of the problem.

Delving further into the fundamental utility of computerized image analysis systems reveals a daunting barrier to bringing even the most effective current systems into the clinical world, where they may be used to directly improve patient care. Highly trained medical professionals, with analysis and diagnosis skills expertly honed through years of study and practice in the clinical environment, are understandably hesitant to spend valuable time learning to operate new image analysis systems. It is widely believed that computerized systems will not soon, if ever, be able to match the effectiveness of analysis that a radiologist can provide. This belief among the medical community has been recognized and accepted by many in the computerized image analysis research community, causing many systems to be designed to aid, rather than replace, expert radiological analysis. [1] Even with systems designed to aid radiologists, a barrier of clinical accessibility remains. This barrier exists because these systems require levels of technical or algorithmic knowledge unacceptable to the average clinician, whose expertise lies in a vastly different area. Thus, the potential benefits of many automated and semi-automated image analysis systems are outweighed by the amount of technical expertise necessary to effectively use them. Algorithm parameters are a prime example of this accessibility barrier. Many techniques in image analysis require detailed knowledge of high level mathematics, statistics, artificial intelligence, machine learning, computer science, or computer vision theory to understand and adjust parameters that will allow them to perform optimally on a given medical image. As noted by Yushkevich et al. in their 2004 technical report, “semiautomatic segmentation techniques rely on parameters whose values often make sense only in the context of their mathematical formulation.” [2] As a result, the clinical world relies mainly on humans to perform image segmentation, and has yet to adopt many potentially beneficial automated analysis techniques. [3]

With these limiting factors in mind, we utilized our novel image analysis framework, Shells and Spheres (S&S) , to design a computational medical image segmentation system specifically to appeal to the clinician. This image segmentation system was designed to be capable of automated 3D segmentation, following a simple manual initialization procedure which specifically utilizes only areas of expertise native to clinicians: anatomy and pathology. This design was made possible by the unique nature of our analysis framework.

The S&S framework is based on a construct called a *sphere map*, which is a set of spheres, exactly one sphere centered at each image pixel, whose radii can be adjusted. Calculations denoted as *Variable-Scale Statistics* (VSS) are performed on populations of pixels within spheres, as well as populations of adjacent and overlapping spheres. The ultimate goal of radius adjustment is to produce a sphere map in which each sphere is as large as possible without crossing an object boundary, such that the spheres' radii are equivalent to what is commonly known as a *distance map* [4]. This process of properly setting the radii of each sphere in a sphere map is known as sphere map optimization.

Certain spheres can be denoted as *medial spheres*, in that they reach more than one boundary point. From a correctly optimized sphere map, a combination of the pixels within medial spheres along the same *medial manifold* will yield a segmentation of the anatomical object they belong to. Both boundary and medial information is embedded in the sphere map, and can easily be extracted for geometric analysis or other purposes. Medial points calculated with our algorithm are each inherently linked to a boundary point, defining surface normals attached directly to the medial manifold. This form of output has proven especially useful for shape modeling and analysis, which has led to our collaborative work with the Right Ventricular Outflow Tract (RVOT), producing segmentations for the purpose of modeling the vasculature surrounding the Pulmonary Valve (PV).

The following sections of this chapter will present the goals and motivations of this work, a brief description of the subsequent chapters, and definitions of the terms and notation used in our research.

1.1 GOALS AND MOTIVATION

The research presented in this dissertation was designed to meet a number of goals, both technical and clinical. Our research was motivated by the desire to create a novel, intuitive image analysis framework utilizing rotationally invariant localized collections of pixels on which population statistics can be calculated. In addition, our work with clinical collaborators and the Insight Toolkit (ITK) lead us to take on the challenge of designing a

system specifically for clinical use, removing direct manual parameter adjustment and generally minimizing supervision. As with the creation of any new technology, experimentation and validation were fundamental aspects of our research, allowing us to verify the accuracy of our automated segmentations, and thereby deliver them with confidence to collaborators for mechanical modeling of the RVOT, towards the development of a Tissue Engineered Pulmonary Valve (TEPV).

Many conventional segmentation techniques are designed around fixed-size sets of pixels, or *kernels*. This allows analysis of local features, based on kernel size, most often used in attempts to detect object boundaries in images. Such approaches ignore global image information, and the detection of object boundaries can therefore be severely corrupted by local noise, producing a strong fake signal in the small region represented by the kernel.

We have designed the Shells and Spheres (S&S) image analysis framework to model the homogeneous regions within objects, rather than local boundary regions, to perform object detection. We developed variable-radius spherical operators to function as our “kernels”, allowing us to change the size of the region from which we sample, generally increasing the number of pixels for use in our statistical calculations. Statistical comparison of adjacent spherical regions from within different objects allows us to differentiate between objects, thereby detecting boundaries.

By incorporating variable sized spheres, we are pursuing an approach similar to that commonly known as *scale space*. [5, 6] The ability to adjust the radii of our spherical operators allows us to explore various levels of scale space in our analysis, incorporating more global image information than kernel-based approaches. However, unlike the common scale space technique of gaussian blurring, our operators maintain sharp boundaries regardless of their radius. This behavior, along with the ability to combine multiple spheres to form complex shapes, allows our spherical operators to closely conform to complex object contours.

Further study and development of our novel framework, along with the development of an associated algorithm to perform image segmentation, were primary goals of the research presented. Initial algorithms required a number of parameters to be set, each with a great impact on the accuracy of segmentations produced. Adjusting algorithm parameters manually required a great deal of specific algorithm knowledge, as well as specialized knowledge of

the image to be segmented, since effective parameter settings varied greatly between different images. A primary goal of our research efforts was to enhance our segmentation system by removing the necessity for manual parameter instantiation, improving segmentation speed and accuracy, and adopting a comfortable, robust user interface.

Because we believe that approaches to medical image segmentation based purely on image data itself will not be effective, [1] we designed and added features to our initial segmentation algorithm to incorporate expert *a priori* information, to automatically optimize algorithm parameters, and to automatically produce a 3D segmentation after a manual initialization procedure taking only a moment of the user’s time. A secondary goal, precipitated by the addition of *a priori* information and considerations about clinical relevance, was to create a clinically appealing automated segmentation system. As discussed in the previous section, the foremost barrier to clinical adoption of automated segmentation software, aside from concerns about its effectiveness, is the complexity of operation. As a result, hospitals still rely primarily on humans to perform necessary segmentation procedures. [3] Although some researchers have developed excellent, intuitive user interfaces for image segmentation, [2] we hypothesize that to be truly viable in the clinical setting, a segmentation system must also limit supervision to tasks familiar to the user. We present such a system in this dissertation, requiring only moments of initialization from clinical users, in a form with which they are familiar and comfortable.

Along with developing our automated segmentation system, we planned studies of the algorithm and its performance, to facilitate improvements and an overall better understanding of the behavior and features of Shells and Spheres. These included a study validating the use of population statistics on our spherical sampling regions, experiments testing various methods of automated algorithm parameter determination, and studies detailing Methods of tuning those parameters using the newly added *a priori* information.

Fundamental goals of any novel algorithm research are experimentation and validation. We have tested multiple variations of our segmentation system, validating their effectiveness in terms of segmentation accuracy as compared to expert manual segmentations, and presenting additional pertinent measurements such as computational time. Validation data, in the form of a database of 3D ovine MRI data sets, were collected and provided by collaborators

at Children’s Hospital, Boston and the University of Pittsburgh. Manual 3D segmentations of the Right Ventricular Outflow Tract (RVOT) in our data sets were produced by three human operators for comparison to our automated segmentations.

Our final research goal was to provide accurate automated segmentations of the RVOT to our collaborators, including Dr. Michael Sacks at the University of Pittsburgh and clinical collaborators at Children’s Hospital, Boston. Their research focuses on the development of a new Tissue Engineered Pulmonary Valve (TEPV) with growth and remodeling capabilities, which would greatly increase the quality of life of young patients with congenital valve defects. Recently, a TEPV was developed using autologous cells and biodegradable polymers and implanted in growing lambs, functioning for up to 5 months. [7] While these results are promising, supplementary research is required for further development of the technology. Key research areas include the structure-strength relationships in the native Pulmonary Valve (PV) and quantification of the 3D geometry of the vasculature structure surrounding the PV, known as the Right Ventricular Outflow Tract (RVOT). Our segmentation system produces useful output for geometric modeling, providing linked medial/boundary point pairs, which may be used in novel ways to describe local geometric features such as medial radius and boundary curvature. Ten ovine MRI data sets have been segmented by our automated system, in addition to the manual segmentations produced for each data set by three experts, for validation purposes. These data have been delivered to our collaborators, and will be used for further study of the RVOT.

1.2 DESCRIPTION OF CHAPTERS

This dissertation is organized as follows.

Following this introduction, sections containing definitions of common terms and a glossary of notations are provided.

Chapter 2 presents background on Tissue Engineered Pulmonary Valves (TEPV) and the mechanical modeling of the Right Ventricular Outflow Tract (RVOT) that will enable the development of new TEPVs. Background is also provided on medical image segmentation, the area into which the present work is being introduced.

In chapter 3, we introduce the theory and notation of the Shells and Spheres framework in detail, including associated VSS. We present our n -dimensional S&S segmentation algorithm, designed for segmenting cardiovascular structures. We also present a streamlined version of S&S analysis, designed to perform analysis on a noiseless image. This process has become an integral part of our real image segmentation system, used to calculate gold standard sphere maps for binary masks, created manually for algorithm initialization and validation.

Chapter 4 presents a technique we have developed for automatically optimizing the parameters of our segmentation algorithm. These parameters, detailed in Chapter 3, are interdependent, and thus cannot be optimized individually. Because testing a single set of parameters on a 3D data set requires on average 1-5 hours of computational time, testing multiple parameter settings on 3D data becomes untenable. A method of determining effective parameters for a 3D data set by optimizing parameters on a selected 2D slice is presented and validated, yielding effective 3D segmentations. By tracing a single contour of the target object in the selected 2D slice, the user initializes the algorithm parameter optimization procedure. In addition to being used to optimize algorithm parameters, information extracted from this 2D tracing is used to calculate all additional values necessary for automated 3D segmentation. This enhances our original algorithm by adding *a priori* information. With clinical accessibility in mind, it takes advantage of a clinician's expertise without requiring algorithmic or technical expertise.

Chapter 5 reports the results of testing and validation experiments for our segmentation algorithm, the original parameter optimization technique, as well as multiple alternative parameter calculation techniques involving calculation of algorithm parameters directly from the manual 2D initialization tracing. Validation is performed using expert manual segmentations of our database of 3D ovine MRI data sets, performed by three individual subjects.

Finally, in Chapter 6 we discuss our conclusions from the research presented and propose future work in development of the Shells and Spheres framework, segmentation algorithms based on Shells and Spheres, and clinically accessible automated segmentation systems.

2.0 BACKGROUND AND SIGNIFICANCE

2.1 TISSUE ENGINEERED PULMONARY VALVES

Surgical replacement of faulty heart valves has become a common medical procedure, being performed on over 70,000 of patients each year in the United States alone. [8] This surgery can significantly increase the chances of survival and quality of life for patients with valve disorders. A number of types of mechanical and tissue valve substitutes have been developed and are currently available for implantation in patients. Congenital valve defects, however, which are seen in over 20,000 children born each year, require exceedingly small valve sizes that are not currently commercially available. Additionally, for pediatric applications, continual replacement of substitute valves is required as the patient grows, since current replacement valves are incapable of growing along with the patient. Development of replacement valves with the capacity for growth would greatly increase the quality of life and decrease the risks associated with continual valve replacement surgeries for such pediatric patients. Ideally, along with the capacity for growth, a perfect replacement valve would be non-obstructive, non-thrombogenic living tissue lasting the lifetime of the patient and providing ongoing remodeling and repair of cumulative injury.

Improving current valve bioprostheses has become a popular research goal in laboratories around the world, but new technologies must be developed to meet the aforementioned requirements of an ideal replacement valve. One promising replacement valve technology is the Tissue Engineered Pulmonary Valve (TEPV), which offers replacement valves containing living cells, with the potential for growth and remodeling. Using autologous cells and biodegradable polymers, a TEPV has been recently developed and implanted in growing lambs. [7] These valves have functioned in the pulmonary circulation of the lambs for up to

five months. Such promising initial results must be supplemented by further research into the structure-strength relationships in the native pulmonary valve (PV) and quantification of the actual 3D geometry of the right ventricular outflow tract (RVOT) resulting after valve implantation. The next section will detail techniques used to model cardiovascular shapes, allowing geometric quantification of structures such as the RVOT.

2.2 MEDICAL IMAGE SEGMENTATION

Medical images provide the means for clinicians to obtain a wide variety of in-situ structural and functional data on which to base diagnoses in a non-invasive manner, and have thus become an extremely powerful clinical tool. The ability to visualize the shape, location, deformation, and other information pertaining to various anatomical structures is useful in surgical planning, diagnosis, and post-treatment evaluation, as well as many avenues of medical research. However, interpreting imaging data can be difficult, and delineating specific anatomical structures and quantifying specific geometric values usually requires highly trained radiologists and technicians.

Segmentation, in particular, is a highly desirable procedure in medical image analysis. Image segmentation is classically defined as the process of partitioning an image into nonoverlapping regions, each of which is homogeneous with respect to some characteristic, often intensity or texture. [9] Given Ω as the domain of an image, the goal of segmentation is to determine K pixel sets $S_k \subset \Omega$, defining the K segments that make up the image. Thus,

$$\Omega = \bigcup_{k=1}^K S_k \quad (2.1)$$

where $S_k \cap S_j = \phi$ for $k \neq j$. The process of segmenting a specific anatomical structure within an image is a complex procedure, hampered by image noise, blurred boundaries, obfuscation of structures due to scanner attenuation, limitations due to data resolution, and ultimately by the subjectivity of the very definition of objects within medical images.

To account for these sources of ambiguity, a number of techniques can be employed. One technique is to allow sets $S_k \subset \Omega$ to be *fuzzy sets*. A fuzzy set is defined as a pair (A, m) where A is a set and $m : A \rightarrow [0, 1]$. For each $x \in A$, $m(x)$ is the *grade of membership* of x to set A . [10] This grade of membership does not represent a likelihood of x being a member of set A . Rather, it represents how similar x is to an element that would belong to set A . In this way, in terms of segmentation, a pixel x can share similarities with pixels from a number of anatomical objects without being strictly defined as a part of any. A second technique is to simply remove the constraint $S_k \cap S_j = \phi$ above, allowing segments to overlap. In this formulation, a given pixel can belong to multiple image objects. It can be said that such a pixel exhibits an equal likelihood of being a member of each object. This method is advantageous in its ease of implementation, and is useful in defining likely object boundaries in cases where segmentation overlap is minimal.

Despite the technical and theoretical challenges, the increasing size and number of medical images used in modern clinical practice has necessitated the use of computational systems for image segmentation. Advances in imaging technology have drastically increased the prevalence of 3-dimensional (3D) imaging data, replacing traditional 2-dimensional (2D) images in many clinical procedures. Processing 3D data, instead of 2D slices, introduces other problems such as anisotropy in resolution, when in-slice pixel resolution is greater than between-slice spacing. In addition, the size of high resolution 3D data can severely tax the memory capacity and speed of even high-performance computers, presenting another challenge to researchers exploring image processing and segmentation.

Manual segmentation is still the most common clinical practice [3], though it is time-consuming (especially for 3D images) and requires a great deal of anatomical knowledge and clinical expertise on the part of the user. In addition to being slow and labor-intensive, manual segmentation is prone to significant variation between users due to the subjectivity of image interpretation. Automated and semi-automated methods for computational image segmentation are active research topics, and include a wide range of approaches. [11,12,13,14,15,16,17,18,19,20,21] The various approaches to image segmentation include statistical and shape-based techniques, as well as hybrids of the two. Making use of statistical information at varying scales to determine local shape features, our Shells and Spheres algorithm falls

in the category of hybrid techniques. Different techniques require varying levels of training information, defining the level of *supervision*, and user interaction, forming the range between manual and automated methods. The majority of techniques, including Shells and Spheres, are classified as semi-automated, though this classification comprises a wide range of user interaction, requiring varying amounts of time, attention, and technical expertise from the user.

Many conventional methods for image processing, including image segmentation, consider a region of fixed size and shape, usually referred to as a *kernel*. Such approaches suffer a tradeoff between detecting small-scale features in images and considering global information. These can be mutually-exclusive goals when using fixed-size sampling regions. In addition, small kernels generally are highly sensitive to image noise, which cannot be distinguished from object boundaries with limited local information. Other common approaches define dynamic regions adjoining boundaries using deformable contours [14] or level sets [11, 22]. While such methods exhibit robustness to the image noise that hampers kernel-based techniques, sensitive manual interaction is required to place an initial surface model within the image and to choose appropriate parameters. Additionally, standard deformable models have suffered from poor convergence to concave boundaries. [9]

Our approach, instead, uses a set of spheres whose individual radii are optimized using Variable-Scale Statistics (VSS) operators to achieve maximum discrimination between image regions. We retain the advantage of kernel-based approaches in that we do not require an initial model, but allow our sampling regions to change size, making use of both local and global image information to gather more meaningful statistics. These features give our method robustness towards local image noise without relying on a user-defined prior shape model. Not only do our spheres provide highly representative populations for boundary detection and region representation, but those spheres that touch at least two boundaries are also *medial*, as classically defined by Blum [23], providing a basis for medial feature extraction. Unlike Gaussian blurring, commonly used in multi-scale analysis [6, 24], Shells and Spheres preserves sharp boundaries with increasing scale.

A technique commonly used to expand 2D segmentation systems to 3D image segmentation is to segment 2D slices and subsequently combine them to reconstruct a 3D segmentation. [12, 13] This approach leads to problems due to plane motion and errors in cases where the surface of a shape becomes approximately parallel to the direction of slicing. To avoid these obstacles, many modern segmentation methods are inherently 3-dimensional. [15, 17, 18, 11, 20] Our Shells and Spheres framework [25], was developed in this spirit, taking advantage of the native symmetry in n -dimensions of hyperspherical operators, which maintain rotational invariance with varying radius in the computation of VSS for statistical pixel population analysis. Spherical operators are preferred over Gaussian kernels for their sharp boundaries, even at large scale. This approach to analysis and its applications to image segmentation have been briefly explored in Cois’s MS thesis [25] by applying a preliminary system to segmentation of the aortic arch in contrast-enhanced CT scans, yielding promising results that motivated the present research.

2.2.1 Fuzzy-Connectedness

Saha, et al., have developed a popular approach to image segmentation known as *fuzzy-connectedness*. This approach is based on creating paths between image pixels, which are used to calculate an *affinity* value representing the likelihood that the pixels at either end of the path are within the same image object. In some of their publications [16, 17, 21], they describe a set of *hyperballs* centered at each image pixel, grown as large as possible without crossing boundaries, used as a scale factor in their affinity equation. The goal of growing spheres at each pixel to meet image boundaries is similar to that of our technique, and thus warrants a detailed discussion. The method used for calculating the size of each of these hyperballs differs significantly from our approach. The fuzzy-connectedness method of deciding the proper scale of hyperballs (spheres) is fully detailed in [16]. The measure used to determine scale is a value denoted $FO_k(c)$, which is the “fraction of the object” of a hyperball $B_k(c)$ of radius k centered at pixel c . The value of $FO_k(c)$ is calculated using a user-defined homogeneity function W . This function is used to find the probability [0-1] that two pixels are in the same homogeneous region, given the intensity values of these two

pixels c and e , as in $W(|f(c) - f(e)|)$. In the calculation of $FO_k(c)$, this function is used to find the homogeneity measure between each pixel e in the outer shell of the hyperball, as represented in set notation by the subtraction of a sphere one size smaller from the current sphere $B_k(c) - B_{k-1}(c)$, and the central pixel c . Using these values, the equation for $FO_k(c)$ is given as

$$FO_k(c) = \frac{\sum_{e \in B_k(c) - B_{k-1}(c)} W(|f(c) - f(e)|)}{|B_k(c) - B_{k-1}(c)|}. \quad (2.2)$$

The numerator can be recognized as the sum of the homogeneity likelihood values given by W for each pixel e in the hyperball's outer shell when compared to the central pixel c . The denominator is a normalizing factor, representing the number of pixels contained in the outer shell being considered.

Thus, in a noiseless image, a sphere with an outer shell completely within a uniform intensity region would have an $FO_k(c)$ value of 1. A manual threshold t_s , under which the $FO_k(c)$ values of spheres must remain in order to continue growth, is required. Saha, et al. present a variety of possible homogeneity functions, each of which perform differently on a given image.

This method of scale calculation is not designed for extreme accuracy, as exact scale values at each pixel are not necessary for effective performance of the overall fuzzy-connectedness algorithm. Instead, Saha et al. have designed a technique that is computationally swift and relatively simple to implement. When considering the needs of our S&S system, namely, a technique that calculates the radius of each sphere as accurately as possible, a number of problems with the fuzzy-connectedness approach become readily apparent:

- Using only the outer shell greatly decreases the statistical population used, weakening the measurement
- The individual measure of homogeneity is derived from the comparison of only two pixels (c and e), rather than an entire population, easily falling prey to noise, tissue inhomogeneity, partial-volume effects, etc
- The necessity of choosing the homogeneity function W adds a knowledge-dependent task, requiring an experienced user

- The approach is highly dependent on having either uniform noise, or a very complex homogeneity function designed to account for a specifically known type of non-uniform noise

In noisy images, especially those with uneven noise characteristics such as MRI, these drawbacks become a major cause for concern. Using only the central pixel c as a basis for homogeneity comparisons is problematic, especially if that pixel is afflicted with a large amount of noise. Homogeneity functions must be chosen very carefully by a knowledgeable user. The interior region of the ball is not considered in any way, meaning that the number of pixels considered to make a decision is small, compared to the pixel population of the entire hyperball. There are two ways of considering this approach. Since the population being considered is small compared to the entire sphere, a few pixels outside the region should have a more profound impact on the sum of the outer shell, and therefore on the sphere growth decision, than the same few pixels would have had when averaged in to the pixel population of the entire sphere. This can be seen as a positive feature. However, giving more weight to a few pixels can also be detrimental, as a few noisy pixels may have a dramatic impact on the homogeneity value calculated. Viewed in this light, it can be seen that the smaller population is more susceptible to the negative effects of noise and other sources of inhomogeneity. The approach of our Shells and Spheres framework is to use the largest population possible to avoid this vulnerability.

2.3 CLINICAL IMAGE ANALYSIS

Medical imaging plays an ever more essential role in the clinical world, and automated segmentation has become an increasingly important goal, as already discussed. Current automated segmentation systems are either largely unreliable or too operationally complex, with the result that manual segmentation remains the standard practice. Some semi-automated algorithms are used, but these must be tuned via manual parameter adjustment for each individual image. The time and technical expertise required to manually set parameters deters most target end-users, particularly clinicians, from adopting image analysis technol-

ogy. Recent research has addressed this problem by creating software systems designed to appeal to the clinician by simplifying the interface and adding useful feedback for parameter initialization. [3] However, irrespective of interface, implementations of current automated segmentation algorithms still require users to learn algorithm specifics to effectively set parameters. As described by Yushkevich et al., most techniques “rely on parameters whose values often make sense only in the context of their mathematical formulation”. [2] This formulation is often outside the professional realm of the clinical end-user, making it highly unlikely that many will expend the time and energy necessary to master this new domain. A viable solution to this problem is to provide a means of automatically calculating segmentation parameters that requires minimal user input, in a form that is within the clinical realm of expertise.

We present such an approach in this dissertation. Using our novel automated segmentation framework, *Shells and Spheres* [25], which will be described in detail in the next chapter, we can produce a reliable 3D segmentation while limiting user interaction to a simple 2D manual tracing of a single contour at the onset of analysis. The system performs automated parameter optimization on the 2D slice and manual contour, deriving statistics and seed points for a fully automated 3D segmentation without further user assistance. We will present details of this system, along with results and validation, following the background on the Shells and Spheres framework.

3.0 THE SHELLS AND SPHERES FRAMEWORK

We present a framework for image analysis that can be used to calculate a distance map directly from statistical analysis of the image data. We introduce the concept of a *sphere map*, a set containing one sphere of adjustable radius centered at each pixel in a given source image. Though our framework is inherently n -dimensional, we simplify our discussion by using the term *pixel* to describe image data points in all dimensions, instead of other dimensionality-specific terminology such as *voxel*. Likewise, we use the term *sphere* when describing our regions, instead of *circle* or *hypersphere*. These spheres are rotationally invariant sets of pixels that define regions within the source image on which statistical operations can be performed. We present *Variable-Scale Statistics* (VSS) that can be computed on populations of pixels within and between spheres as they evolve. The primary objective of algorithms utilizing our framework is to make use of VSS operators to optimize the sphere map, i.e., to deduce the proper radius of each sphere such that it reaches, but does not cross, the nearest object boundary. The optimal spheres' radii are thus equivalent to a distance map, indicating the distance from the pixel at the center of each sphere to the nearest object boundary. Not only do such spheres contain highly representative pixel populations for boundary detection, but those spheres that touch at least two object boundaries are *medial*, as classically defined by Blum [23], providing a basis for medial feature extraction. Figure 1 shows a sampling of 2D medial spheres in a rectangle.

Unlike Gaussian blurring, a common technique for multi-scale analysis which increasingly blurs boundaries to obtain image features at higher scales [5], Shells and Spheres preserves sharp boundaries with increasing scale, i.e. sphere radius. The curved surfaces of spheres provide the ability to model complex object shapes by combining multiple overlapping spheres, each touching at least one point on the object boundary, with their individually varying radii

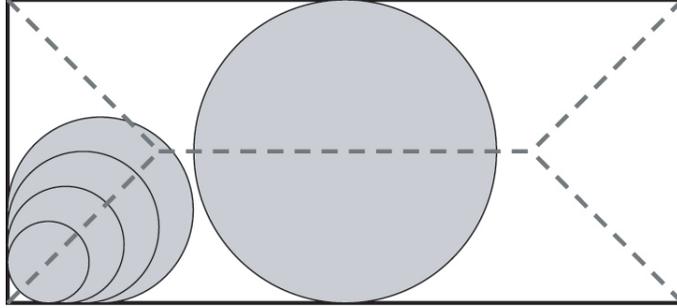


Figure 1: The Blum medial manifold (dotted line) of a rectangle, with selected medial spheres shown.

allowing for simultaneous, rotationally invariant, multi-scale analysis tuned to individual locations in an image. Thus, segmentations produced using Shells and Spheres, as shown in Figure 2, can accurately follow the complex boundary contours common in medical images.

The following sections will present the fundamental framework and basic operators of Shells and Spheres. Using this framework, a wide variety of algorithms for sphere map optimization are possible. One such algorithm designed for cardiac image segmentation will be presented, with experiments and validation to follow in subsequent chapters.

3.1 FRAMEWORK

We begin by defining our notation. As previously stated, Shells and Spheres is inherently n -dimensional. For brevity, we use the term *sphere* instead of *circle* or *hypersphere*, regardless of image dimension. Figures are presented in 2D for ease of illustration.

3.1.1 Shells and Spheres

Since the framework of Shells and Spheres is used to gather statistics on dynamic populations of pixels, we adopt a hybrid form of notation derived from standard set theory and statistics.

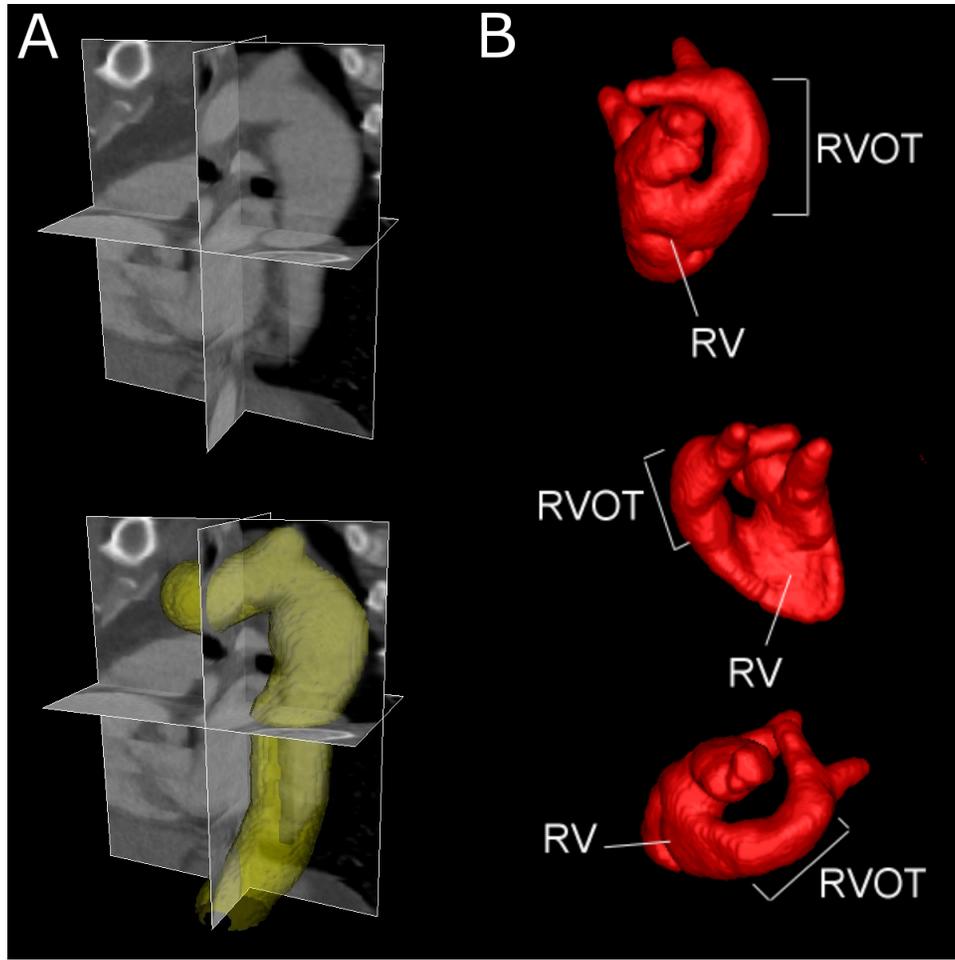


Figure 2: **A**: 3D contrast-enhanced CT scan of the aortic arch (top left), and the same scan with an overlaid 3D segmentation (bottom left) achieved using the Shells and Spheres framework. **B**: Surface model of a Shells and Spheres segmentation of the right heart with labeled Right Ventricle (RV) and Right Ventricular Outflow Tract (RVOT), shown from three different perspectives.

We denote vectors by lowercase bold-faced letters (\mathbf{x}), scalars by lowercase italic letters (r), and sets by uppercase letters (S). We use \mathbb{Z} to denote the set of all integers, and $\Omega \subset \mathbb{Z}^n$ to denote the set of all pixel locations in a sampled n -dimensional image.

Given an n -dimensional image with intensities $f(\mathbf{x})$ for $\mathbf{x} \in \Omega$, we define a *sphere map*, which assigns the radius $r(\mathbf{x})$ to the sphere centered at each pixel \mathbf{x} .

We define a sphere to be an n -dimensional neighborhood of pixels that lie within a radius r of a center point. We use an integer value for r , such that a sphere of radius r centered at a pixel \mathbf{x} is given by

$$S_r(\mathbf{x}) = \{\mathbf{y} : \text{round}(|\mathbf{y} - \mathbf{x}|) \leq r, \mathbf{y} \in \Omega\}. \quad (3.1)$$

Note the shorthand notation for the subscript r , meaning $r(\mathbf{x})$, the radius of the particular sphere at \mathbf{x} as given by the sphere map. In some instances, the reader will encounter an example with a different subscript, such as $S_1(\mathbf{x})$, meaning a sphere of radius 1, irrespective of $r(\mathbf{x})$. By definition, $\mathbf{x} \in S_r(\mathbf{x})$ for all \mathbf{x} , even when $r(\mathbf{x}) = 0$, and hence $S_r(\mathbf{x})$ is always non-empty.

A *shell* is a set of all pixels whose distance to the center rounds to a given radius, defined for radius r as

$$H_r(\mathbf{x}) = \{\mathbf{y} : \text{round}(|\mathbf{y} - \mathbf{x}|) = r, \mathbf{y} \in \Omega\}. \quad (3.2)$$

Shells are non-overlapping such that for concentric shells,

$$H_p(\mathbf{x}) \cap H_q(\mathbf{x}) = \emptyset, \quad p \neq q. \quad (3.3)$$

Additionally, shells are space-filling, and thus a sphere of radius r may be formed from a union of shells,

$$S_r(\mathbf{x}) = \bigcup_{k=0}^r H_k(\mathbf{x}). \quad (3.4)$$

Figure 3 illustrates the distribution of pixels in a series of concentric shells surrounding a central pixel in a 2D image. Each pixel is labeled with its integer radius from the central pixel (labeled “ \mathbf{x} ”). Fig. 4 shows an image containing two noiseless objects with pixel intensities of 1 and 9 respectively. Note that pixels in this case are represented by their intensity. The boundary between the image objects is identified by a straight dashed line. Pixel \mathbf{x} is surrounded by a concentric set of four shells $H_0(\mathbf{x})$, $H_1(\mathbf{x})$, $H_2(\mathbf{x})$, and $H_3(\mathbf{x})$, shown

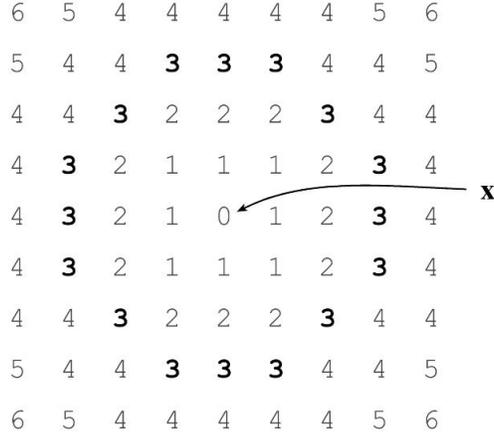


Figure 3: Each pixel is shown as a number indicating its integer distance from the central pixel. If we denote the central pixel as \mathbf{x} , then pixels labeled n are members of the set $H_n(\mathbf{x})$. For example, the pixels labeled “3” (shown in bold) comprise the set $H_3(\mathbf{x})$.

separated by dashed circles. Shell $H_3(\mathbf{x})$ is truncated by the edge of the image. The union of all four shells is $S_r(\mathbf{x})$, shown enclosed by a solid circle, also truncated by the edge of the image, with a radius governed by the value of $r(\mathbf{x}) = 3$ in the sphere map. Similarly, on the other side of the boundary, pixel \mathbf{y} with an intensity value of 9 has three shells whose union $S_r(\mathbf{y})$ has a radius $r(\mathbf{y}) = 2$. Both $S_r(\mathbf{x})$ and $S_r(\mathbf{y})$ touch but do not cross the boundary, and are therefore correctly optimized.

The correctly optimized sphere map of the image in Fig. 4 is shown in Fig. 5, with each pixel represented by the radius of the sphere centered at that pixel. Note the linear increase in sphere radius with distance from the boundary and the fact that the radius equals zero adjacent to the boundary.

3.1.2 Variable Scale Statistics

We derive a number of statistics at each pixel \mathbf{x} , calculated on the intensities of pixels within spheres. Since these statistics depend on the radii of the spheres, we call them Variable Scale Statistics (VSS). We denote as *primary statistics* those VSS at \mathbf{x} calculated using only the

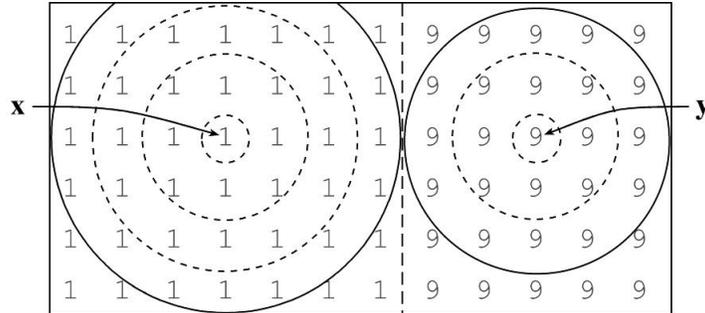


Figure 4: Noiseless image with boundary between two objects. Correctly scaled spheres $S_r(\mathbf{x})$ with $r(\mathbf{x}) = 3$ and $S_r(\mathbf{y})$ with $r(\mathbf{y}) = 2$ touch, but do not cross, the boundary. Numbers indicate pixel intensity.

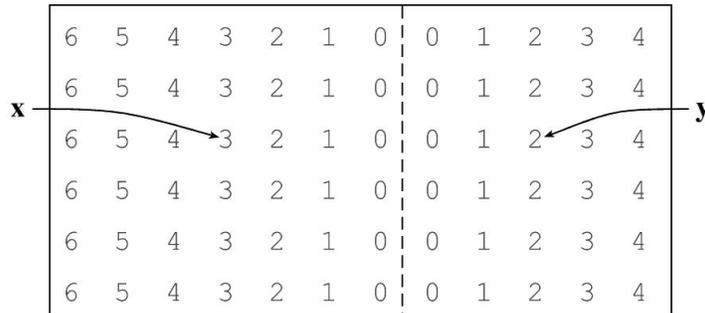


Figure 5: Correctly optimized sphere map of the image in Fig. 4. Numbers indicate the integer radius of the sphere at each pixel. Pixels \mathbf{x} and \mathbf{y} are labeled as before.

population of pixels within $S_r(\mathbf{x})$. Higher-order *secondary statistics* are VSS derived from multiple spheres.

3.1.2.1 Primary Statistics. The primary statistics at pixel \mathbf{x} concern only the population of pixels within the sphere $S_r(\mathbf{x})$. Thus the *mean* at pixel \mathbf{x} is the mean intensity of all pixels within the population $S_r(\mathbf{x})$, defined as

$$\mu(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} \sum_{\mathbf{y} \in S_r(\mathbf{x})} f(\mathbf{y}), \quad (3.5)$$

where $|S_r(\mathbf{x})|$ is the number of pixels in $S_r(\mathbf{x})$ and $f(\mathbf{y})$ is the image intensity at \mathbf{y} . The *variance* at pixel \mathbf{x} is defined as

$$\sigma^2(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})| - 1} \sum_{\mathbf{y} \in S_r(\mathbf{x})} [f(\mathbf{y}) - \mu(\mathbf{x})]^2. \quad (3.6)$$

where $|S_r(\mathbf{x})| > 1$. The standard deviation $\sigma(\mathbf{x})$ is simply the square root of the variance.

The *first-order moment* of intensities within $S_r(\mathbf{x})$ is given by

$$\mathbf{m}(\mathbf{x}) = \sum_{\mathbf{y} \in S_r(\mathbf{x})} (\mathbf{y} - \mathbf{x})f(\mathbf{y}). \quad (3.7)$$

Due to the finite extent of an image's domain Ω , a sphere may be truncated by one or more edges of the image (for example, $S_3(\mathbf{x})$ in Fig. 4). Unlike conventional kernels, which usually require pixel values outside the image to be arbitrarily defined, our spherical sets simply exclude such locations from all calculations. Thus, truncation will not adversely affect $\mu(\mathbf{x})$ or $\sigma(\mathbf{x})$. However, the first-order moment will suffer a bias due to asymmetrical pixel distribution within the truncated sphere. We compensate for this by defining a measure that shows no edge effect, which we call *VSS gradient*. Given the center of mass of pixel locations in sphere $S_r(\mathbf{x})$,

$$\mathbf{c}(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} \sum_{\mathbf{y} \in S_r(\mathbf{x})} \mathbf{y}, \quad (3.8)$$

the VSS gradient at \mathbf{x} is

$$\nabla f(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} [\mathbf{m}(\mathbf{x}) - \mu(\mathbf{x})(\mathbf{c}(\mathbf{x}) - \mathbf{x})]. \quad (3.9)$$

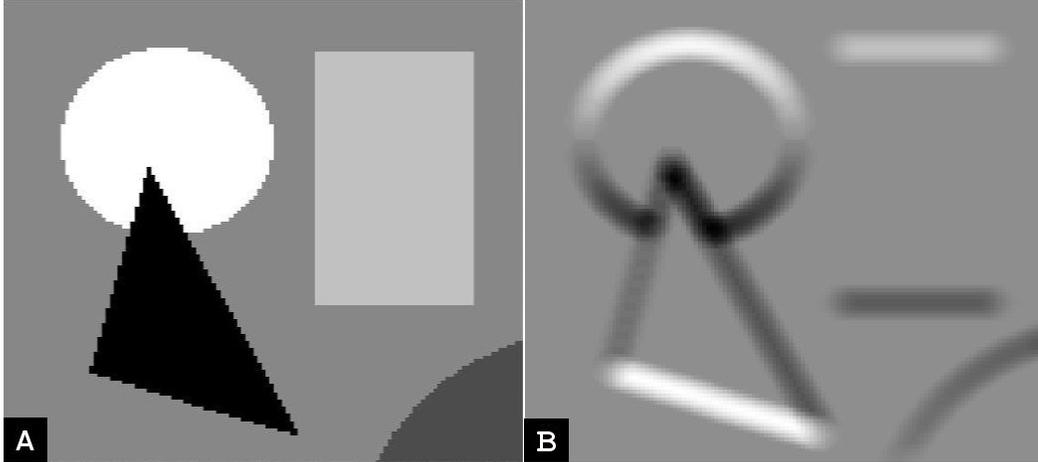


Figure 6: **A:**A synthetic noiseless image. **B:**A corresponding vertical gradient image, calculated using the VSS gradient. Note that there are no edge effects, such as those seen with conventional gradient calculations.

The VSS gradient does not suffer from the usual edge effects of convolution kernels, as seen in the vertical VSS gradient image shown in Figure 6.

Note that for non-truncated spheres $\mathbf{c}(\mathbf{x}) = \mathbf{x}$ and VSS gradient is equivalent to the moment vector normalized to the number of pixels, $\mathbf{m}(\mathbf{x}) / |S_r(\mathbf{x})|$.

All of the above statistics can be computed incrementally as shells are added to, or removed from, a given sphere, significantly reducing computational load during sphere map optimization.

3.1.2.2 Secondary Statistics. We define secondary statistics as higher order VSS derived by combining multiple spheres to form more complex neighborhoods. One such neighborhood, $S^{-1}(\mathbf{x})$, is defined as

$$S^{-1}(\mathbf{x}) = \{\mathbf{y} : \mathbf{x} \in S_r(\mathbf{y})\}, \quad (3.10)$$

the set of all pixels whose spheres contain \mathbf{x} . The -1 superscript is used to impart the flavor of an inverse function. Note that since it is always true that $\mathbf{x} \in S_r(\mathbf{x})$ it must likewise

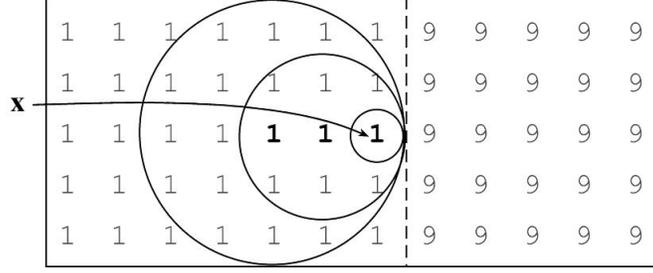


Figure 7: The $S^{-1}(\mathbf{x})$ set of spheres that contain pixel \mathbf{x} , adjacent to the boundary between two noiseless objects with respective intensities of 1 and 9.

always be true that $\mathbf{x} \in S^{-1}(\mathbf{x})$.

Given an optimized sphere map, $S^{-1}(\mathbf{x})$ will consist entirely of pixels within the same object as pixel \mathbf{x} . Fig. 7 shows members of one such $S^{-1}(\mathbf{x})$ set, consisting of three pixels (bold), whose spheres contain \mathbf{x} . Notice that all three spheres touch but do not cross the boundary, so this particular $S^{-1}(\mathbf{x})$ set is correctly optimized.

Secondary statistics are derived from populations of spheres such as $S^{-1}(\mathbf{x})$. Thus, the *mean of means* at pixel \mathbf{x} is defined as

$$\mu_{\mu}(\mathbf{x}) = \frac{1}{|S^{-1}(\mathbf{x})|} \sum_{\mathbf{y} \in S^{-1}(\mathbf{x})} \mu(\mathbf{y}), \quad (3.11)$$

or the mean of the mean intensities for all the spheres in $S^{-1}(\mathbf{x})$. In a noiseless image containing distinct homogeneous regions, the $\mu_{\mu}(\mathbf{x})$ yielded by a correct sphere map will be identical to the original image.

Likewise, the *standard deviation of the means* is defined as

$$\sigma_{\mu}(\mathbf{x}) = \left[\frac{1}{|S^{-1}(\mathbf{x})| - 1} \sum_{\mathbf{y} \in S^{-1}(\mathbf{x})} [\mu(\mathbf{y}) - \mu_{\mu}(\mathbf{x})]^2 \right]^{\frac{1}{2}}. \quad (3.12)$$

Note that the above definition of $\sigma_{\mu}(\mathbf{x})$ is given only for $|S^{-1}(\mathbf{x})| > 1$. For noiseless images,

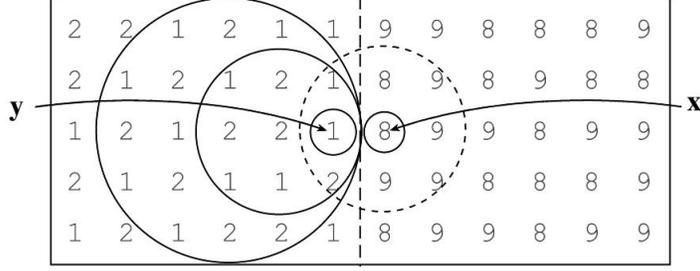


Figure 8: Image with noise. Pixel \mathbf{x} is deterred from extending its sphere across the boundary because its mean is an outlier in the population $S^{-1}(\mathbf{y})$.

a correct sphere map will yield $\sigma_\mu(\mathbf{x})$ values of 0 for all pixels.

We compute a *z-value* to provide a measure of how well $S_r(\mathbf{x})$ fits into the current $S^{-1}(\mathbf{y})$ set,

$$z_\mu(\mathbf{x}|\mathbf{y}) = \frac{|\mu(\mathbf{x}) - \mu_\mu(\mathbf{y})|}{\sigma_\mu(\mathbf{y})}. \quad (3.13)$$

The justification is that in an optimized sphere map, if $S_r(\mathbf{x})$ were to contain pixel \mathbf{y} , then $\mu(\mathbf{x})$ should fall well within the distribution of means for all spheres that already contain \mathbf{y} . This concept is illustrated in Fig. 8, which shows pixel \mathbf{x} attempting to extend its sphere across the boundary to include pixel \mathbf{y} . We have included noise in the image to demonstrate that a high *z-value* could be used to stop the growth of $S_r(\mathbf{x})$ at the boundary, even in the presence of noise. It should be noted that the utility of this statistic is dependent on a reasonable initialization of the sphere map, such that the percentage of spheres not crossing boundaries is high enough to lend statistical validity to $\mu_\mu(\mathbf{y})$ and $\sigma_\mu(\mathbf{y})$.

3.1.2.3 Validation of Population Testing During initial development of the Shells and Spheres framework and segmentation algorithm, we accepted a number of assumptions about the statistical properties of our spherical sampling regions and of medical images. Our current techniques are based on using statistical population testing between adjacent

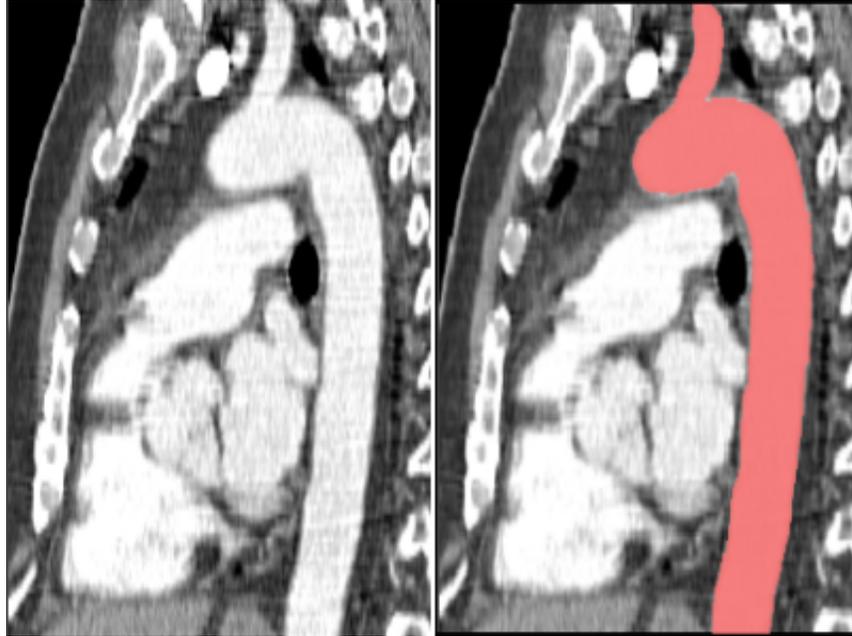


Figure 9: Left: 2D CT image of the Aortic Arch (AA) used in preliminary testing. Right: 2D AA image with manual tracing highlighted (pink).

regions to delineate boundaries and discern objects within images. The use of such population testing relies on the assumption that the intensity distributions of the populations of pixels within these regions are approximately Gaussian. We conducted a brief study to test this assumption by validating the normality of our spherical operators at various sizes when placed in various positions in a real image relative to object boundaries. Our hypothesis was that the normality of our spheres would increase as they grew within relatively homogeneous image objects, reaching a maximum just before crossing a boundary, and decrease rapidly upon crossing.

Tests were conducted on selected spheres using a CT image of the aortic arch and an expert manual tracing, shown in Figure 9. We will briefly study and explain the behavior of the normality of the populations of pixels within spheres as they grow, to validate our analysis techniques and facilitate a deeper understanding of the statistical properties of our spheres.

The mask created by our manual tracing was used to determine a ground truth sphere map by applying a noiseless S&S algorithm, which will be described in detail in the next section. For our experiments, we chose the set of all spheres with a gold standard radius of 11. The average diameter of the aortic arch in our image is approximately 26 pixels, thus the largest spheres within (medial spheres) have a radius of approximately 13 pixels. This places our test spheres roughly two pixels off the medial manifold of our target object. These spheres were chosen because they contain a large enough pixel population to provide accurate statistics (441 pixels) and are not likely to lie along the medial manifold of the object, so they will cross one boundary at a time as they grow. This choice generalized our experiment by removing the possibility of an amplifying effect on our normality signal when a spheres cross two boundaries simultaneously. The normality of the population of pixels within each sphere was calculated using the Shapiro-Wilks (SW) test for normality in SPSS version 14.0 (SPSS, Inc., Chicago IL). [26]

Figure 10 shows the normality, measured as the p-value of the SW test, of two randomly selected spheres in our image as they are grown from radius 5 to radius 15. The p-value approaches 1 for perfect normality. Recall that the correct radius of these spheres is 11. Radius 11 is in fact the peak normality of both of these spheres, though the drop in normality after this radius, as an object boundary is crossed, is more pronounced in one than in the other. Both spheres also generate a small increase in normality past the notable drop after crossing an object boundary at radius 11. This short increase lasts until a second object boundary, on the opposite side of the sphere, is crossed, at which time the normality drops more dramatically.

To more accurately describe the signals seen in Figure 10, consider the fact that the two spheres measured were both centered within the aortic arch in our image, which we will refer to as object A . As a sphere centered in object A crosses an object boundary, a small number of pixels from another object, which we will refer to as object B , are included in its population, decreasing the normality of said population. As the sphere grows farther past this boundary into object B without crossing a boundary at any other point along the circumference of the sphere, the number of pixels added to the population from object A is greater than the number of pixels added from object B . This uneven influx of pixels causes

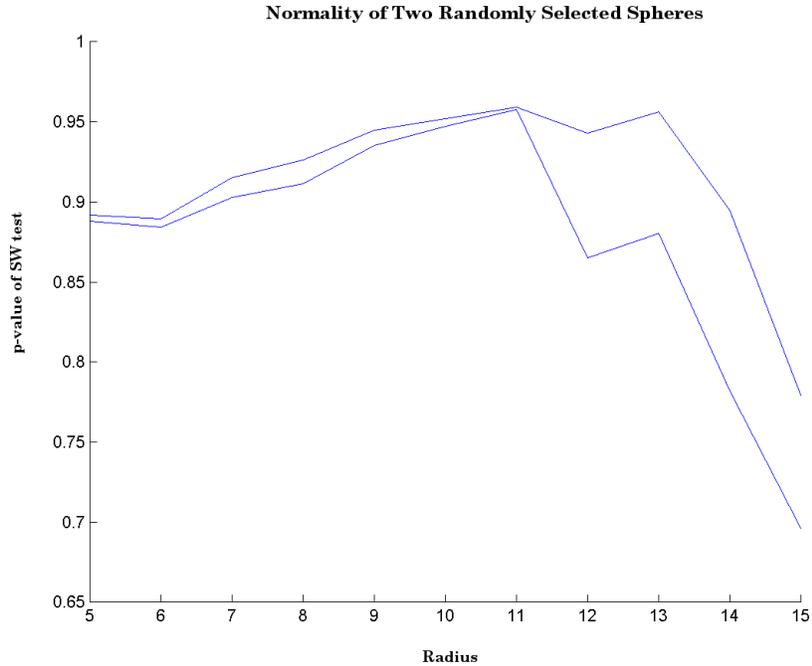


Figure 10: Plot of the normality, measured as the p-value of a SW test, of two randomly selected spheres for radii 5-15.

the slight rebound in normality seen in the figure from radius 12 to 13, until the second object boundary is crossed at radius 13, adjusting the ratio of pixels being added in favor of object *B*. From this point on, the normality is continually decreasing.

For further study, we recorded Shapiro-Wilks normality data on all pixels in our test image with a gold standard radius of 11, 585 pixels in total. The normality of each sphere was calculated at each radius within the range of 5 to 15. It was hypothesized that the maximum normality of these spheres would occur at radius 11, the radius before they crossed the object boundary to include pixels outside the AA. Figure 11 shows a histogram of the radius of maximum normality of all spheres in our study. We see that the predominant radius of maximum normality among these spheres is radius 13. Figure 12 shows a similar histogram, but with the SW normality measurements normalized by the statistical significance of the normality test. While the most common radius of maximum normality remains 13, mea-

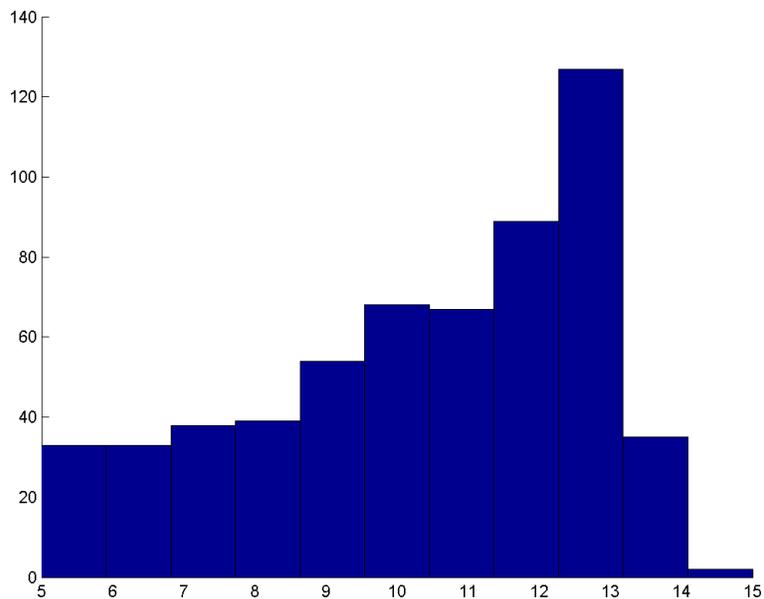


Figure 11: Histogram of the radius of maximum normality of all spheres with a gold standard radius of 11 in our test image.

measurements of maximal normality at lower radii are, in many cases, removed in favor of radii closer to the gold standard radius because of the low statistical significance of their normality measurements. The lower statistical significance of measurements taken on spheres of smaller radius is due to their smaller pixel populations.

The prevalence of radius 13, two sizes larger than the known correct radius, as the radius of maximum normality for our spheres can be explained by the aforementioned fact that upon initially crossing an object boundary, very few pixels outside of the object are included in a sphere's pixel population. In many cases, a sphere may be required to grow a few more steps before the number of pixels included from outside the original object is large enough to decrease the normality measurement.

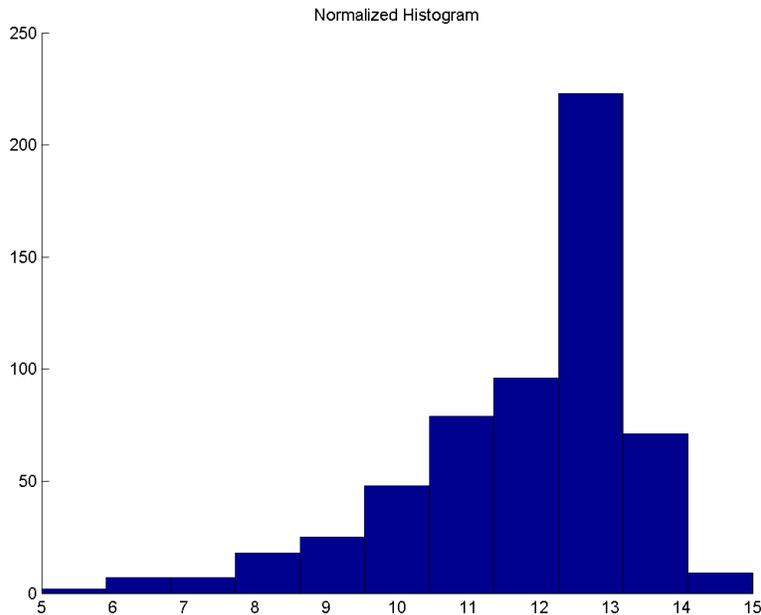


Figure 12: Histogram of the radius of maximum normality, normalized by statistical significance of the normality measurement, of all spheres with a gold standard radius of 11 in our test image.

To illustrate the behavior of the normality of our spheres as they grow, Figure 13 shows the mean normality of all of our test spheres (the 585 spheres of gold standard radius 11) over the previously given range of radii. This plot shows the trend of mean normality for our spheres, verifying that for this data it increases as expected until reaching the radius at which the object boundary is crossed, and then exhibits a decline as spheres grow past the boundary. The normality measurements of above 0.95 until radius 14 show that our assumption of normal populations of pixel within spheres, and thus our use of population testing, is valid until a sphere has taken several steps past a boundary.

While this data indicates that normality measures are not accurate enough to independently indicate the correct radius of a sphere, it also shows that the populations of pixels within our spheres are predominantly maximally normal at scales close to their correct scale,

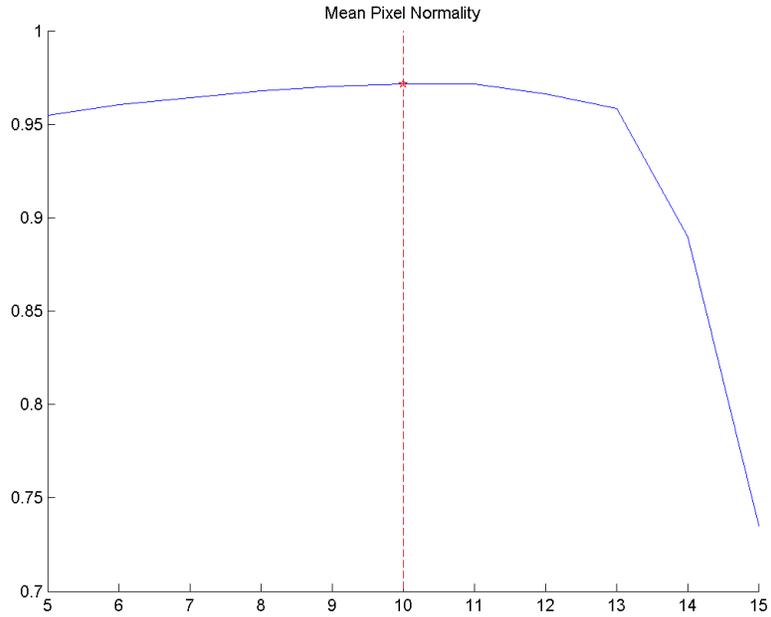


Figure 13: Plot of the mean SW normality of all spheres at each radius, from 5 to 15, with the maximum mean normality marked in red.

the common range of sizes at which population testing occurs in our algorithm (presented in the next section). This finding helps to verify the statistical validity of our population tests for region/boundary determination.

Using the tools provided by the Shells and Spheres framework, an algorithm has been developed for the purpose of optimizing a sphere map for, and segmenting objects within, cardiac images. This algorithm is based on the use of the population statistics and population testing described earlier in this chapter. A simpler algorithm has been developed to produce an optimized sphere map for noiseless data, and has proven essential to our initialization and error calculation techniques. The next section will present the noiseless sphere map optimization algorithm, while the subsequent section will introduce our full algorithm for real medical image segmentation.

3.2 NOISELESS SPHERE MAP OPTIMIZATION ALGORITHM

We have developed algorithms to optimize a sphere map, i.e. to set the correct radius of each sphere $S(\mathbf{x})$ in an image such that it reaches, but does not cross, the nearest boundary. The first, and simplest, algorithm will be referred to as the Noiseless Algorithm. This algorithm is specifically designed to optimize a sphere map for a noiseless image, a far simpler task than optimizing a sphere map for a real image. The Noiseless Algorithm has become an integral part of our research, utilized in the initialization phase of our Real Image Segmentation Algorithm, presented in the next section, as well as in our validation and error visualization techniques. It has also been used to determine the medial parameters of manually segmented shapes for diagnosis (not described in this thesis. [27])

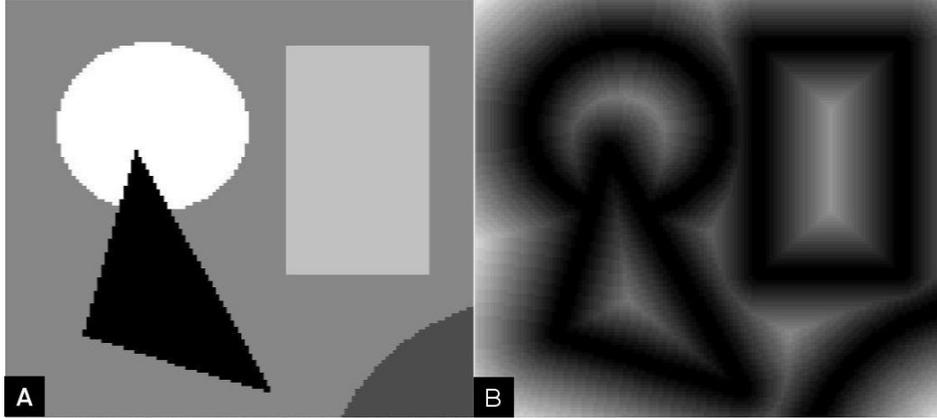


Figure 14: A: An example noiseless image. B: The sphere map created by the Noiseless Algorithm, represented as an intensity image where each pixel \mathbf{x} has an intensity value equivalent to the radius of its sphere $S(\mathbf{x})$.

A noiseless image consists of a set of objects, each containing a set of pixels of uniform intensity. Figure 14A shows an example of a synthetically generated noiseless image. Object boundaries can easily be determined in such images by detecting any change in intensity between pixels. Our Noiseless Algorithm functions simply by growing each sphere $S(\mathbf{x})$ in the image until it contains a pixel \mathbf{y} in its next outer shell, $\mathbf{y} \in H_{r+1}(\mathbf{x})$, with a different intensity value than that of pixel \mathbf{x} . This process produces a sphere map containing the correct radii of all spheres within the image, such as the one shown in Figure 14B.

In our research, this process has been used to produce gold standard sphere maps on binary masks of image objects created by manual or automated segmentations, as seen in Figure 15. This allows us to catalogue pixels by their correct radius, and to calculate various image-defining statistics for analysis of algorithms. Furthermore, it creates a gold standard sphere map within a manually segmented object against which the accuracy of a sphere map created by our automated system can be compared. In this case, the Noiseless Algorithm can be made even more computationally efficient by only growing spheres at pixels within the target segmented object. This comparison provides a means of quantifying and visualizing localized error in our automated segmentations, an extremely useful technique

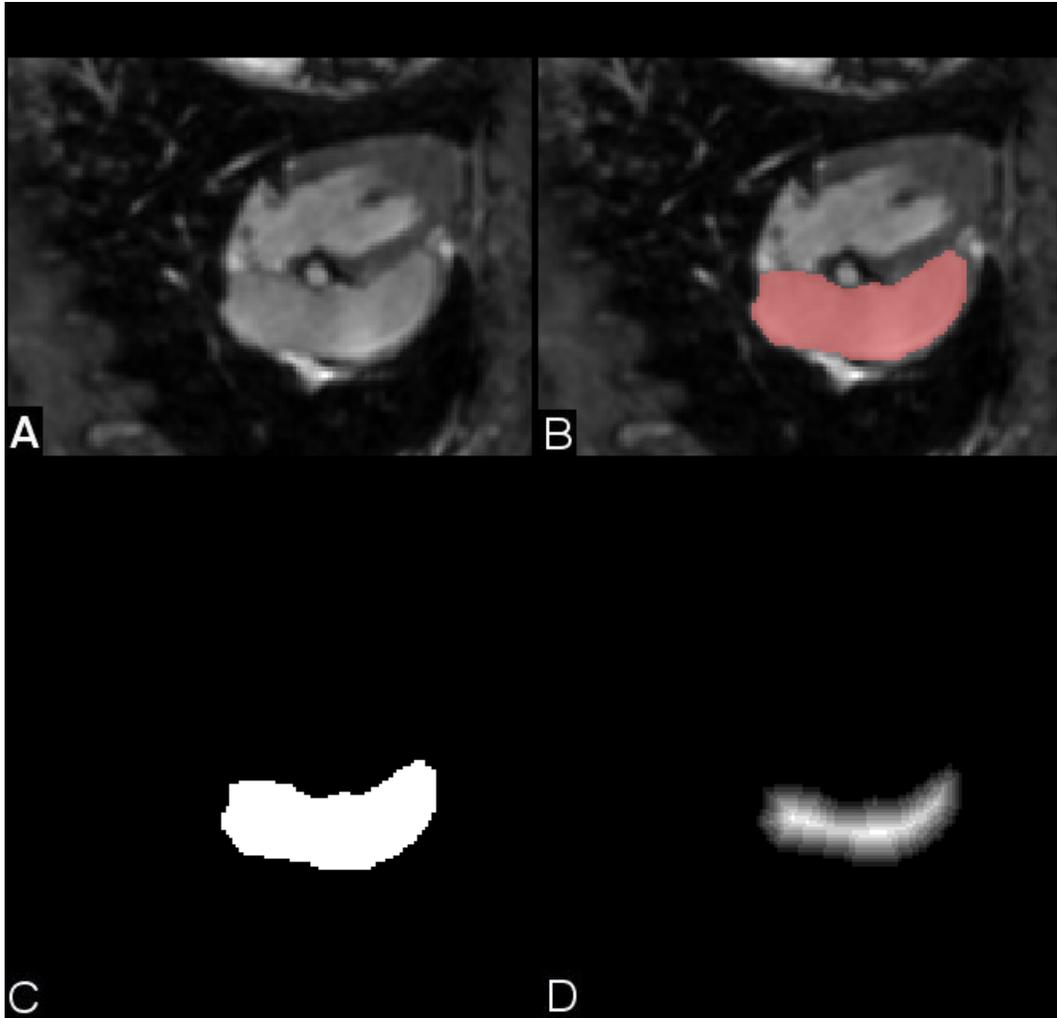


Figure 15: A: A 2D slice from one of our 3D ovine MRI data sets. B: The same 2D slice with a manual tracing overlaid (pink). C: The manual tracing shown as a binary mask. D: The gold standard sphere map produced by running the Noiseless Algorithm on the segment mask, growing spheres only at pixels within the segment.

in understanding the performance of our system. This error visualization technique will be presented in detail in Section [5.2](#).

3.3 REAL IMAGE SEGMENTATION ALGORITHM

The Shells and Spheres framework presented thus far in this chapter provides a basic set of versatile image analysis tools. A number of methods for the use of these tools in analyzing real images have been explored, and the remainder of this chapter will present one effective multistage algorithm for optimization of a sphere map on real medical images. The presentation of this algorithm provides not only a functional Shells and Spheres real image segmentation system, but also insight into the advantages, limitations, and potential overall utility of the Shells and Spheres framework.

The algorithm described takes the form of a six step process, with steps 1-4 optimizing the sphere map, step 5 finding medial pixels, and step 6 producing a segmentation of the target object. Step 1 uses VSS gradient to detect boundaries and creates an initial approximation of the correct sphere map, from which acceptable statistical values can be obtained for use in subsequent steps. Step 2 utilizes the pronounced discrepancy in pixel variance between spheres that have incorrectly grown across boundaries and those that have not to reduce the size of incorrect spheres, placing them correctly within their appropriate image objects. Step 3 introduces specialized boundary indicators, known as *outposts*, to the image via boundary information extracted from population testing between spheres in adjacent image objects. These outposts then influence the radii of nearby spheres, resizing them to adhere to a consensus on boundary location. Step 4 revisits variance calculation using the current sphere map, which is more accurate than the sphere map previously available. The new variance measure is applied to spheres, encouraging them to fully grow within their respective image objects, effectively smoothing the radius image and sharpening its “boundaries” (as defined by all spheres that remain at the initial scale of 0, or a size of 1 pixel) simultaneously. Step 5 identifies medial pixels, i.e. those whose spheres touch at least two boundaries. Given a seed point within the target object, step 6 locates the nearest medial pixel and connects neighboring medial pixels, combining their corresponding spheres to produce a segmentation. The following sections describe each step in detail.

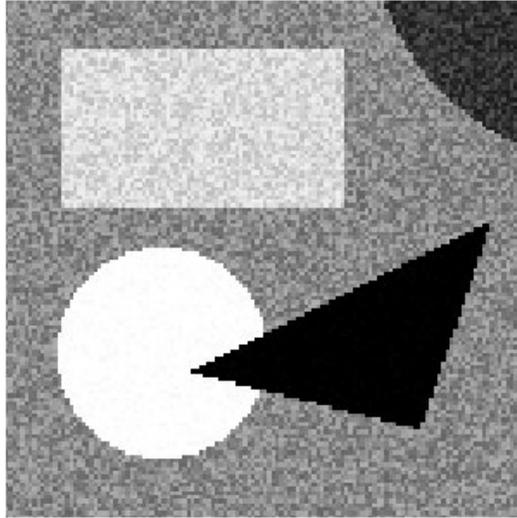


Figure 16: Our synthetic test image from Figure 14A with gaussian noise added.

3.3.1 Step 1: VSS Gradient-Based Radius Approximation

As previously illustrated, it is trivial to optimize the sphere map for a noiseless image by growing spheres until a new intensity value is detected. When analyzing real images, however, this approach will fail, because intensity variation due to noise may be indistinguishable from an object boundary, especially within small spheres. For illustrative purposes, we will consider the synthetic image from Figure 14A with gaussian noise added, shown in Figure 16.

Steps 2, 3, and 4 depend on there already existing a sphere map that is at least somewhat accurate, because those steps make use of secondary VSS, based on collections of existing spheres. For Step 1 to accomplish this, all spheres are first set to $r(x) = 0$ and are then allowed to grow until a persistent increase in VSS gradient magnitude (Eqn. 3.9) is detected over a range of sphere sizes. Unlike conventional gradient measured with a fixed-scale kernel, the VSS gradient depends locally on $r(\mathbf{x})$, and is based on the first-order moment of intensity normalized by the number of pixels in the sphere. Thus the VSS gradient can be expected to increase monotonically as a sphere grows past a boundary, since the first order moment

favors the outer pixels. A persistent increase in VSS gradient for g consecutive steps is sought, indicating that a boundary has been crossed. At this point $r(\mathbf{x})$ is reset to the scale just before the increase began. Empirically, it has been found that a value of $g = 5$ performs well for both MRI and CT cardiac data to achieve a reasonable first approximation of the optimized sphere map. Theoretically, the ideal value for g would change in proportion to the amount of uniform image noise, as a higher level of noise would cause more obfuscation of boundaries, impeding the detection of gradient. Note that any increase in gradient, regardless of magnitude, will count as one of the g increases. In the same manner, even the slightest decrease in gradient will break the chain of consecutive gradient increases being sought. Noise can cause random small fluctuations in gradient when no actual gradient is present. Because of the possibility of this anomaly occurring just before a growing sphere encounters an actual gradient, it is not unlikely that a sphere’s position at the end of Step 1 will be one or two steps short of the actual boundary. Since it is far more likely that noise could overcome the influence of an actual boundary on the VSS gradient within a small population of pixels (i.e. a small-scale sphere), it is equally likely that such an error would only occur early in a sphere’s growth, again deviating the sphere’s end position from its ideal radius. A less likely, but still possible, error is the case in which noise may cause enough of a decrease in perceived gradient to cause a break in the chain of g consecutive gradient increases caused by an actual boundary.

We found Step 1 to be effective at growing spheres past tissue inhomogeneity and noise. However, spheres may not stop exactly on the boundary because of the aforementioned effect of noise on the detection of gradient increase. Using VSS gradient to govern sphere growth can also fail completely for a sphere that encounters two opposing boundaries simultaneously, as their contributions to the gradient may cancel.

3.3.2 Step 2: Variance-Constrained Radius Reduction

After Step 1, three possible states exist for each sphere: The sphere can be too large, too small, or the correct size (i.e. it touches the nearest boundary but does not cross it). The most glaring error in the sphere map after Step 1 is the presence of large-scale spheres that

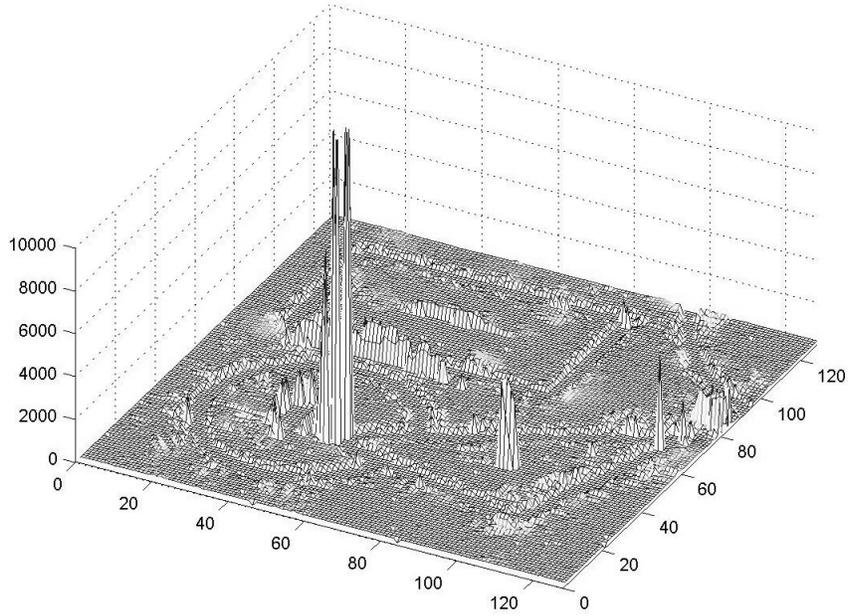


Figure 17: A colored height map of the internal variance of all spheres from Fig. 16 after algorithm Step 1, with the variance of each sphere assigned to its central pixel.

have incorrectly grown past boundaries. As previously mentioned, this type of error typically occurs when a growing sphere contacts multiple boundaries at once, which in many cases indicates that the sphere lies on the medial manifold. In such a case, the contributions to the VSS gradient from multiple boundaries may cancel, allowing the sphere to grow much larger than its correct radius. Such pixels, whose spheres cross object boundaries, have variance values orders of magnitude higher than those that remain within a single object. Figure 16 shows our synthetic image with added noise while Figure 17 shows a height map of corresponding variance values for all spheres in the same image after Step 1. The flat “floor” section of the height map is roughly the variance produced by the noise added to the image, which has a value near 200. Spheres crossing boundaries, however, have a larger variance $\sigma^2(\mathbf{x})$ (extending to a value of nearly 16000) than spheres that correctly remain within the object boundaries, a fact which can be exploited. To correct this error in the sphere map, all spheres with a variance above a certain threshold α_s are shrunk by decrementing $r(\mathbf{x})$

for each sphere $S_r(\mathbf{x})$ until $\sigma^2(\mathbf{x}) < \alpha_s$. The threshold α_s is set relative to μ_{σ^2} and σ_{σ^2} , the mean and standard deviation, respectively, of the variance of all the spheres in the current sphere map, as defined by

$$\alpha_s = \mu_{\sigma^2} + \beta_s \sigma_{\sigma^2}. \quad (3.14)$$

The positive constant β_s represents the number of standard deviations above the mean permitted for a sphere’s variance without the sphere being required to shrink. This parameter is not particularly sensitive, as the difference between spheres with erroneously high variance and all others is approximately two orders of magnitude. For the example in this section, a value of $\beta_s = 0.2$ was used.

This global threshold for $\sigma^2(\mathbf{x})$ is not ideal, because it assumes a constant expected variance throughout the image. This expectation is likely untrue, given factors such as tissue inhomogeneity and nonuniform noise, and future work may explore locally normalized thresholds. It should also be noted that $\sigma^2(\mathbf{x})$, and thus the threshold, depends on the current sphere map, $r(\mathbf{x})$, which is not yet fully optimized at this step. This deficiency is addressed by returning to variance in Step 4, once a more accurate $r(\mathbf{x})$ is available.

Reducing the radius of pixels with extremely high variance will correct a majority of the spheres that have incorrectly grown past boundaries. The updated variance height map after Step 2 is shown in Figure 18, which effectively illustrates the fact that the high variance regions in the centers of objects have been eliminated.

3.3.3 Step 3: Outpost Selection and Exclusion

A primary use of VSS is to differentiate regions on opposite sides of an object boundary. Following Step 2, many spheres are correctly sized and face each other across boundaries. Likely boundary candidates can thus be identified for a given sphere by finding a pixel \mathbf{y} in its $H_{r+1}(\mathbf{x})$ shell with a high value for $z_\mu(\mathbf{x}|\mathbf{y})$. As already discussed, and illustrated in Figure 8, such a sphere will detect growth past a boundary by finding itself “unlike” the $S^{-1}(\mathbf{y})$ population of spheres containing the pixel \mathbf{y} just across the boundary. The sphere at \mathbf{x} is said to place a *reflector* at such a location, a metaphorical construct denoting a vote by the sphere for the pixel \mathbf{y} as a point across its nearest boundary. Note that we do not

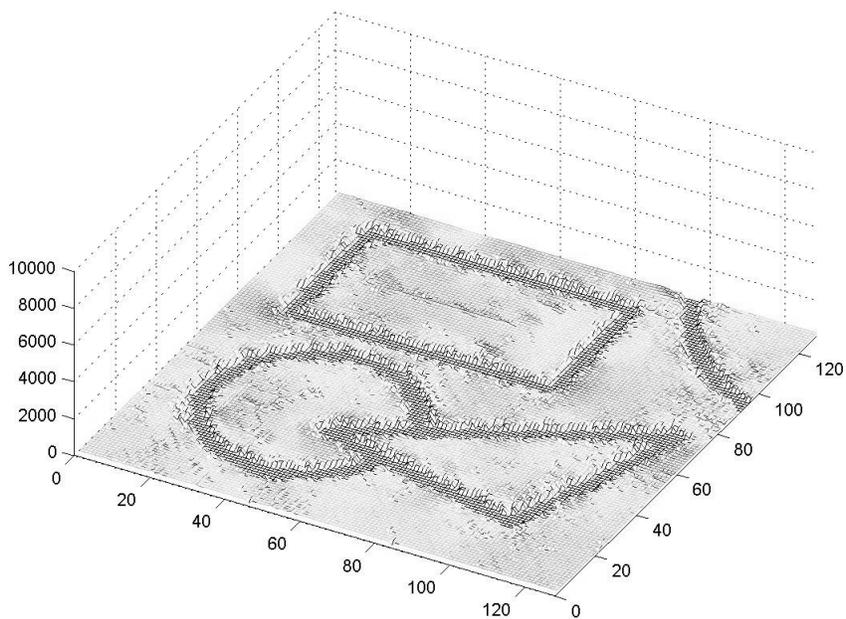


Figure 18: A colored height map of the internal variance of all spheres from Fig. 16 after algorithm Step 2, with the variance of each sphere assigned to its central pixel.

vote for this pixel as a boundary itself, since boundaries may exist between pixels, as in Figure 4. Instead, pixels on each side of a detected boundary are marked. Thus, referring again to Fig. 8, $S_r(\mathbf{x})$ could place a reflector at pixel \mathbf{y} . The set of reflectors placed by a given sphere $S_r(\mathbf{x})$ is denoted $K(\mathbf{x})$. In the present algorithm the constraint $|K(\mathbf{x})| = 1$ is applied, limiting each sphere to placing only one reflector, for reasons discussed below. This constraint leads to the definition of $K(\mathbf{x})$ as

$$K(\mathbf{x}) = \{\mathbf{y} : \mathbf{y} = \operatorname{argmax}_{\mathbf{y} \in H_{r+1}(\mathbf{x})} z_\mu(\mathbf{x}|\mathbf{y})\}. \quad (3.15)$$

Thus there are no thresholds or other parameters required in determining the placement of a sphere's reflector. If a boundary exists just beyond the outer shell of $S_r(\mathbf{x})$, it will be located at the pixel \mathbf{y} for which the highest $z_\mu(\mathbf{x}|\mathbf{y})$ is calculated. If multiple object boundaries exist

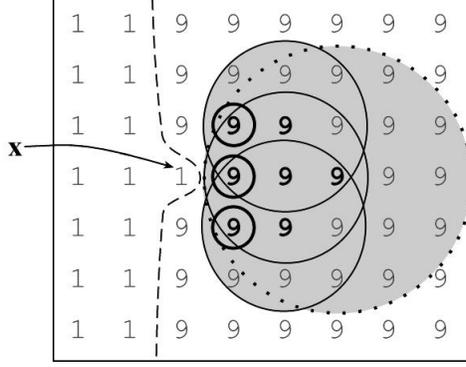


Figure 19: Illustration of $K^{-1}(\mathbf{x})$ containing 7 pixels (bold), each of whose sphere would place its reflector across the boundary at \mathbf{x} .

just beyond the outer shell of $S_r(\mathbf{x})$, the boundary producing the largest $z_\mu(\mathbf{x}|\mathbf{y})$ will be marked with a reflector.

Each pixel may contain reflectors placed by a number of spheres. The set of spheres that have placed reflectors at a pixel \mathbf{x} is defined as

$$K^{-1}(\mathbf{x}) = \{\mathbf{y} : \mathbf{x} \in K(\mathbf{y})\}, \quad (3.16)$$

invoking the same inverse notation used for $S^{-1}(\mathbf{x})$ in Eq. 3.10. Figure 19 shows a set of spheres placing their reflectors across a boundary at pixel \mathbf{x} .

$|K^{-1}(\mathbf{x})|$ is referred to as the *reflector count*, i.e. the number of reflectors that have been placed at location \mathbf{x} . For example, in Figure 19, the reflector count $|K^{-1}(\mathbf{x})| = 7$.

A reflector placed by \mathbf{y} at \mathbf{x} has an inherent direction governed by the vector $(\mathbf{y} - \mathbf{x})$. The vector sum of the directions of all of the reflectors at \mathbf{x} is denoted the *reflectance* $\mathbf{k}(\mathbf{x})$, defined by

$$\mathbf{k}(\mathbf{x}) = \frac{1}{|K^{-1}(\mathbf{x})|} \sum_{\mathbf{y} \in K^{-1}(\mathbf{x})} \frac{\mathbf{y} - \mathbf{x}}{|\mathbf{y} - \mathbf{x}|}. \quad (3.17)$$

This measure provides the average orientation of the $K^{-1}(\mathbf{x})$ population, which describes the direction normal to the boundary, pointing to the center of the region represented by $K^{-1}(\mathbf{x})$. Since it was decided that each sphere will contribute exactly one reflector, reflector

density and reflectance are normalized over the image. Therefore, reflector count can be used to differentiate between significant collections of reflectors correctly placed at boundaries and sparse distributions of reflectors incorrectly placed in the interior of objects. To denote pixels containing a significant number of reflectors, the term *outpost* is adopted, since such pixels serve as border markers, and in fact face each other across boundaries much the same way that military outposts of opposing armies face each other across a battle line.

The set of all pixels in an image chosen to be outposts is denoted by P . In the present algorithm this set is found in two steps. First, the set of *primary outposts* P' is established, containing all pixels with zero radius and at least κ reflectors, i.e.

$$P' = \{\mathbf{x} : |K^{-1}(\mathbf{x})| \geq \kappa, r(\mathbf{x}) = 0\}. \quad (3.18)$$

For all results presented in this dissertation, $\kappa = 4$. To increase the density of outposts along the boundaries, a set of *secondary outposts* P'' is generated, containing all pixels with zero radius that adjoin an outpost in P' and have at least λ reflectors, where $\lambda < \kappa$,

$$P'' = \{\mathbf{x} : |K^{-1}(\mathbf{x})| \geq \lambda, H_1(\mathbf{x}) \cap P' \neq \emptyset, r(\mathbf{x}) = 0\}. \quad (3.19)$$

For the results presented, $\lambda = 2$. By combining the sets of primary and secondary outposts, the set of all outposts,

$$P = P' \cup P'' \quad (3.20)$$

is formed.

Each outpost $\mathbf{y} \in P$ has a reflectance $\mathbf{k}(\mathbf{y})$. A sphere at \mathbf{x} can distinguish whether a given outpost is on its side of the boundary, constituting a *friendly outpost*, or the other side of the boundary, constituting an *enemy outpost*, based on the direction of the outpost's reflectance. The set of enemy outposts (those with reflectance facing \mathbf{x}) within the sphere of radius $r(\mathbf{x})$, is defined as

$$E_r(\mathbf{x}) = \{\mathbf{y} : \mathbf{y} \in P \cap S_r(\mathbf{x}), \mathbf{k}(\mathbf{y}) \cdot (\mathbf{y} - \mathbf{x}) < 0\}, \quad (3.21)$$

where the sign of the dot product determines the direction of $\mathbf{k}(\mathbf{y})$ relative to \mathbf{x} . The exceedingly unlikely case of an outposts with $\mathbf{k}(\mathbf{y}) = 0$ can be considered a friendly outpost, as it does not indicate a boundary facing the sphere.

In governing the growth of a sphere, enemy outposts are to be avoided, while friendly outposts can be included. More specifically, enemy outposts should stop the growth of spheres, as they represent a different image region than the one in which the sphere resides, while friendly outposts do not. Step 3 uses the number of enemy outposts to adjust the sphere size as follows: If the pixel contains no enemy outposts in its next shell out, $S_{r+1}(\mathbf{x})$, the sphere grows until it does. That is

$$\text{If } |E_{r+1}(\mathbf{x})| = 0, \text{ increase } r(\mathbf{x}) \text{ until } |E_{r+1}(\mathbf{x})| > 0.$$

If the number of enemy outposts included in the set $S_r(\mathbf{x})$ is greater than γ , the radius is decreased until this is no longer true, i.e.

$$\text{If } |E_r(\mathbf{x})| > \gamma, \text{ reduce } r(\mathbf{x}) \text{ until } |E_r(\mathbf{x})| \leq \gamma.$$

In the present implementation, $\gamma = 2$. This value prevents lone pixels that have been improperly labeled as outposts from incorrectly causing spheres to shrink.

After Step 2, in which incorrectly large spheres have been adjusted to a more correct size, the most pressing problem with the sphere map is the scattered effects of noise on $r(\mathbf{x})$. Step 3 focuses on spheres that have incorrectly stopped growth at image noise, instead of growing to reach an actual boundary. Outposts provide a means for correcting these errors, as well as for adjusting spheres that have grown slightly too large or remained too small relative to their nearest boundary. The effect of these outpost-driven operations is that significant densities of reflectors placed by correctly-sized spheres along boundaries are used to govern the size of other spheres, sweeping incorrect reflectors from the within objects to the boundaries (since a sphere’s reflector is redistributed when its radius is altered). With spheres placing only one reflector each, some pixels along object boundaries may remain unmarked, leading to a slightly sparse collection of outposts along boundaries. This state will not adversely effect the evolution of our sphere map, due to another advantage of our spherical operator design. Because most spheres are large relative to the spacing of outposts along the boundary, their growth will be stopped and they will not “leak” or “bleed” across boundaries, as conventional deformable contours are prone to do.

Much of the sporadic noise in $r(\mathbf{x})$ is removed by Step 3, continuing the work of Step 2 by correcting additional spheres that had been crossing object boundaries.

3.3.4 Step 4: Variance-Constrained Scale Growth

At this point in the analysis, our sphere map has achieved a configuration generally representative of the shapes within the image, but it still retains adverse effects from noise and sub-optimal boundary detection. Although Step 3 results in a reasonably accurate $r(\mathbf{x})$, some spheres still may not reach boundaries, due to pixels incorrectly labeled as outposts. These false outposts will stop spheres in the interior of image objects, leading to potential errors in segmentation. Since the issue at hand is spheres stopping their growth short of the nearest boundary, an added measure to force spheres to grow maximally within their image objects is required. To facilitate this growth variance is used a second time, calculated in the same manner as in Step 2, but used to force growth, rather than limit it.

The global variance threshold for growth, α_g , similar to the threshold described in Section 3.3.4, is calculated as

$$\alpha_g = \mu_{\sigma^2} + \beta_g \sigma_{\sigma^2}. \quad (3.22)$$

This threshold is used to smooth the boundaries in the radius image by forcing spheres to grow up to the actual boundary using the current, more accurate variance representative of the intensity variance of objects within the image. Spheres are grown as large as possible without their internal variance exceeding threshold α_g . Note that there are two differences between the α_s threshold used in Step 2 and this α_g threshold. First, it is defined by a new parameter β_g , which may or may not equal β_s from Equation 3.3.2. Second, this threshold uses global values based on the current improved state of the sphere map, which has changed since Step 2. Due to further optimization of the sphere radii, α_g represents a more appropriate variance threshold than it did in Step 2, allowing Step 4 to perform more sensitive region-approximating operations. Invoking the variance threshold, the value of $r(\mathbf{x})$ is incremented for all spheres while their internal variance $\sigma^2(\mathbf{x}) < \alpha_g$. This creates a radius image defining spheres which more accurately match the contours of the objects in the image.

The effects of this final measure are pivotal, smoothing the sphere map $r(\mathbf{x})$ by fixing incorrectly stopped spheres. In addition to smoothing the $r(\mathbf{x})$ image, Step 4 also sharpens and accurately characterizes the boundaries by ensuring that spheres grow to reach them. We now have the optimized set of radius values for use in the segmentation routine to be described in the following section. At this point the sphere map is considered optimized.

3.3.5 Step 5: Medial Pixel Identification

Given the optimized sphere map $r(\mathbf{x})$, the next goal is to extract medial pixels. To facilitate this, a dense set of boundary pixels B is first defined as those pixels from the sphere map with radius 0 or 1,

$$B = \{\mathbf{x} : r(\mathbf{x}) \leq 1\}. \quad (3.23)$$

The sets $S^{-1}(\mathbf{b})$ for all boundary pixels $\mathbf{b} \in B$ can be used to find pixels on the medial manifold, whose spheres touch two boundaries while still lying completely within the object. Recall that the $S^{-1}(\mathbf{b})$ set for pixel \mathbf{b} contains all spheres in the sphere map that themselves contain pixel \mathbf{b} (Eqn. 3.10). Given a correct sphere map, this set will necessarily contain at least one sphere that touches both the boundary that pixel \mathbf{b} borders as well as an opposing boundary across the object region (and also across the sphere) from pixel \mathbf{b} . Figure 20(left) shows such a medial pixel (labeled “m”) on the medial manifold of an object of intensity 1, between two regions of intensity 9.

To find such medial pixels within $S^{-1}(\mathbf{b})$, we first define an orientation $\mathbf{s}(\mathbf{b})$ roughly orthogonal to the boundary as the vector sum of the normalized offsets relative to \mathbf{b} for pixels within $S^{-1}(\mathbf{b})$ as

$$\mathbf{s}(\mathbf{b}) = \frac{1}{|S^{-1}(\mathbf{b})|} \sum_{\mathbf{y} \in S^{-1}(\mathbf{b})} \frac{\mathbf{y} - \mathbf{b}}{|\mathbf{y} - \mathbf{b}|}. \quad (3.24)$$

For each boundary pixel $\mathbf{b} \in B$, the pixel $\mathbf{m} \in S^{-1}(\mathbf{b})$ furthest from the boundary along $\mathbf{s}(\mathbf{b})$ is identified as a medial pixel, as depicted in Fig. 20 (left). The set of all medial pixels M is thus

$$M = \{\mathbf{m} : \mathbf{m} = \underset{\mathbf{y} \in S^{-1}(\mathbf{b})}{\operatorname{argmax}}((\mathbf{y} - \mathbf{b}) \cdot \mathbf{s}(\mathbf{b})), \mathbf{b} \in B\}. \quad (3.25)$$

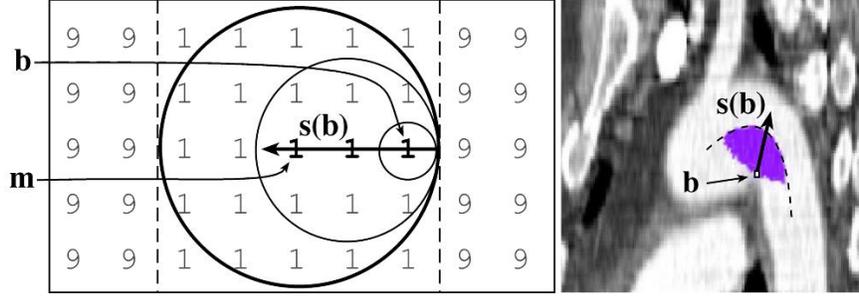


Figure 20: **Left:** Diagram of an object with intensity 1 between two regions of intensity 9, showing set $S^{-1}(\mathbf{b})$ of pixels (bold) whose spheres contain pixel \mathbf{b} . This set produces an $\mathbf{s}(\mathbf{b})$ vector (see text) along which the furthest bold pixel \mathbf{m} is the center of a medial sphere (circle in bold) touching both boundaries (dashed lines). **Right:** Image of a 2D slice through a CT scan of the aorta with contrast showing an actual $S^{-1}(\mathbf{b})$ set (purple/grey), the resulting $\mathbf{s}(\mathbf{b})$ vector, and the medial manifold (dashed curve) on which the furthest sphere along $\mathbf{s}(\mathbf{b})$ must lie.

Figure 20(right) shows an actual $S^{-1}(\mathbf{b})$ set for a pixel \mathbf{b} on the boundary of the aorta in a CT scan with contrast.

Selecting a single pixel from each $S^{-1}(\mathbf{b})$ set overlooks a potentially large number of additional medial pixels on the outer edge of each set, especially for concave boundary points such as those in Figure 20, but no reliable method has yet been found for extracting only medial pixels from an $S^{-1}(\mathbf{b})$ set. This is in part due to the high variation in shape of $S^{-1}(\mathbf{b})$ sets, which is a product of the local variability of the image objects in question. A standard distance from the boundary for each medial pixel cannot be assumed, as this distance is dependent upon the local thickness of the object and the relative orientation of its boundaries. One can, however, be certain that each $S^{-1}(\mathbf{b})$ set contains a minimum of one medial pixel, as the center of the largest sphere in the direction roughly orthogonal to the boundary is necessarily a medial pixel. The set M derived taking advantage of this fact is a sparse but reliable set of pixels on the various medial manifolds within the image.

3.3.6 Step 6: Medial Flood-Fill Segmentation

To segment a particular object, a seed pixel $\mathbf{p} \in M$ on that object's medial manifold is needed. To find it, a sample pixel is manually selected by the user, and a search is conducted for the closest medial pixel by iterating through successive shells moving radially outward from the selected pixel. The first medial pixel encountered is accepted as \mathbf{p} . A flood fill operator is then used to find a connected subset $C \subseteq M$ containing medial pixels that are connected to \mathbf{p} . Pixels belonging in C are found iteratively using a series of sets C_i starting with C_0 , a set containing just the seed pixel \mathbf{p} . At each subsequent step $i + 1$, the set C_{i+1} is created by adding medial pixels within a radius ϕ of pixels already in set C_i . More precisely, C_i is defined inductively as

$$C_0 = \{\mathbf{p}\} \tag{3.26}$$

$$C_{i+1} = \{\mathbf{x} : \mathbf{x} \in M, S_\phi(\mathbf{x}) \cap C_i \neq \emptyset\}. \tag{3.27}$$

For the results presented, scale ϕ was dynamically set to $\phi(\mathbf{x}) = r(\mathbf{x})/2$, as this causes the algorithm to search half-way from the medial manifold to the boundary for new medial pixels to include, therefore staying within the designated object. When a final step f adds no new pixels, such that

$$C_f = C_{f-1}, \tag{3.28}$$

the flood-fill is complete, as the set of connected medial pixels within the object is the current pixel collection, or

$$C = C_f. \tag{3.29}$$

The union of the set of spheres centered at these medial pixels effectively segments the object by including all of the pixels designated as within the object. These spheres, centered on the medial manifold, extend to all points on the boundary.

3.4 DISCUSSION

The Shells and Spheres framework for image analysis and the associated n -dimensional segmentation algorithm introduced represent a novel system designed to facilitate image segmentation via variable-scale region detection. Our minimally supervised segmentation algorithm focuses on local statistical measures using our spherical operators to overcome the effects of scanner noise, partial-volume effects, and tissue inhomogeneity, which hinder many statistical segmentation approaches from performing well, especially on MRI data. See [28] for a review of segmentation techniques for MRI data that is especially relevant to our own experiments, which we will describe in the following chapters. By performing statistical comparisons on adjacent spheres across boundaries, we restrain our comparisons to local regions, effectively assuming that the mean tissue intensities of objects within the image are spatially varied as a means of surmounting these obstacles. [9]

A properly optimized sphere map, with each sphere grown to a radius appropriate to its position relative to object boundaries within the image, can be thought of as a manifold cut through conventional scale space. The magnitude of each point along this manifold is defined by the corresponding value of the distance map of the image. The population statistics used on spheres are *variable-scale* in the sense that they remain valid and statistically representative of the sphere, and thus the variably-sized local regions around a given pixel, as the sphere moves through scale space by changing size. This approach facilitates simultaneous medial manifold detection along with segmentation because the medial manifold, as described by Blum, is defined by the centers of spheres grown as large as possible within an object. [23] Perhaps most importantly, unlike Gaussian blurring, commonly used in conventional multi-scale analysis [24, 6], Shells and Spheres operators preserve sharp boundaries with increasing scale.

The Shells and Spheres framework as presented thus far provokes a number of concerns. Computation time and memory are obvious limitations to the use of the Shells and Spheres framework, given the size of 3D medical images and the fact that spheres are maintained at each image pixel. Our current computational needs, which will be discussed in subsequent sections, are met by a quad opteron system with 32GB of RAM. However, we submit that

by Moore's Law [29] and current trends in computational innovation, the processing power and memory of standard computers will soon enable processing of large 3D images on standard desktop systems. Potential shortcomings of our system beyond computational concerns include the fact that our initial algorithm requires a number of parameters to be set manually, producing sub-optimal results due to our inability to determine ideal parameters for a given image. The next chapter will introduce a novel method of parameter optimization, made computationally tenable by utilizing a representative 2D image to produce effective parameters for segmentation of a 3D data set. We will explore this algorithm parameter optimization technique in detail, and then present results and validation using 3D ovine cardiac MRI data used in research for quantitative shape analysis of the Right Ventricular Outflow Tract (RVOT).

4.0 ALGORITHM PARAMETER OPTIMIZATION

While initial studies applied our algorithm to segmentation problems with promising results, algorithm parameters were manually derived for the data tested, limiting the systems general accuracy, versatility, and clinical appeal. For example, when the parameters used to successfully segment the aortic arch in [25] were applied to the RVOT data used in the validation studies described below, the automated segmentation failed completely because the parameters were dependent on imaging modality and target structure. Additionally, user interaction was required at multiple points in the segmentation procedure, both to set algorithm parameters at the onset of analysis, and to seed the medial flood-fill which produces a segmentation after the sphere map had been optimized. We will show that a greater degree of automation can be instituted into our algorithm using a new approach to simple manual initialization. We also propose that this approach may be applied to improve other established automated segmentation algorithms.

The time and technical expertise required to manually set parameters deters most end-users, particularly clinicians, from adopting image analysis technology. Recent research has addressed this problem by creating software systems designed to appeal to the clinician by simplifying the interface and adding useful feedback for parameter initialization. [3] However, users may still be required to learn algorithm specifics to set parameters, which may be daunting or unacceptably time-consuming. A viable solution to this problem is to provide a means of automatically optimizing segmentation parameters that requires minimal user input, in a form that is acceptable to the clinical professional. Since a single segmentation procedure may require hours of computational time for 3D images, iterative optimization

involving the computation of 3D segmentations is untenable. Additionally, limiting user interaction to the onset of analysis is preferable, also because of the time required for each 3D segmentation.

We have developed a technique to automatically optimize parameters for our segmentation algorithm, given a gold standard segmentation. In order to further realize a clinically appealing segmentation system as described above, we implemented a minimally-interactive approach to initialize the optimization procedure. The next section will present the optimization technique in detail, followed by a description of the initialization method developed to allow our system to automatically calculate optimization parameters from an intuitive, clinically accessible user initialization process.

4.1 2D OPTIMIZATION FOR 3D SEGMENTATION

To improve segmentation accuracy and adapt our algorithm to new data sets, two key algorithm parameters, β_s and β_g , need to be optimized. These parameters control variance thresholds regulating sphere shrinkage and growth, respectively, as described in Section 3.3. Other parameters that have significantly less impact on the resulting segmentation were left at the default values given in Chapter 3. Those values were found to be effective on all data sets tested, spanning multiple images, modalities, and target structures.

The parameters β_s and β_g , used to calculate the variance thresholds in steps 2 and 4, can be optimized to maximize the resulting agreement, defined by the Dice Similarity Coefficient (DSC) [30], between the automated segmentation and a gold standard manual segmentation. The DSC, briefly, is the ratio of twice the number of intersecting pixels between the two objects and the the combined number of pixels within both objects. This coefficient is always between 0 and 1, with a value of 0 resulting from no intersection and 1 resulting from complete overlap. We denote as D the coefficient measured given particular values for β_s and β_g ,

$$D = DSC(\beta_s, \beta_g). \tag{4.1}$$

To prove the efficacy of the DSC as our means of comparing segmentations, relative to other measures such as volumetric overlap or average symmetric surface distance [31], we present a small comparison study. Figure 21 shows normalized values for each of these three segmentation accuracy measures produced by comparing ninety automated segmentations to gold standard manual segmentations. The graph illustrates the roughly proportional relationship between all three measures, with the DSC values falling between the other two measurements. Hence, the DSC is equivalent to these other common means of measuring segmentation accuracy.

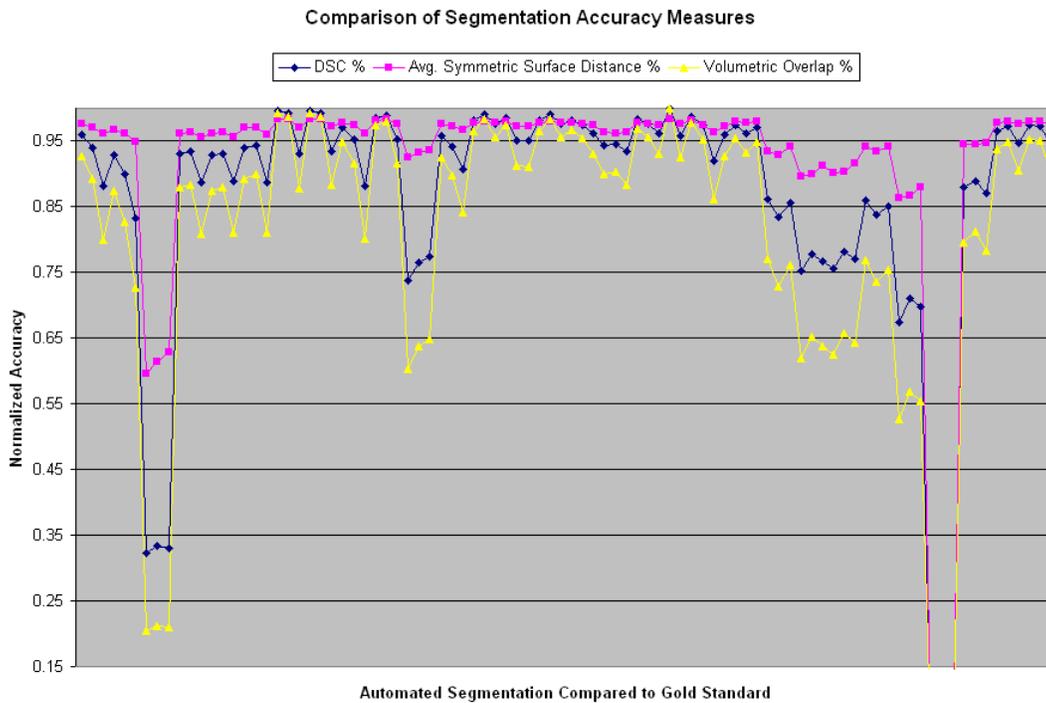


Figure 21: A graph showing the normalized DSC, Average Symmetric Surface Distance, and Percent Volumetric Overlap for ninety automated segmentations compared to manual gold standards.

A brute-force optimization approach, testing every combination of parameter values within given ranges at a given increment, was applied to deduce the optimal parameter

values for a given data set. The optimal parameter values are defined by the maximal DSC, as

$$D_{max} = \underset{\beta_s \in \beta_{si} : \beta_{sf}, \beta_g \in \beta_{gi} : \beta_{gf}}{\operatorname{argmax}} (DSC(\beta_s, \beta_g)). \quad (4.2)$$

The value of β_s is discretely sampled at intervals of j over a range $\beta_{si} : \beta_{sf}$, where β_{si} indicates β_s *initial* and β_{sf} indicates β_s *final*. Similarly, β_g was sampled over a range of $\beta_{gi} : \beta_{gf}$. This brute-force data gathering approach produces a DSC space as seen in Figure 22. In this example, the maximal DSC value of 0.96 on the upper ridge of the curve, as indicated by an arrow, corresponds to the parameter values $\beta_s = 2.1$ and $\beta_g = 0.7$.

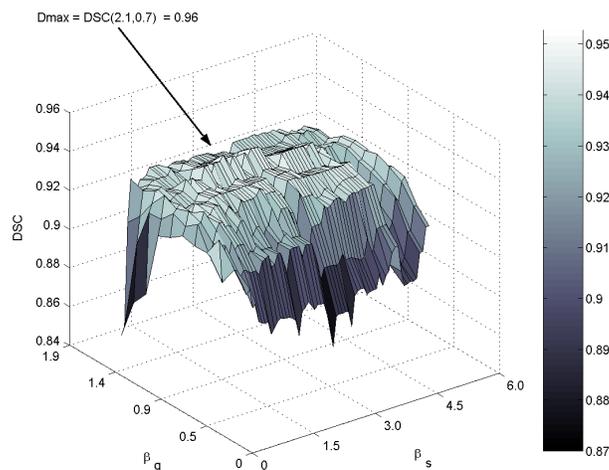


Figure 22: Example DSC curve produced by a 2D parameter optimization with $\beta_s \in 0.0 : 6.0$, $\beta_g \in 0.0 : 1.9$, $i = 0.1$. The optimal DSC value 0.96, produced by the parameter set $(\beta_s = 2.1, \beta_g = 0.7)$, is indicated with an arrow.

Though effective at producing parameters that result in accurate segmentations, this optimization technique can be computationally time-consuming. The 2D image used to produce the optimization in Figure 22 was an MRI image of 256x256 pixel resolution. A segmentation of this single 2D image via our Shells and Spheres system took about 1 minute. The curve shown represents 61 values of β_s and 20 values of β_g , with each combination of

parameter values producing a different segmentation. Thus, it required 1,220 analyses to create, or 20.3 hours of computation. Luckily, we can reduce computation time by applying dynamic programming [32] techniques, separating our algorithm steps and recalculating them only as necessary. For example, the parameters we are optimizing have no bearing on algorithm Step 1, a particularly time-consuming step. Thus, rather than computing this step for each set of parameters, 1220 times in the example above, we need only compute this step once on a given image and save the resulting sphere map. Likewise, iterations over the range of β_g values corresponding to the same value of β_s require only computation of algorithm steps 4-6, providing that the sphere map resulting from Step 3 has been saved.

These dynamic programming solutions have reduced the time of brute force optimization on 2D images to under 30 minutes in most cases. However, recall that a single 3D segmentation can take a number of hours to complete. Because of this, even with our dynamic programming solution, any optimization that requires repeated testing of parameters by performing a 3D segmentation is unacceptable because of the computational time required for each 3D analysis.

To address this computational concern of achieving effective parameter optimization for 3D data, more efficient optimization procedures from numerical methods or artificial intelligence were considered to reduce the time required for optimization. However, many of these are incompatible with our parameters, due to the fact our algorithm parameters are dependent on the state of the sphere map at each step of analysis, and thus represent a highly non-linear system. Instead, we hypothesized that parameters calculated in the proven brute-force manner on a representative 2D slice selected from a 3D data set would yield optimal parameters similar to those that would be calculated from a full 3D optimization, in a fraction of the time. Additionally, optimizing on a 2D slice requires only a 2D manual tracing as a gold standard, rather than a time-consuming manual 3D segmentation.

4.1.1 Preliminary Experimental Validation

To test the effectiveness of our novel parameter optimization technique on 3D analysis and segmentation, a preliminary study was performed on one of our 3D ovine cardiac MRI scans.

The data set can be seen in the upper row of Figure 26.

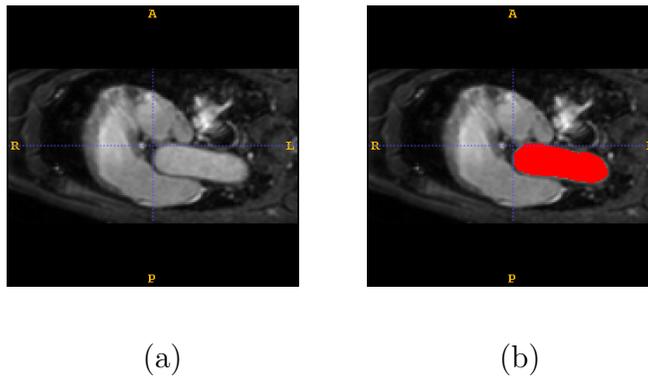


Figure 23: (a) 2D slice of MR cardiac data and (b) manual segmentation (red or dark) used for parameter optimization in the 3D segmentation study.

A 2D slice, shown in Figure 23, was manually selected and a target object traced by an expert user. The slice and its 2D contour were used to perform a parameter optimization, searching the space of algorithm parameters β_s and β_g to yield the segmentation most similar to the manually traced contour, producing the DSC data space shown in Figure 24. The slice, and the selected object within it (the aortic arch), were chosen because the thin boundaries between adjoining larger objects simulated conditions for the 3D segmentation of the right ventricle adjoining other cardiac structures. The impact of the slice chosen on the optimization, and ultimately on the segmentation, is explored further in a later section of this dissertation.

The 2D optimization yielded optimal parameter values of $\beta_s = 1.9$ and $\beta_g = -0.2$ for segmenting the 2D slice, which produced a DSC of 0.97 similarity to the 2D manual tracing. This set of parameters determined on a 2D slice, when applied to segmentation of the full 3D data set, enabled our Shells and Spheres algorithm to produce the 3D segmentation of the right heart shown anatomically labeled in multiple orientations in Figure 25 and overlaid on the input MRI data in Figure 26.

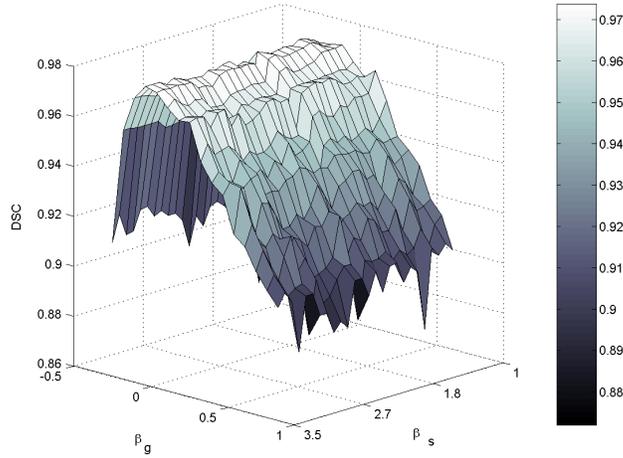


Figure 24: DSC curve produced by a 2D parameter optimization with $\beta_s \in 1.0 : 3.5$, $\beta_g \in -0.5 : 1.0$, $i = 0.1$. The optimal DSC, 0.97, is indicated with an arrow at values $(\beta_s = 1.9, \beta_g = -0.2)$

To determine the accuracy of the resulting 3D segmentation, a validation study was conducted, comparing it to three manual 3D segmentations performed by expert users. These segmentations were performed by manually tracing the 3D data set, slice by slice, through the entire right heart. Each manual segmentation was produced by a different user. Table 1 shows the DSC values for our optimized automated 3D segmentation when compared to the expert manual segmentations, as well as the DSC values for the manual segmentations when compared to each other. It can be seen that our automated system produced a segmentation with agreement values between 0.83 and 0.86 to manual segmentations. It should be noted that a DSC of 0.70 is considered satisfactory spatial overlap in the literature [33, 34, 35], although the definition of sufficient accuracy is, of course, specific to the application. While the manual segmentations produced slightly higher agreement to each other than to the automated segmentation, variation between the manual segmentation was due, at least in part, to the difficulty defining the extent of the “right heart” along the continuum of the circulatory system. Subject 3, who showed greater similarity to the automated segmentation

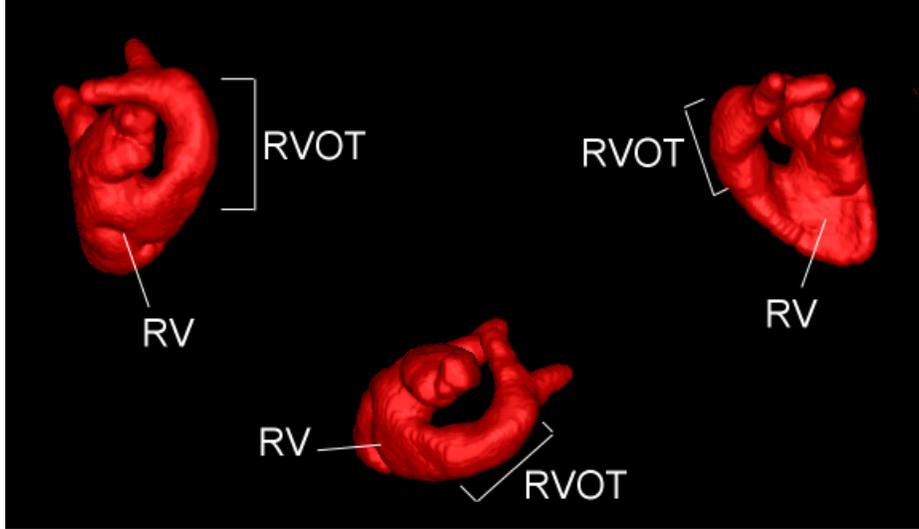


Figure 25: Surface model of the automated segmentation of the right heart with the RV and RVOT labeled, shown from three different perspectives.

than the other subjects, elected to include less of the branching vasculature connected to the main cardiac structures, which led to slightly greater agreement with the automated segmentation and less agreement with the other manual segmentations. Despite the variation in manual segmentations, our algorithm still produced highly reliable segmentation results, indicating promise for our technique of calculating 3D segmentation parameters using 2D image and tracing data.

We now turn our focus to exploring the optimization procedure more deeply. Though effective, the optimization procedure itself requires input parameters, including optimization ranges and seed points, and thus segmentation could not immediately follow optimization without user interaction in our present implementation. The next section will present the steps taken to automate the optimization procedure, reducing the user input for the entire segmentation process to a single 2D manual tracing from the selected 2D slice at the onset of analysis.

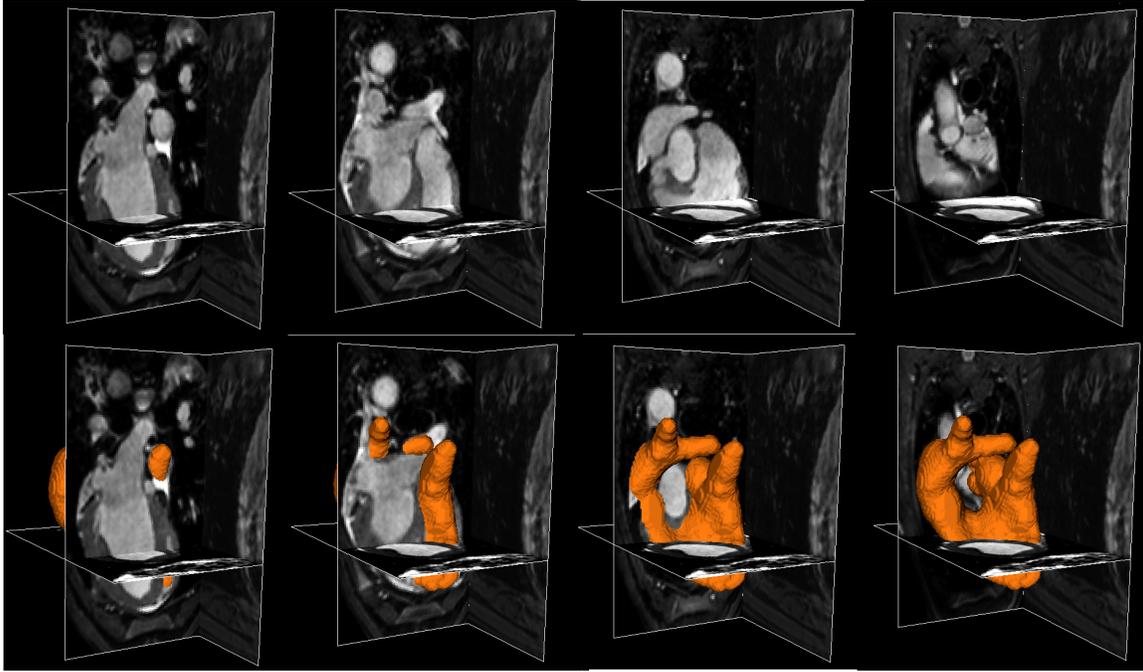


Figure 26: Top row: The MR data set with one plane moved incrementally backwards through the RV. Bottom row: Images corresponding to those above, with the segmentation surface model overlaid on the source data.

Table 1: Table of DSC values comparing segmentations produced by 3 independent subjects and our Shells and Spheres (S&S) algorithm.

	Subject 1	Subject 2	Subject 3
S&S	0.83	0.84	0.86
Subject 1	*	0.91	0.88
Subject 2	*	*	0.89

4.1.2 Automating Parameter Optimization

A fundamental concept in this research was to design a system to take full advantage of the expected user’s specific knowledge, requiring no information from the user that does not come directly from professional expertise. With the addition of our parameter optimization scheme a necessary *a priori* input to the system was introduced, consisting of a 2D manual tracing on a selected slice from the particular 3D data set to be analyzed. The two components of this input are slice selection, and 2D manual anatomical tracing, both of which fall well within the realm of expertise of our target user: a clinical professional. From this simple initial input, information can be extracted that will provide all parameter values necessary for algorithm parameter optimization and subsequent 3D segmentation.

To perform parameter optimization, we require a range of values for each parameter over which we will test algorithm performance. Determination of this range by the user adds a pre-processing task to the analysis, and necessitates an understanding of the statistical properties of the image and of expected algorithm performance. Though the user no longer selects segmentation algorithm parameters, selection of the optimization range itself requires a level of algorithmic expertise outside the realm of professional expertise of most users. Additionally, seed points for segmentation must be provided subsequent to analysis, requiring the user to return to the system after analysis to provide additional interaction. While anatomical seed points are not outside the realm of a clinician’s expertise, added time and responsibility is a deterrent to the adoption of any new system. Leaving these tasks to the user adds interaction time, combining with the added complexity of expert supervision to result in a less attractive system from a clinical standpoint. The following sections will describe how effective optimization ranges and seed points can be extracted from our 2D *a priori* data, allowing our system to function autonomously after our simple initialization procedure.

4.1.2.1 Determining Optimization Ranges Though the technique for parameter optimization introduced and tested in the previous section removes the burden of manual parameter selection from the user and yields accurate 3D segmentations, parameters of the

Table 2: Parameters necessary to perform algorithm parameter optimization, along with descriptions.

Parameter	Description
β_{si}	The minimum value in the range of β_s to be tested, corresponding to a variance threshold value of α_{si} for a given image and sphere map state.
β_{sf}	The maximum value in the range of β_s to be tested, corresponding to a variance threshold value of α_{sf} for a given image and sphere map state.
β_{gi}	The minimum value in the range of β_g to be tested, corresponding to a variance threshold value of α_{gi} for a given image and sphere map state.
β_{gf}	The maximum value in the range of β_g to be tested, corresponding to a variance threshold value of α_{gf} for a given image and sphere map state.
j	The increment by which parameter values are sampled through the given range.

optimization procedure itself must still be manually instantiated. Initializing a 2D optimization requires the parameters displayed in Table 2, originally introduced in Section 4.1.

While these values, defining the range of β_s and β_g over which optimization will be performed, may be familiar to a user with numerical optimization experience, their ideal values are dependent on a number of esoteric factors. The ideal optimization range is dependent on the image in question and its statistical properties. Recall that β_s and β_g are variables indicating the number of standard deviations from the mean in equations 3.3.2 and 3.3.4, resulting in variance thresholds α_s and α_g . The calculation of these thresholds is dependent on the global mean variance of all spheres in the sphere map at its current state, which is its

state after algorithm Step 2 (Subsection 3.3.2) in the case of α_s , and after algorithm Step 4 (Subsection 3.3.4) in the case of α_g . Since the global mean variance for a given sphere map configuration cannot be calculated until the algorithm is in progress, the possible values of the threshold produced cannot be calculated before run-time. Optimal threshold values, and therefore effective optimization ranges, have been observed to be highly varied between source images. These facts make determination of the an effective range over which to perform parameter optimization a challenging problem. For the segmentation validation study presented later in this dissertation, tests performed by experienced users allowed optimization ranges to be determined experimentally, relying on educated assessments of initial range settings based on user expertise.

To fulfill our goal of a fully automated system, we must determine all values necessary for parameter optimization and segmentation from our 2D manual tracing. Here we propose our technique for calculating effective search ranges for parameter optimization. These ranges are effective, though further innovation to determine more concise search ranges would further reduce parameter optimization time. Recall the behavior of parameter optimization over the parameters β_s and β_g . The β_g parameter forms a ridge in the DSC search space produced through brute-force parameter optimization. A parameter value quickly becomes ineffective as it moves away from this optimal ridge. This behavior is to be expected, since the algorithm step using β_g is the last in sphere map optimization, which was designed to fine-tune sphere radii to meet boundaries and is followed by no corrective steps. In contrast, in all cases studied, effective values of β_s exist over a large range of data, provided the value of β_g lies on the aforementioned optimal ridge. This is also consistent with expectations, as β_s represents some value within a wide range of variance values between the variance of a sphere remaining in its image object and one that has crossed multiple boundaries. Mistakes in the algorithm step utilizing this parameter may also be corrected at a later stage in the algorithm.

Given the theoretical and observed behavior of these parameters, calculation of an effective range $[\beta_{gi} : \beta_{gf}]$ is the more pressing concern. We begin by modeling the statistics of our spheres using the given 2D initialization tracing. Essentially, β_g is meant to provide a variance threshold α_g that represents the variance of spheres at their correct radius. Being a global value, α_g cannot be correct for all spheres, but will be optimized to a value performing

effectively for the largest number of spheres in the target object. We can approximate an effective value of α_g by finding the mean of the variance values of correctly sized spheres within our traced 2D object. We obtain correctly sized spheres within our target object by applying the Noiseless Sphere Map Optimization Algorithm to the mask created by the 2D tracing provided by the user. This streamlined algorithm, described in Section 3.2, grows spheres at each pixel within the contour, stopping each sphere’s growth at the radius before encountering a pixel outside the contour. We limit the spheres to those with a radius greater than 3, creating a set R of spheres within our target object such that

$$R = \{\mathbf{x} : r(\mathbf{x}) > 3\}. \quad (4.3)$$

The lower bound on radius ensures that the spheres contain enough pixels to be a statistically valid sample population. We calculate ρ to be the mean of the internal variance values of spheres within R ,

$$\rho = \frac{1}{|R|} \sum_{\mathbf{x} \in R} [\sigma^2(\mathbf{x})]. \quad (4.4)$$

We then find the standard deviation of the variance values of spheres in R as

$$\sigma(\rho) = \frac{1}{|R| - 1} \sum_{\mathbf{x} \in R} [(\sigma^2(\mathbf{x}) - \rho)^2]^{\frac{1}{2}}. \quad (4.5)$$

Using this standard deviation, we calculate an optimization range $[\alpha_{gi} : \alpha_{gf}]$ such that

$$\alpha_{gi} = \rho - \sigma(\rho) \quad (4.6)$$

and

$$\alpha_{gf} = \rho + \sigma(\rho). \quad (4.7)$$

Corresponding values of β_{gi} and β_{gf} can be determined during run-time, calculating the values of β_g that cause α_g to equal α_{gi} and α_{gf} . We compared α_{gi} and α_{gf} values calculated by this technique using an axial initialization slice to the optimal values determined with user-defined ranges from our segmentation studies, and observed that the $[\alpha_{gi} : \alpha_{gf}]$ range calculated included the optimal values of α_g for all data sets.

An effective range for optimization of β_s is more difficult to approximate, because the only definitive trend in the calculated optimal threshold values of α_s that has been observed is that they are always higher than corresponding values of α_g . We propose an approach similar to the one used to calculate an effective range for α_g , altered to reflect the theoretical properties of α_s . Since optimal values of α_s have been observed to fall significantly above (usually at least 3 times larger) the corresponding value of α_g , we conclude that an effective optimization range should be based on the heightened variance of spheres after they cross boundaries. To this end, we grew spheres in our axial initialization tracings larger than their correct radius while monitoring their variance values. We focused our test on medial spheres because they are the spheres directly responsible for segmentation, as explained in Chapter 3. We used the set M from Section 3.3.5, which in this case included the medial spheres from our 2D tracing. We recorded the mean variance ν of all spheres in M for each radius increment δ above their correct radius value. For example, $\delta = 1$ indicates that the medial spheres were grown 1 radius larger than their correct value. We calculated ν as

$$\nu(\delta) = \frac{1}{|M|} \sum_{\mathbf{x} \in M} [\sigma^2(\mathbf{x})]. \quad (4.8)$$

We compared the recorded values of ν to the optimal α_s threshold values calculated in our validation studies using manually determined optimization ranges. Table 3 shows the range of eight values of δ tested and the percentage of data sets for which the optimal value of α_s is less than $\nu(\delta)$.

From these data, we postulate that an optimization range of $[\nu(1) : \nu(5)]$ would produce optimal parameters for most data sets, and effective parameters for the remainder. Thus, values of β_{si} and β_{sf} can be determined during run-time given $\alpha_{si} = \nu(1)$ and $\alpha_{sf} = \nu(5)$.

Table 3: Percentage of data sets observing the optimal value of α_s as the value of δ is increased.

δ	% Data Sets
1	0%
2	20%
3	60%
4	70%
5	90%
6	90%
7	90%
8	100%

The increment value j should be leveraged against the size of the optimization ranges to keep the time of optimization reasonable. For the present purposes, we empirically set j to be 0.1. Further development of more sophisticated methods of determining parameter optimization ranges promises a significant decrease in optimization time by intelligent range determination and adjustment of this parameter.

4.1.2.2 Seed Point Extraction Our segmentation system is initialized by the user selecting a 2D slice from the 3D volume. That 2D slice is within the anatomical object to be segmented, so the manual tracing produced is of that particular object. Figure 27 shows an example 2D slice extracted from one of our RVOT data sets, containing a cross-section of the RVOT.

A manual tracing of the RVOT produced by an expert on the initialization slice can be seen on the left of Figure 28. Extracting a medial seed point for 3D segmentation from this tracing will save the user an interactive step, streamlining our segmentation system. We begin by determining a medial point in the 2D RVOT. This can be accomplished by

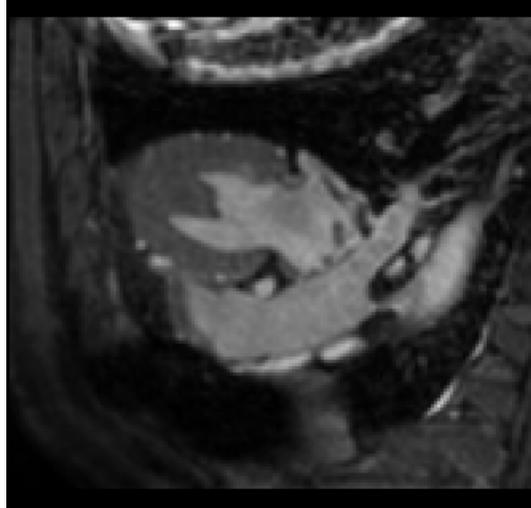


Figure 27: Example 2D slice extracted from a 3D MRI data set. This slice, along with a manual 2D tracing of the RVOT within it, is used to optimize parameters and provide seed points for a 3D segmentation of the RVOT in the full volume.

running our Noiseless Sphere Map Optimization Algorithm on the 2D binary segmentation mask produced by the user. We can determine the medial manifold of the 2D object in the manner described in section 3.3.5, but since we require only a single medial point, we can simply select the largest sphere grown within the object, which will necessarily be centered on the 2D medial manifold. An example of such a sphere can be seen on the right side of Figure 28, highlighted in green.

Because the traced object in the 2D slice is the same object we desire to segment in 3D, the coordinates a medial point in the 2D object, along with the index of the slice from the 3D volume, will provide a seed point within the desired 3D object. Though a medial point in the 2D object will not necessarily fall on the medial manifold of the 3D object, it will be closer to the medial manifold of the target 3D object than it will be to the medial manifold of any other object in the image. Because of this, we can apply a simple radial search outward from this point in the 3D data set to find a medial seed point for 3D segmentation, confident that the first medial point encountered will fall on the target 3D medial manifold.

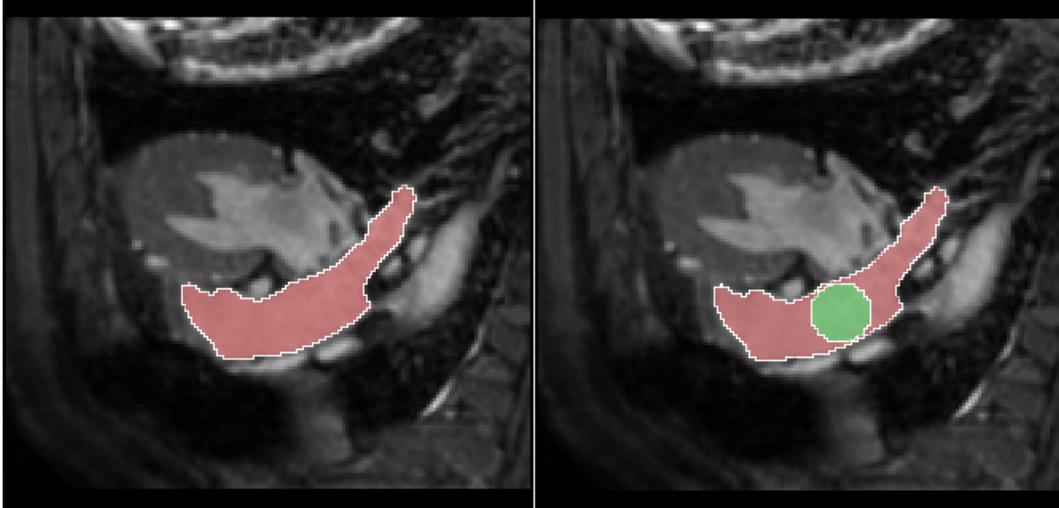


Figure 28: Left: Example 2D slice with manual tracing (red) overlaid. Right: Example 2D slice with manual tracing (red) and medial sphere used to extract seed point (green) overlaid.

4.2 2D SLICE SELECTION

The first supervision step taken by the user in our segmentation process is to select a 2D slice from the 3D image, containing the anatomical object to be segmented. This 2D slice, which we will refer to as the *initialization slice*, will be used to initialize our parameter optimization process. This 2D slice of the target object provides a sample population of pixels and boundary areas from the 3D target object, allowing the calculation of the necessary values described in previous sections of this chapter. In the pilot study presented in Section 4.1, the 2D slice used for initialization was arbitrarily selected. We hypothesize that the choice of initialization slice will have a significant impact on the automated 3D segmentation produced. To test this hypothesis, an experiment was designed using our ovine MRI data, ten sets in total, spanning multiple subjects. Here we present this study, which was designed to facilitate a deeper understanding of the impact of slice selection on parameter optimization, and ultimately on 3D segmentation accuracy.

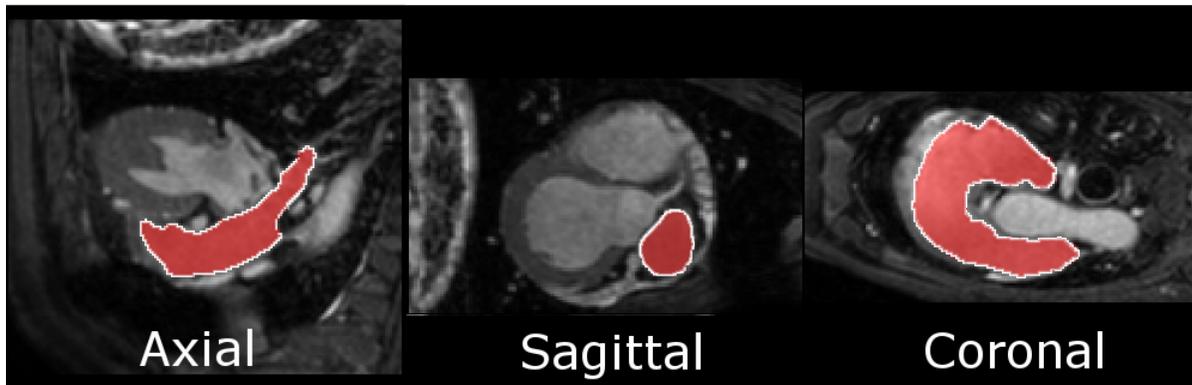


Figure 29: An example of the three slice orientations used for initialization in the study presented. The RVOT (red, outlined) has been manually traced in each slice.

Because individual slices may each contain different aspects of the full 3D object, as well as its neighboring objects, it is likely that there will be corresponding differences in parameters calculated when using different initialization slices. In an effort to quantify these differences, we performed a study to test the impact of the initialization slice selected on the algorithm parameters calculated, and on the resulting 3D segmentation.

For the purpose of this study, rather than selecting a 2D slice directly, our expert user was instructed to select a reproducible point within the target 3D object: the center of the pulmonic artery as seen in the particular sagittal slice in which the left coronary artery leaves the aorta. This point was contained by three unique slices in the standard views: axial, sagittal, and coronal. (Note: the meaning of these standard orientations differs from humans to sheep) Figure 29 shows the three 2D slices produced by our chosen point on a particular RVOT data set. The user was asked to trace the target object, the RVOT, in each of these 2D slices. Each tracing was then used as input for the algorithm parameter optimization scheme described in Section 4.1, customizing three sets of parameters for 3D segmentation of each data set.

The goal of the study was to determine which of the standard views centered around the given point within the target object (axial, sagittal, or coronal) produced the best 3D segmentations when used as our initialization slice and how much variation results from different slices. ITK-SNAP was used for slice selection and manual tracing. [36]

4.2.1 Experimental Setup

We used ten ovine cardiac MRI data sets, with the Right Ventricular Outflow Tract (RVOT) as the target object for segmentation. Three orthogonal slice orientations were chosen for this study, to provide different anatomical views. In each 3D data set the user chose a reproducible point in the RVOT, through which the slices were made (See Fig. 29). The point chosen for slice selection was the center of the pulmonic artery as seen in the particular sagittal slice in which the left coronary artery leaves the aorta. Since the ovine subjects were all scanned in a standard orientation relative to the magnet, the anatomy generally corresponded along the cardinal slices. An expert user traced the RVOT in each of these slices, providing a 2D segmentation denoted as $T(d, s)$, where $d \in \{1 : 10\}$ is the data set and $s \in \{axial, sagittal, coronal\}$ is the slice orientation. Using these tracings as ground truth, we performed parameter optimization on each 2D slice, minimizing the error produced by 2D segmentation using our Shells and Spheres algorithm. This process yielded three sets of potential 3D algorithm parameters for each data set, one for each slice orientation. A given set of parameters optimized for a given slice orientation s from 3D image d is defined as

$$\theta_{d,s} = [\beta_s, \beta_g]. \quad (4.9)$$

Thus, the set of all parameters calculated for the study is

$$\Theta = \{\theta_{d,s} : d \in \{1 : 10\}, s \in \{axial, sagittal, coronal\}\}. \quad (4.10)$$

Automated 3D segmentations using the parameters derived from each of the three slices were performed, denoted as $A(\theta_{d,s}, d)$, where d is both the 3D image from which the optimization slice is taken and the 3D image on which segmentation is performed, and s is the slice orientation used. Three trained users produced 3D manual segmentations of the RVOT,

denoted $M(u, d)$, where $u \in \{1 : 3\}$ is the user, by tracing a stack of slices for each of the ten data sets. We determine the accuracy of the automated segmentations by computing the similarity of the 2D automated segmentations to the corresponding 2D manual tracings

$$DSC(A_s(\theta_{d,s}, d), T(d, s)) \tag{4.11}$$

where the slice through $A(\theta_{d,s}, d)$ corresponding to $T(d, s)$, denoted $A_s(\theta_{d,s}, d)$, is used for comparison, and by computing the similarity of the 3D automated segmentations to the corresponding 3D manual segmentations collected,

$$DSC(A(\theta_{d,s}, d), M(u, d)), \tag{4.12}$$

for each of our three expert users.

4.2.2 Validation

To determine if one particular slice orientation was best suited for initializing the parameters for 3D segmentation, we analyzed the similarity of the automated segmentations to the ground truth manual segmentations for all data sets. We performed a brute force parameter optimization on each data set, using each of the three initialization slice orientations. Figure 30 shows the mean agreement to the 2D initialization tracing achieved, indicating how well the system was able to match the training data. We then performed automated 3D segmentations on each data set using the parameter values calculated through these optimizations, and compared these segmentations to our expert 3D segmentations. Results of this comparison are given in Fig. 31, showing the mean agreement of the automated 3D segmentations produced using each of the three initialization slices.

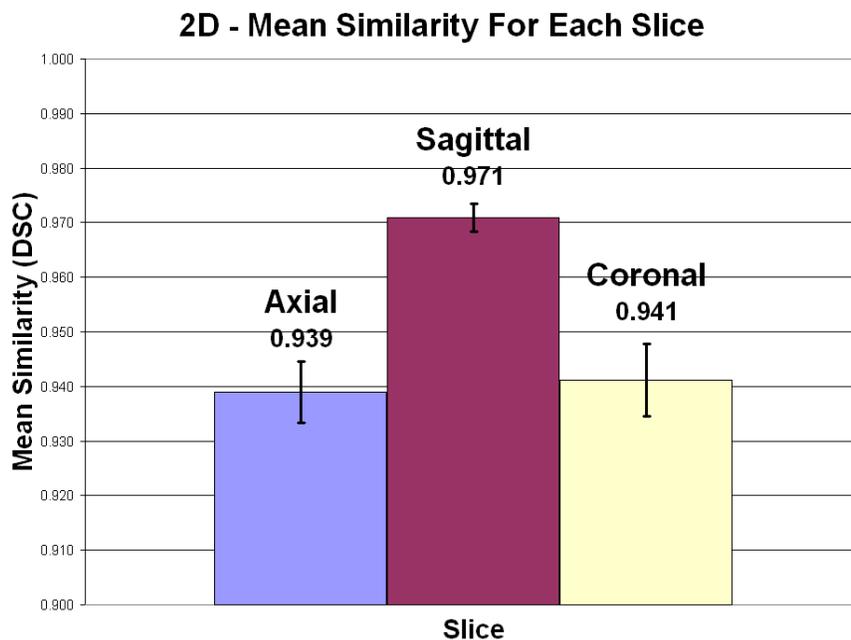


Figure 30: A bar graph showing the mean agreement to the 2D initialization tracing achieved using each of the three slice orientations for parameter optimization, indicating how well the system was able to match the training data. Error bars indicate the standard error of the mean.

Though the sagittal slice orientation produced automated 2D segmentations that most accurately matched the 2D manual tracings, the axial slice orientation produced parameters that yielded the most accurate automated 3D segmentations.

Figure 32 shows the 3D segmentation accuracy, for all data sets, for each slice orientation in boxplot form. Boxes are bounded by the first and third quartiles of the set of DSC values. This visualization gives a better sense of the variation in accuracy over all data sets for each initialization slice orientation. We can clearly see that the axial initialization slice produced the highest median DSC with the smallest inter-quartile range (IQR) surrounding it, indicating a reliable and accurate set of automated segmentations. Further emphasizing the reliability of this axial slice orientation as an initialization slice, the minimum and maximum recorded DSC values fall in a tight range, with the minimum DSC being 0.78.

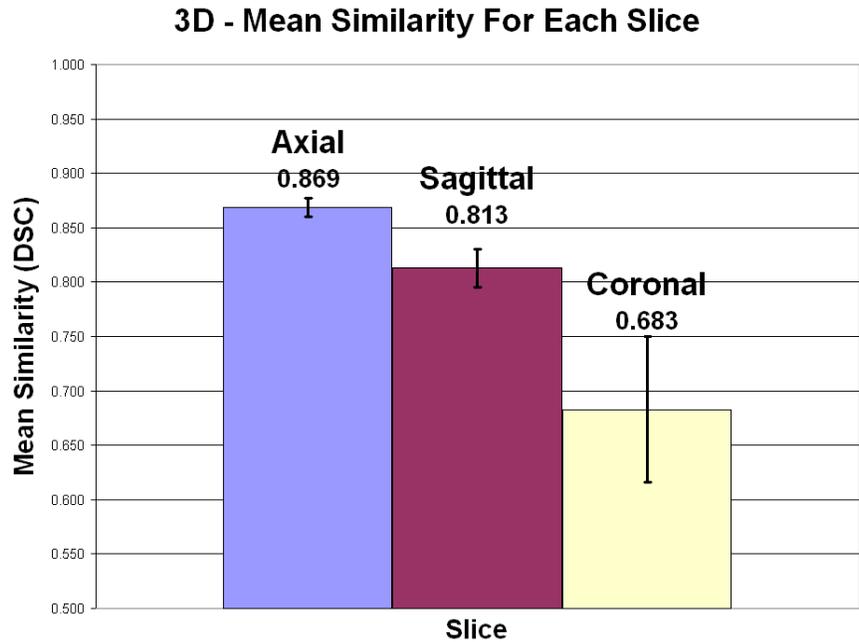


Figure 31: The mean agreement of 3D automated segmentations to our expert 3D segmentations produced using the optimal parameter values calculated by brute force parameter optimization. Error bars indicate the standard error of the mean.)

As a DSC value of 0.70 has been accepted as satisfactory spatial overlap [33, 35, 34], this is an acceptable minimum value. The other two slice orientations each present a much larger IQR, which extends much lower in the range of DSC value, and significantly lower minimum recorded DSC values. These results indicate that the other two slice orientations are notably less reliable for the production of automated 3D segmentations of the RVOT, and, in one case for the coronal slice, provided a segmentation that did not overlap whatsoever with the true anatomy.

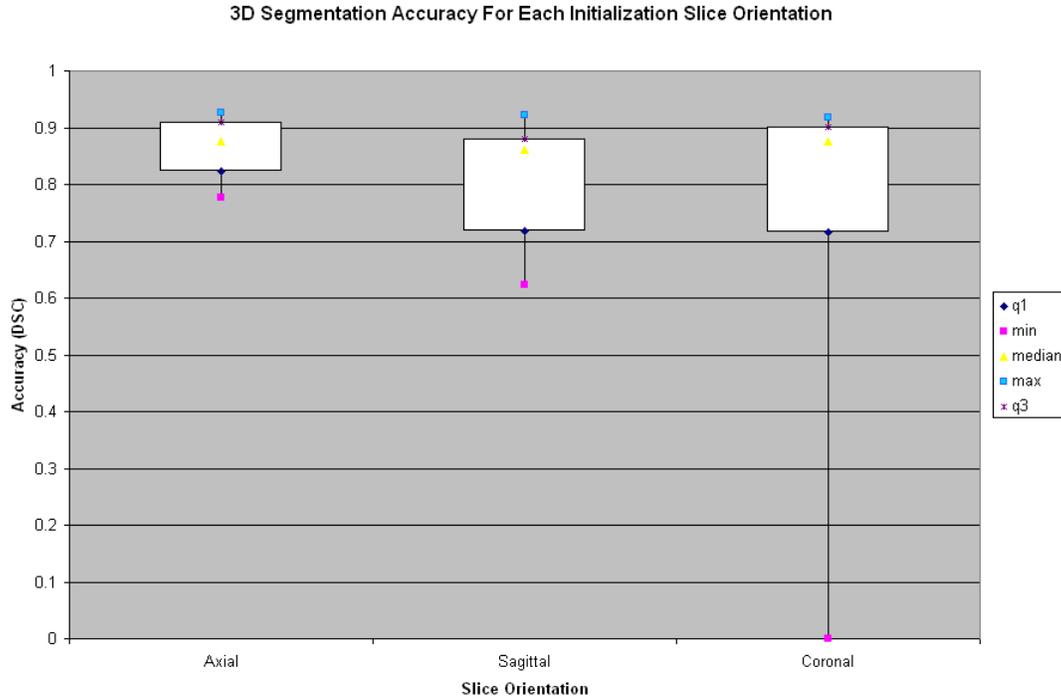


Figure 32: A box plot of the 3D segmentation accuracy yielded by each of the initialization slice orientations over all data sets. Inter-quartile range (IQR) is bounded by the first and third quartiles, and the range from minimum to maximum recorded DSC value is indicated by the vertical lines, with the minimum and maximum values themselves indicated by pink and blue squares, respectively.

We expect that the high 2D agreement to manual tracings of the sagittal slice orientation can be attributed to the trivial nature of tracing the contour of the RVOT in this slice. The area traced is small, has smooth contours with relatively constant curvature, and provides little internal tissue inhomogeneity, as it is a cross-section of the pulmonary artery (PA). The axial slice orientation, however, showed superior performance in 3D segmentation, despite its lesser DSC mean value in 2D, because the RVOT region in the axial slice was most representative of the challenges of the full 3D segmentation. Metaphorically speaking, the axial slice orientation received a lower grade in a harder course, better preparing it for the challenges

to come. The coronal slice, on the other hand, was an exceedingly difficult segmentation challenge. This view of the RVOT generally included an uncharacteristically high degree of tissue inhomogeneity, as well as subjective cardiac walls and structures along the external edge of the target region. These cardiac walls also consisted of intensity values closer to that of the RVOT than other surrounding areas, resulting in variance thresholds not conducive to the majority of RVOT boundaries. The axial slice orientation, however, contained homogeneous intensity regions representative of the 3D RVOT and complex boundaries allowing effective training for the challenges of the 3D RVOT.

Our results yielded strong performance when using the axial slice for initialization on all data sets, producing a mean DSC of 0.867 ± 0.046 when compared to expert manual segmentations. We can conclude that parameters calculated using this slice orientation perform notably better than the other two slice orientations tested, with a higher degree of reliability. We conclude from this study that slice selection has a powerful impact on the 3D segmentation produced, with widely different levels of segmentation accuracy resulting when various slice orientations are used for initialization. A more thorough search of possible 2D slices for initialization would likely yield even more effective algorithm parameters, and such a study could greatly enhance our understanding of the impact of the initialization slice on the parameters calculated, uncovering key features in the 2D initialization slices correlated with particular aspects of 3D segmentation performance.

4.3 ALTERNATIVE PARAMETER CALCULATION TECHNIQUES

Calculating algorithm parameters with our brute-force optimization method proved effective, but is computationally time-consuming when applied to 3D data, hence our 2D optimization approach for 3D data described in the previous chapter. In addition, we have explored alternative methods for determining algorithm parameters that reduce computation times. First, it was hypothesized that our algorithm parameters, which are actually variance thresholds (Eqn. 3.3.2 and Eqn. 3.3.4), could be calculated directly from the sample population of object pixels defined by the 2D initialization tracing. This type of method would require

no optimization at all, removing a large percentage of our total computation time. It was also hypothesized that, given the apparent insensitivity of parameter β_s as compared to β_g , the same value could be used for both parameters without compromising the accuracy of 3D segmentation. The following sections present four alternative methods for parameter calculation, designed to test hypotheses and reduce the computation time of our parameter optimization process.

4.3.1 Alternative Method A

The first method of reducing computation time was designed to retain some level of optimization of algorithm parameters, to keep segmentation accuracy high. To accomplish this, we proposed to optimize one of the two variance thresholds, while calculating the other directly from the initialization data. This reduced our optimization from a 2-dimensional problem to a 1-dimensional problem, exponentially reducing the computation time. Referring to an example graph of our parameter optimization space, reproduced from Section 4.1, in Figure 33, we see that the optimal DSC ridge runs along the axis of the β_s parameter.

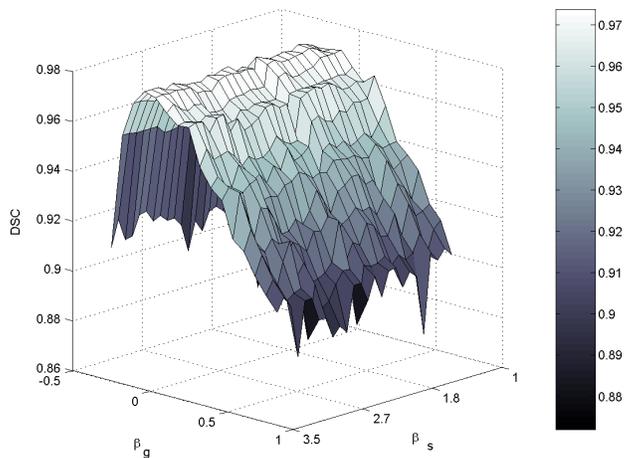


Figure 33: An example of optimization parameter space.

This trend implies that the parameter β_s , and thus, variance threshold α_s , is relatively insensitive, since it can be varied across a wide range of values and still fall on the optimal ridge. This behavior indicates that α_s is an ideal candidate for calculation from the initialization data, while the more sensitive α_g value, corresponding to the sensitive β_g parameter shown in Figure 33, should still be optimized as before to preserve effective segmentation. We will refer to this parameter calculation method as Alternative Method A (AM-A).

The variance threshold α_s is used to shrink spheres that have grown past boundaries by limiting their variance to an acceptable value. This acceptable value is, ideally, the variance value of the sphere at its correct radius. The large range of acceptable β_s values seen in Figure 33 exists because of the swift, extreme increase in the variance measured within a sphere after it has crossed a boundary. This increase is large enough that a sizeable range of variance values exist between the variance of a sphere at its correct scale and the variance of a sphere after it has crossed an object boundary, encompassing all of the variance values along the optimal ridge. Given this knowledge, we can calculate α_s in AM-A from our manual 2D initialization tracing by finding a sample correct 2D sphere and measuring its internal variance. We chose the largest sphere within the a priori tracing, which will likely give the best representation of the pixel population within the object.

4.3.2 Alternative Method B

Further reducing computational load, we consider removing optimization processes altogether, instead calculating both variance thresholds directly from the initialization data. Both of the threshold values to be determined are loosely based on the internal pixel variance of the target image object. A study of the parameters resulting from the original algorithm parameter optimization method presented in Section 4.1 shows that the optimal α_s is much higher than the optimal α_g in all tests performed. This phenomenon is illustrated in Table 4, which shows the optimal values of α_s and α_g resulting from the β_s and β_g values calculated as in Section 4.1 (segmentation results using these values will be presented in the next chapter).

Table 4: Optimal threshold values α_s and α_g for each data set, calculated using full parameter optimization method.

<i>Data Set</i>	α_s	α_g
1	41304.9	8210.8
2	29039.5	10728.7
3	17293.1	4284.1
4	24184.0	5542.5
5	34641.6	6988.4
6	121456.0	11811.8
7	88003.4	7863.3
8	28837.3	8694.1
9	43491.8	8485.5
10	37624.8	8828.3

The likely explanation of this observed trend is that after algorithm Step 2, spheres grown slightly past an object boundary are less detrimental to the remainder of the sphere map optimization process than spheres that are too small. This is because spheres that are too small place reflectors within the target object, potentially blocking numerous other spheres from reaching the object boundary. On the other hand, spheres that are too large place reflectors outside the object boundary. Reflectors placed by these spheres will be “facing” the wrong way to directly impact spheres within the object they reside in, and will be corrected by the larger population of correct outposts.

Taking the information from this analytical study of optimal thresholds, a method of calculating both algorithm thresholds directly from the initialization data was created. The candidate variance values for use from the initialization data are:

- Variance of all pixels within the 2D target object, or Internal Variance (IV)
- Variance of pixels within the largest sphere in the 2D target object, or Largest Sphere Variance (LSV).

Of these values, the IV of a particular initialization slice is experimentally seen to be consistently larger than the LSV. Given the observed trend of optimal α_s being larger than α_g , we assigned the calculated algorithm parameters for Alternative Method B (AM-B) as

$$\alpha_s = \text{IV},$$

$$\alpha_g = \text{LSV}.$$

4.3.3 Alternative Method C

Recall again the wide range of acceptable values for the variance threshold α_s . In devising Alternative Method C (AM-C), we entertained the possibility that the wide range of effective values for this parameter removes the need for independence from the second threshold parameter, α_g . Using the same value for both parameters allows us another means of reducing our problem to a 1-dimensional optimization, saving computation time while maintaining a search for ideal values.

To test this technique, AM-C was designed to make use of a single value for both thresholds, optimized by searching the 1-dimensional space of possible parameter values.

4.3.4 Alternative Method D

A final algorithm parameter calculation method, Alternative Method D (AM-D), retains the theory of using a single value for both variance thresholds, but minimizes computational time by calculating this value directly from the 2D initialization slice and tracing. We again consider the candidate variance values presented in Section 4.3.2. The candidate value closest

to the optimized value of α_g , the parameter with the smallest acceptable range of the two, was the internal pixel variance of the largest sphere found in the 2D initialization tracing, LSV. AM-D assigns this value to both thresholds. Alternative Methods A-D will be tested in Chapter 5.

4.4 DISCUSSION

We have presented our novel approach to algorithm parameter optimization, using a 2D slice to optimize parameters for 3D segmentation. Preliminary tests have illustrated the effectiveness of this technique. Also presented is a means of initializing the parameter optimization system that requires only a single 2D tracing of the target anatomical object in the 2D initialization slice, yielding a system that is intuitive and accessible to clinical users. We have studied the choice of initialization slice, proving its impact on the automated 3D segmentation eventually produced.

In addition to the brute-force parameter optimization method presented, we have presented four alternative methods for calculating algorithm parameters, along with their theoretical justifications, that greatly reduce the computational time necessary to determine segmentation algorithm parameters.

The next chapter will present validation studies of our S&S segmentation system, using both standard and alternative methods of parameter calculation. Automated segmentations will be performed on our database of 3D ovine cardiac MRI data sets, using all variations of our segmentation system, with computation times recorded. Segmentation accuracy will be assessed through comparison to multiple expert segmentations.

5.0 VALIDATION RESULTS

Though the Shells and Spheres framework has been theoretically defined and our segmentation algorithm presented, our previous studies performed limited, primarily qualitative, validation. Through a collaboration with Dr. Michael Sacks and Children’s Hospital, Boston, we obtained ten ovine cardiac 3D MRI data sets. These data sets encompass seven different scans involving four animals, and thus include anatomical variation between subjects. This chapter will present validation studies performed using these data, designed to quantify the performance of our segmentation algorithm. Studies were conducted using the various methods of algorithm parameter calculation presented in the previous chapter.

Segmentation accuracy was determined by comparing automated segmentations to our database of expert manual segmentations of the RVOT, which will be presented in the next section. Subsequent sections will present our novel method of error visualization designed to show localized errors in sphere map optimization, and validation studies comparing our segmentation algorithm to a widely used segmentation technique, level sets, using software designed for clinical use.

5.1 3D MANUAL SEGMENTATIONS

In order to quantitatively assess the accuracy of our segmentation algorithm, a database of gold standard expert manual segmentations was created. Three expert users produced 3D segmentations of the RVOT in our 10 ovine MRI data sets. Segmentations were created with ITK-SNAP by merging slice-by-slice manual traces through the 3D volume.

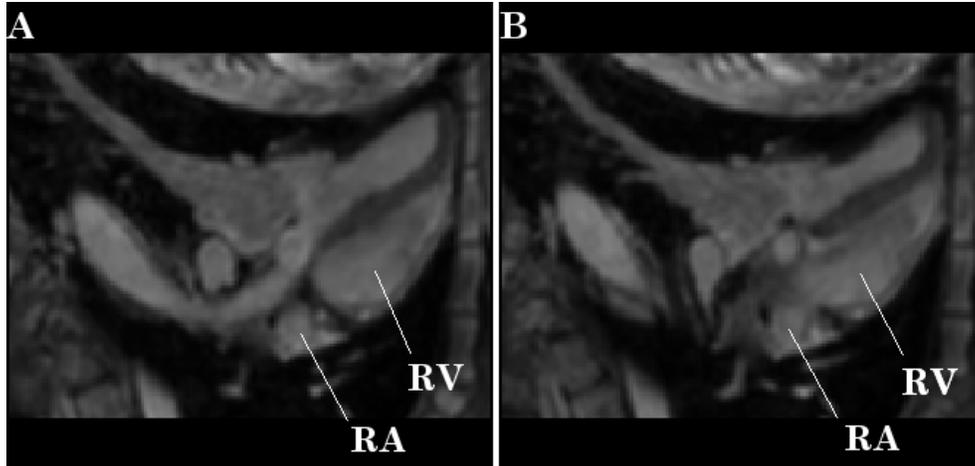


Figure 34: An example axial bounding plane, taken from Data Set 1. A: Bounding plane slice. B: Subsequent slice, moving outside our defined RVOT, where the RV merges with the RA. (See text)

Since the RVOT is an arbitrarily defined cardiac region, bounding planes were established based on anatomical landmarks to encompass the region of interest. These bounding planes were defined as standard views, easily reproducible in any cardiac imaging data set. Note that orientations in ovine data are not the same as those for human data. We are defining our standard views according to metadata in our dicom images, though they differ from the human cardiac standard views. The axial bounding plane was defined as the last axial plane, traveling through the Right Ventricle (RV) away from the RVOT, where the RV is a distinct structure before merging with the Right Atrium (RA). Figure 34A shows an example of the axial bounding plane, taken from Data Set 1. This slice is the last slice considered to be within the RVOT in the axial sequence moving away from the RVOT. Figure 34B shows the next slice in the 3D volume, moving away from the RVOT. The anatomical landmark used to determine the bounding plane can be seen, as the RV begins to merge with the RA.

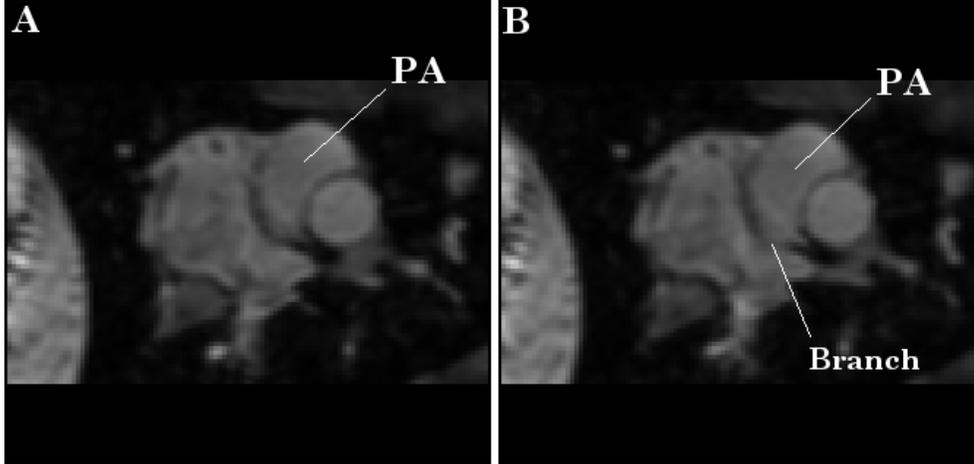


Figure 35: An example sagittal bounding plane, taken from Data Set 1. A: Bounding plane slice. B: Subsequent slice, moving outside our defined RVOT, where the PA branches.

The sagittal bounding plane was defined as the last sagittal slice, moving downstream in terms of blood flow through the Pulmonary Artery (PA), in which the PA is a single, separate shape before the branch point just beneath the Aortic Arch (AA) . The point at which the PA branches can be seen in Figure 35B. This slice is the next slice downstream from our bounding plane shown in Figure 35A, where the PA is still an independent structure.

A surface rendering of one of our bounded segmentations of the RVOT can be seen in Figure 36, with the bounding planes labeled. Since these bounding planes allow our segmentation to extend from the RV to the branch point of the PV, our segmentations necessarily capture the loosely-defined region of the RVOT for subsequent modeling and analysis. Manual segmentation of each 3D RVOT required 1-2 hours of time from each user. Similar manual segmentation times have been reported by other investigators for segmentation of cardiovascular structures. [37, 38]

The two bounding planes described above are sufficient to define the extent of the RVOT. Before comparing automated segmentations to this gold standard database, it was first necessary to determine the inter-rater reliability, or level of observed agreement, of the expert users who provided the manual 3D segmentations. We model inter-rater reliability

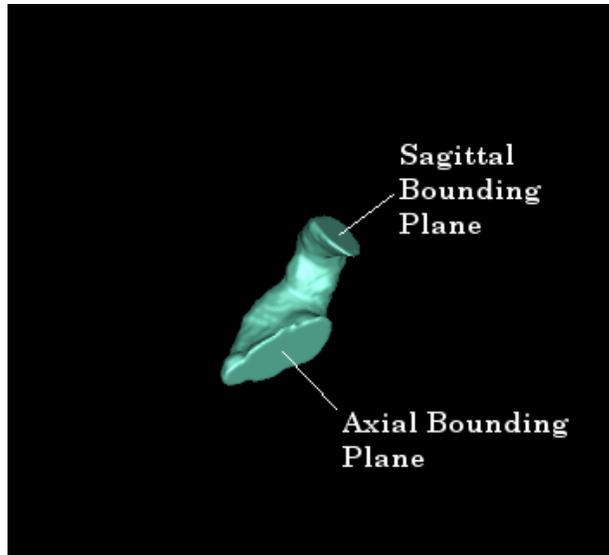


Figure 36: A surface rendering of a bounded RVOT segmentation, with the axial and sagittal bounding planes labeled.

in our study as the mean DSC of the $\binom{n}{2}$ combinations of manual segmentations for each data set. In the case of this study $n = 3$, because we have three experts producing manual segmentations. Figure 37 shows the inter-rater reliability for each of our MRI data sets.

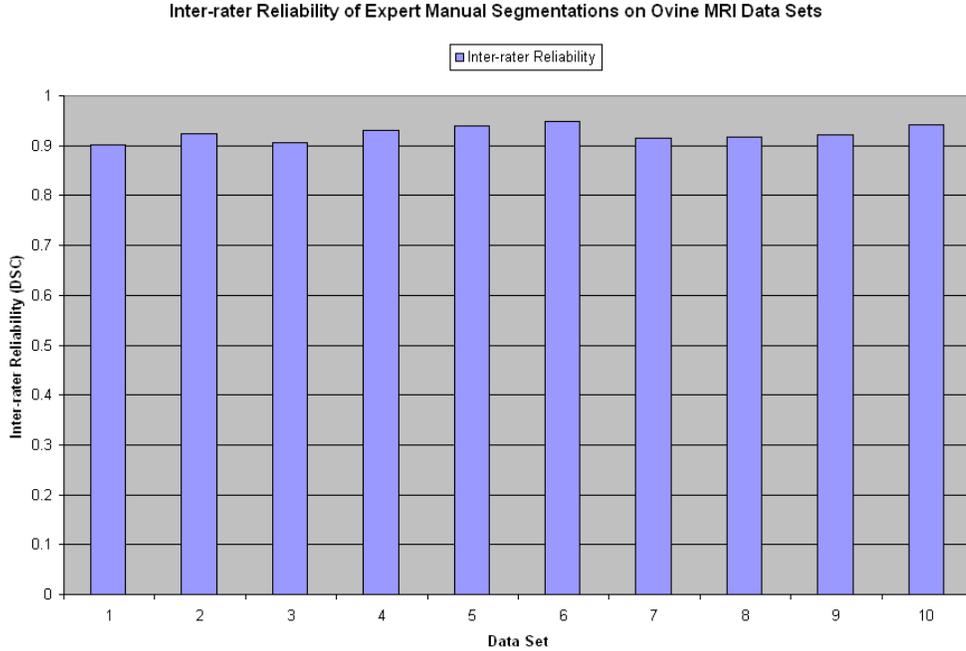


Figure 37: Bar graph of the Inter-rater reliability for our expert manual segmentations, calculated as the mean DSC between manual segmentations for each data set.

The mean inter-rater reliability across all data sets was 0.925, indicating an average agreement of 0.925 between our experts when segmenting the RVOT in the set of images presented in this study. In the rest of this chapter, we will show agreement between various automated segmentations of the RVOT in our data. For quantitative comparison to our expert segmentations, we will introduce the automated segmentation into the set of expert manual segmentations for a particular data set, and recalculate inter-rater reliability as the mean DSC of the $\binom{4}{2}$ combinations of segmentations including this new segmentation. The error, or amount by which this reliability measure increases or decreases with the addition of the automated segmentation, is a measure of the performance of the automated system as a segmentation expert. This technique helps to clarify the meaning of the automated accuracy results by giving a measure of the behavior of our automated segmentation system in the context of other expert segmentations.

5.1.1 Standard 2D Parameter Optimization Technique

The first study performed was a 3D segmentation study using the brute force parameter optimization methods presented in Chapter 4. This study was done in conjunction with the initialization slice selection study presented in Section 4.2, and the 3D automated segmentation accuracy presented for each data set was achieved using the 2D initialization slice exhibiting the best 3D performance of those tested.

2D slices were selected from each of our 3D data sets as described in 4.1. Optimization of the β_s and β_g parameters of our segmentation algorithm was performed on each 2D slice, as described in Chapter 4, searching for optimal parameter settings throughout the parameter space as shown in Figure 38. The calculated optimal parameters were then applied to their respective 3D data sets, with results shown in Figure 39. Shown is the mean DSC value for each automated segmentation, representing the average agreement of a given automated segmentation to the three expert manual segmentations for the same data set. Over all data sets, compared to all expert segmentations, our S&S segmentation algorithm produced a mean segmentation accuracy of 0.87 ± 0.0498 .

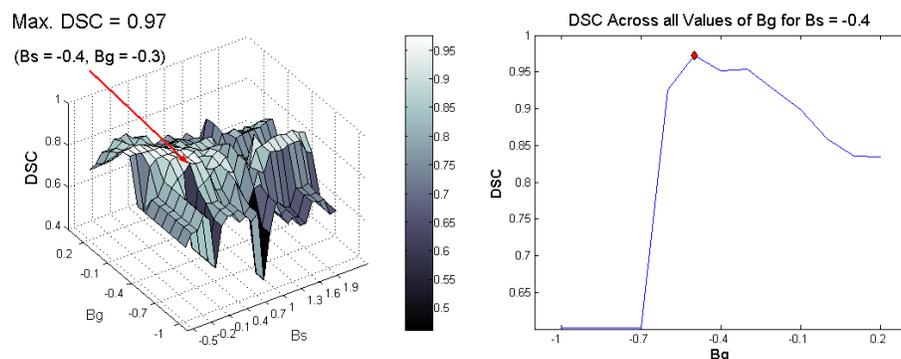


Figure 38: Example DSC curve produced by a 2D parameter optimization The optimal DSC value 0.96 found during this optimization procedure is indicated with an arrow.

However, these values are less meaningful than they may appear, because the expert manual segmentations exhibit disagreement between themselves that is not accounted for. As described in the previous section, a set consisting of the S&S automated segmentation

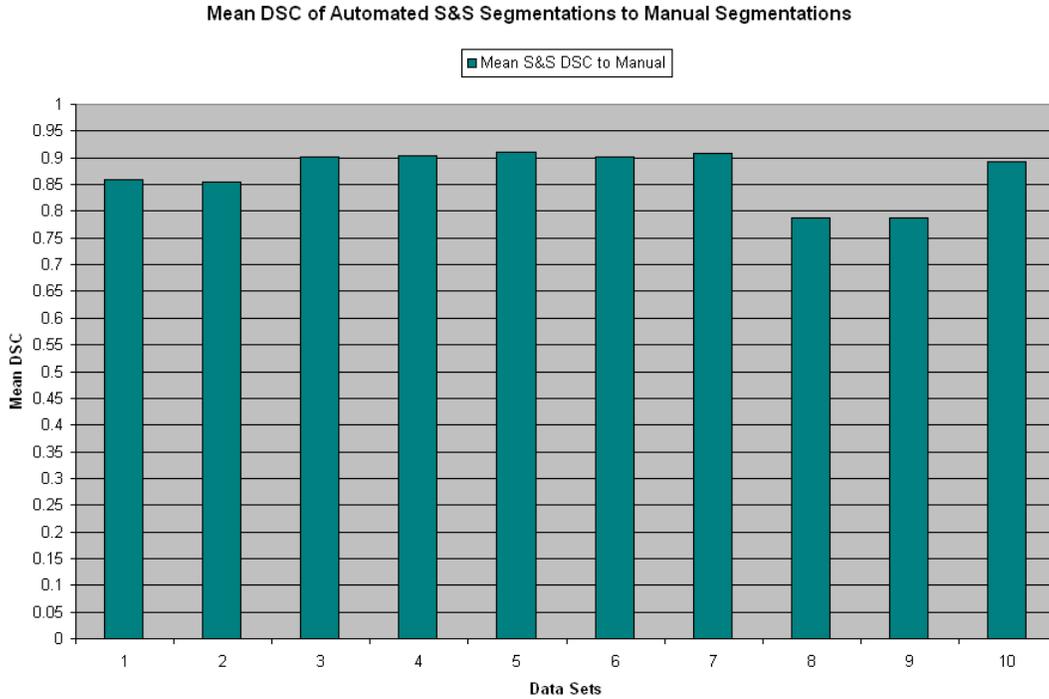


Figure 39: Bar graph of mean DSC of S&S automated 3D segmentations created using brute force 2D parameter optimization with the axial initialization slice, when compared to our three expert manual segmentations, for each of our MRI data sets.

and the three expert manual segmentations can be created, and the inter-rater reliability of this set can be calculated to provide more insight into the accuracy of the automated system. This new inter-rater reliability, as well as the original inter-rater reliability of the set of expert manual segmentations alone, can be seen in Figure 40. These data allows us to compare the inter-rater reliability of our set of segmentations both with and without the addition of our automated segmentation, showing the degree to which the automated segmentation appears to be as accurate as an additional expert.

For added clarity, Figure 41 shows the change in inter-rater reliability produced by the addition of the automated 3D segmentation for each data set. It can be seen that in most cases the error was small, falling below 0.075 DSC for all data sets, although qualitative

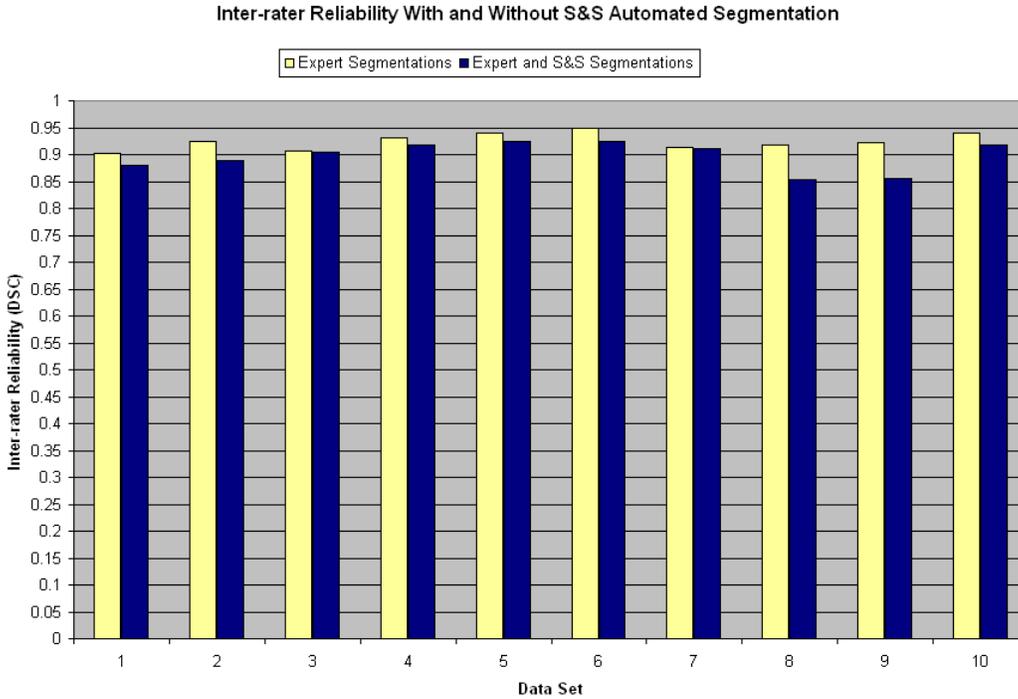


Figure 40: Bar graph showing the inter-rater reliability of sets containing only the expert manual segmentations (yellow) and the expert manual segmentations along with the automated S&S segmentation (blue) for comparison.

judgement based on this measure must be made with caution. For example, it will decrease as the number of observers increases. The two data sets with the highest error, data sets 8 and 9, were both from the same MRI study, a study that produced particularly poor automated segmentation accuracy due to lower quality images. The mean error over all ten data sets was 0.027 DSC. Though not flawless, the automated segmentations produced were effective, and in some cases (data sets 3 and 7) nearly indistinguishable from expert manual segmentations, given that their addition to the set of segmentations produced a nearly identical inter-rater reliability.

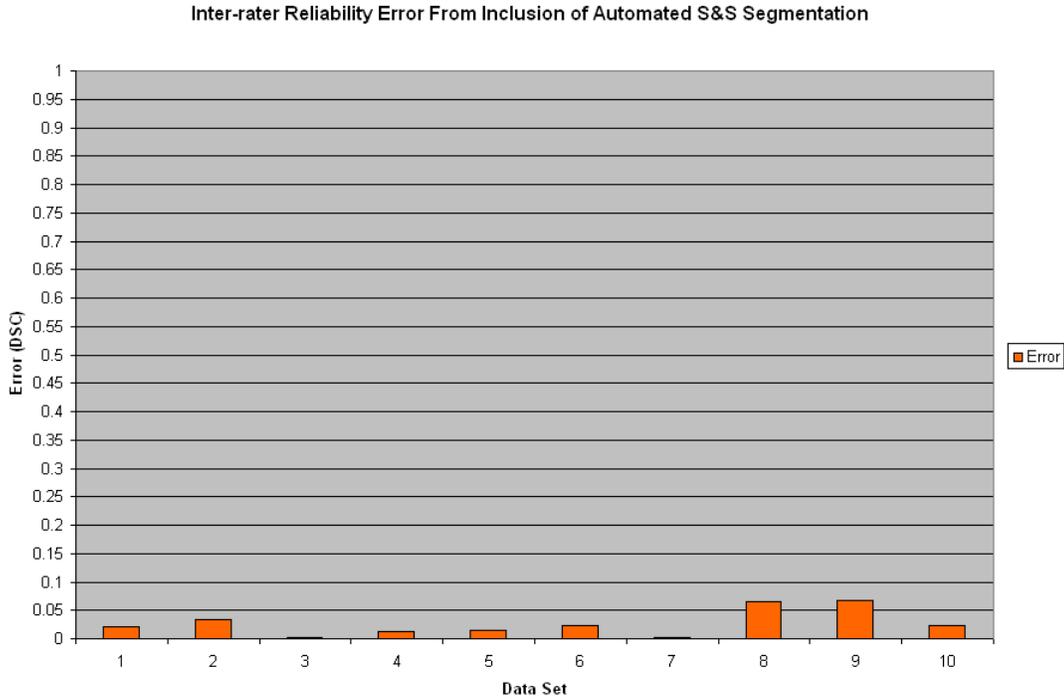


Figure 41: Bar graph showing the difference between the inter-rater reliability measurements of the set of expert manual segmentations with and without the addition of the automated S&S segmentation.

We now turn our attention to the computational time of automated 3D segmentations. Manual segmentations of 3D objects in medical images is a time-consuming process, and it is expected that an automated segmentation system must produce 3D segmentations in comparable or faster times to be appealing in the clinical setting. The fundamental measure of time in a clinical imaging setting is the amount of time from image acquisition to diagnosis, which is aided by segmentations produced. The time to produce a segmentation, with either automated or manual methods, may directly delay a clinician’s ability to make a diagnosis. Thus, while computers are a more numerous resource than experts in medical segmentation, allowing many automated segmentations to be performed simultaneously, we focus our discussion on the total time to produce a single automated 3D segmentation.

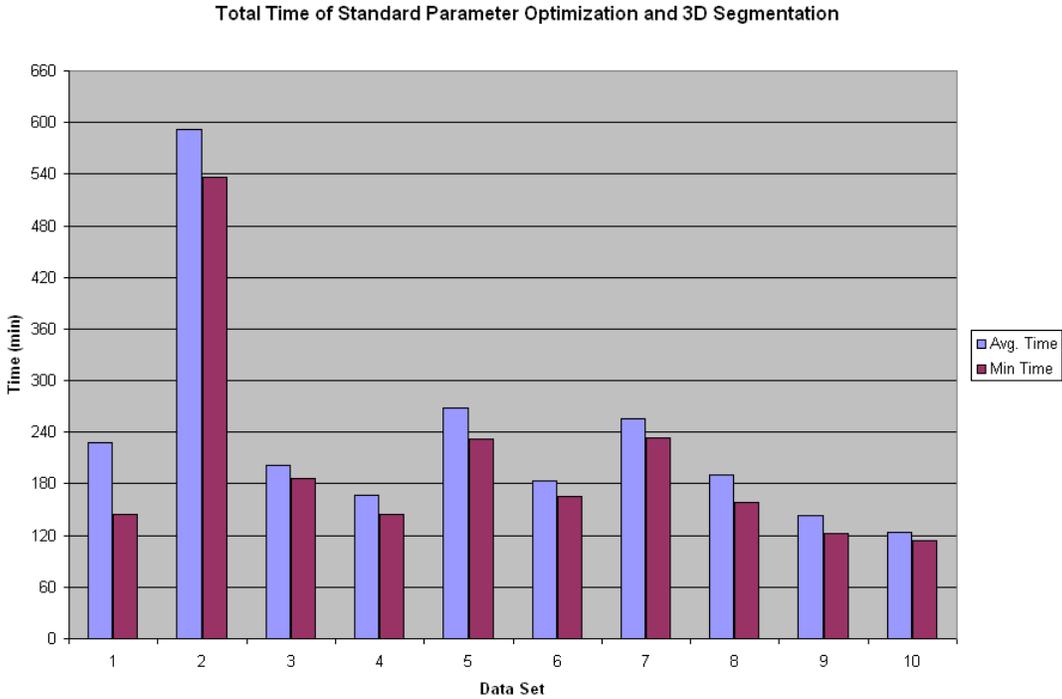


Figure 42: Bar graph of the average (blue) and minimum (maroon) times of 3D segmentation, including parameter optimization for each data set. Average and minimum values are in the context of the three initialization slices tested.

We present computational time data for our automated S&S system in producing 3D segmentations of the RVOT in Figure 42. Shown is the mean time, averaged over the three 2D initialization slice tested, and minimum time of the initialization slices for each data set. Interestingly, it has been observed that the most accurate segmentation of the three initialization slices generally required the least computational time. This correlation exists because the most effective algorithm parameters limit sphere growth to the correct radius, while less optimal algorithm parameters allow spheres to grow larger than they should, taking excess computational time to iterate through the extra shells.

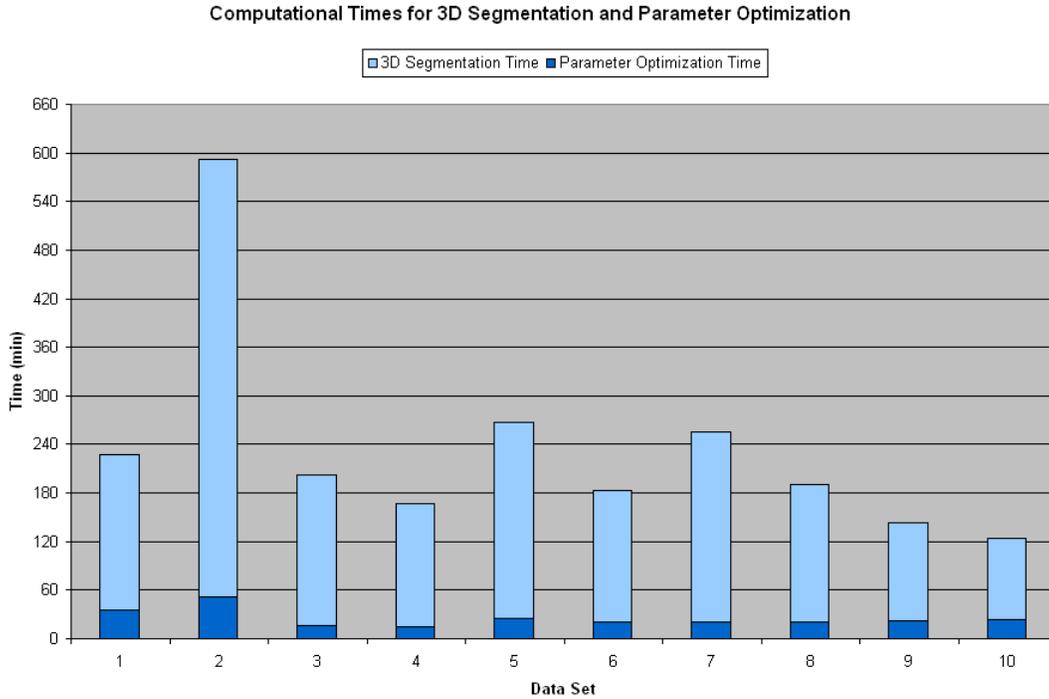


Figure 43: Bar graph of the average time, split into parameter optimization (dark) and algorithm run time (light) for each data set.

Figure 43 presents the average computation time for each data set, separated to show the time required for algorithm parameter optimization (using the brute-force technique) and the additional time required to optimize a sphere map for the image using the automated S&S system. The time spent in parameter optimization is shown by the darker region of each time bar, while the time to produce a 3D segmentation using those parameters is represented by the lighter region of each bar. The data shows that parameter optimization time is a small fraction of the total time to produce a 3D segmentation, taking less than 30 minutes in most cases. It can also be seen that, aside from the atypically large computation time for data set 2, all of our data sets required between two and four hours of computation. This time is roughly twice as long as the time observed for a human to produce a manual 3D segmentation of similar anatomical objects, both in our studies and in others. [37, 38]

However, the amount of human interaction time required by our S&S system is under 10 minutes. It should also be noted that a single analysis with our automated system produces a sphere map for the entire image, including all anatomical objects within. Given a properly optimized sphere map, the segmentation flood-fill at the end of our algorithm, detailed in Section 3.3.6, rarely requires more than 1-2 minutes. This time is negligible compared to the time required for sphere map optimization, which consumes the great majority of the overall computational work for 3D segmentation. Thus, a single S&S analysis, given proper algorithm parameters, can be said to produce segmentations for all objects within an image in a similar time frame to that required to produce one manual 3D segmentation.

5.1.2 Alternative Parameter Calculation Techniques

Quantitative analysis was also performed on 3D segmentations produced using the alternative methods of parameter calculation presented in Section 4.3. As in that section, the alternative methods will be referred to here as AM-A, AM-B, AM-C, and AM-D. Each alternative method was used to produce a 3D segmentation on all MRI data sets, with the same three initialization slices used in the validation study in the previous section. Analysis times were also recorded, for comparison to the computational time necessary for the brute-force parameter optimization method.

Alternative Method A (AM-A) utilized separate variance thresholds, with α_s calculated from the 2D initialization tracing and α_g determined via optimization through the 1-dimensional search space of potential variance values. Segmentation accuracy using AM-A with our S&S algorithm to produce 3D segmentations of our ovine data sets using each of the three candidate 2D initialization slices can be seen in Figure 44. When compared to the accuracy of 3D segmentations produced with the 2D brute-force parameter optimization technique presented in the previous section, it can be seen that AM-A produced less accurate results, with much more variation in performance between data sets. We again found heavy dependence on the slice orientation used for initialization, but found the sagittal slice to be preferable when using AM-A. If we consider a DSC of < 0.7 to be a failure, the sagittal slice yielded only one failed segmentation, while the axial and coronal slices yielded three

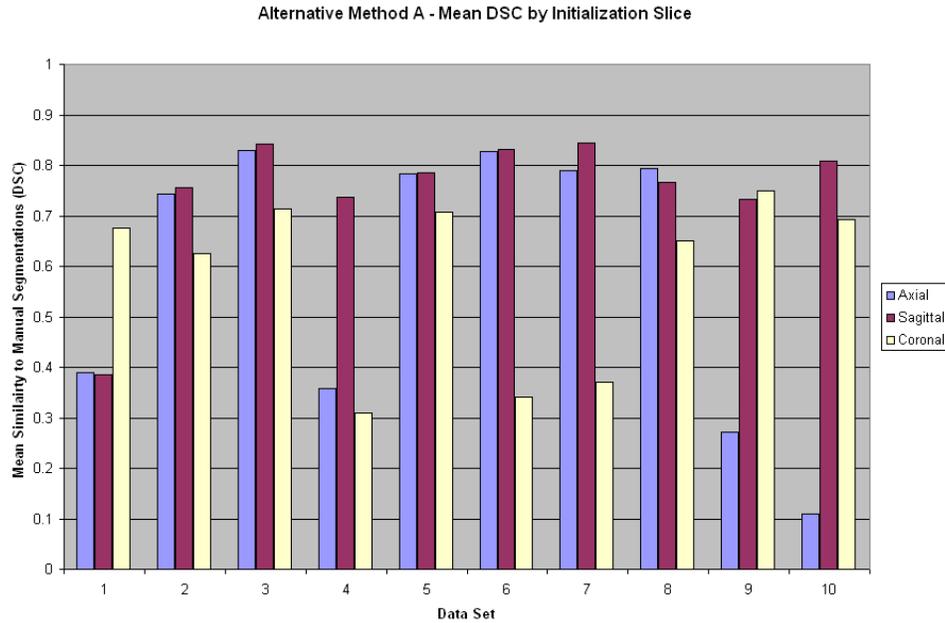


Figure 44: Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-A, using each of the three 2D initialization slices.

and eight failures, respectively. While our brute-force optimization method, using the axial slice for initialization, produced DSC values above 0.78 consistently, AM-A was unable to produce consistent segmentation accuracy, regardless of initialization slice. The best performance of AM-A, offered by using a sagittal initialization slice, produced an average DSC of 0.75 ± 0.13 on all data sets, with a minimum DSC of 0.38 on data set 1.

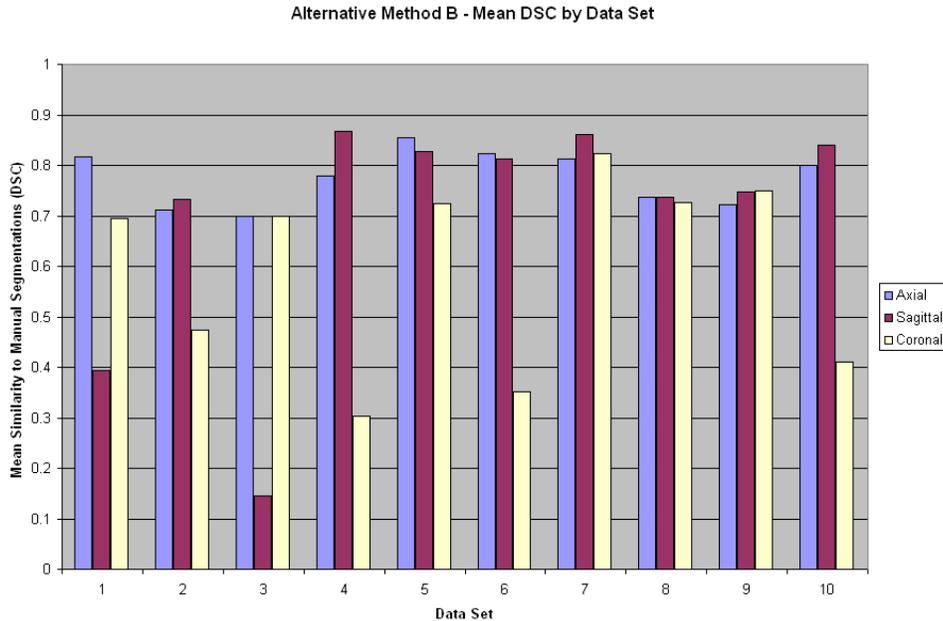


Figure 45: Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-B, using each of the three 2D initialization slices.

Figure 45 shows 3D segmentation performance using for AM-B in the same format. The results from AM-B are more promising, with the axial initialization slice consistently producing a DSC of at least 0.7 on all data sets. Other initialization slices produced inconsistent results, however, making overall performance of this parameter calculation method questionable. The average DSC recorded using AM-B with an axial initialization slice was 0.78 ± 0.06 , over all data sets. In fact, AM-B yielded the most accurate, reliable results of all the alternative parameter calculation methods tested, though still falling short of the segmentation accuracy produced by brute-force parameter optimization.

Figure 46 reports 3D segmentation performance when using AM-C. This parameter calculation method showed the highest level of reliability when using the sagittal initialization slice, producing a DSC ≥ 0.70 for all but one data set. The overall mean DSC produced using AM-C with a sagittal initialization slice was 0.76 ± 0.14 .

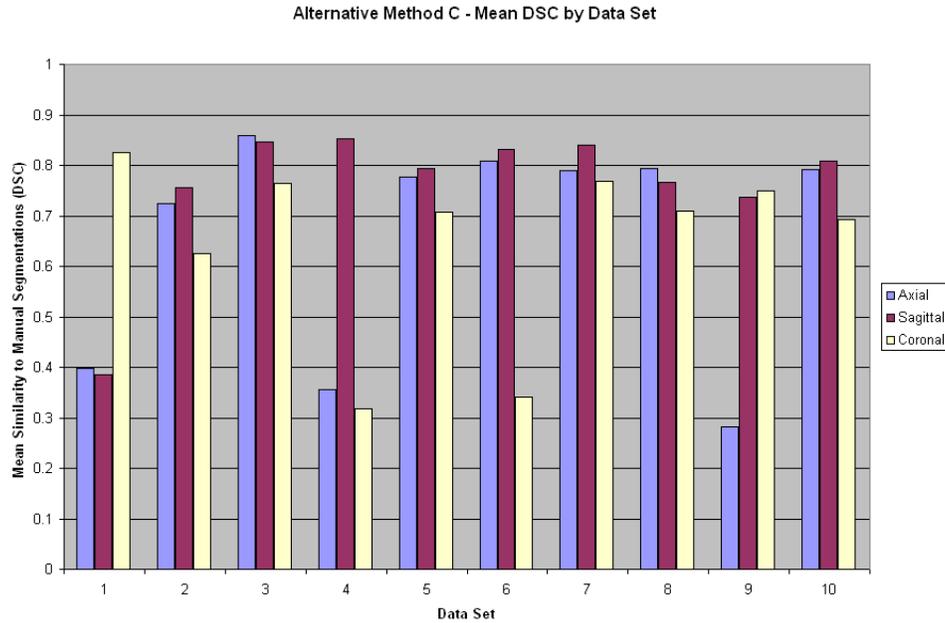


Figure 46: Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-C, using each of the three 2D initialization slices.

AM-D yielded segmentation accuracy nearly identical to AM-B, as can be seen in Figure 47. As with AM-B, using the axial initialization slice produced the most consistent and accurate performance over all data sets, with DSC values to expert segmentations ranging from 0.69 to 0.86. The overall mean DSC recorded using AM-D with an axial initialization slice was 0.77 ± 0.06 , slightly below that of AM-B.

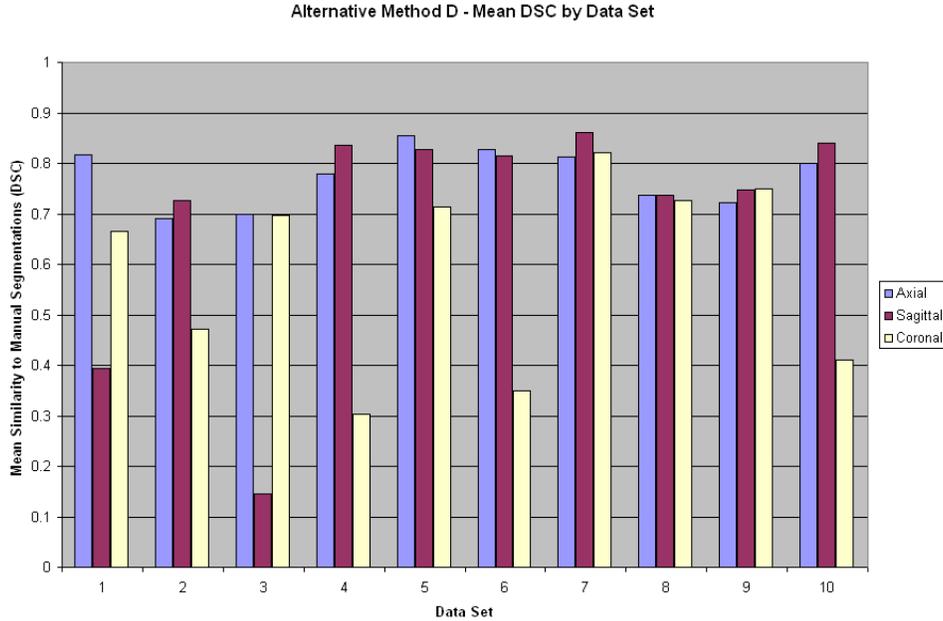


Figure 47: Bar graph of mean 3D DSC for all MRI data sets using parameter calculation method AM-D, using each of the three 2D initialization slices.

A summary comparison of all parameter calculation methods, including our standard brute-force technique, is shown in Figure 48. The box plot shows the area, measured in units of DSC, between the first and third quartiles. The median DSC value is marked with a triangle. The minimum and maximum 3D DSC values calculated are also shown by lines extending vertically above and below the boxes. Values below the standard outlier range of $1.5 * IQR$, where IQR is the Inter-Quartile Range, have *not* been removed from the graph in order to give a more exact representation of all data collected.

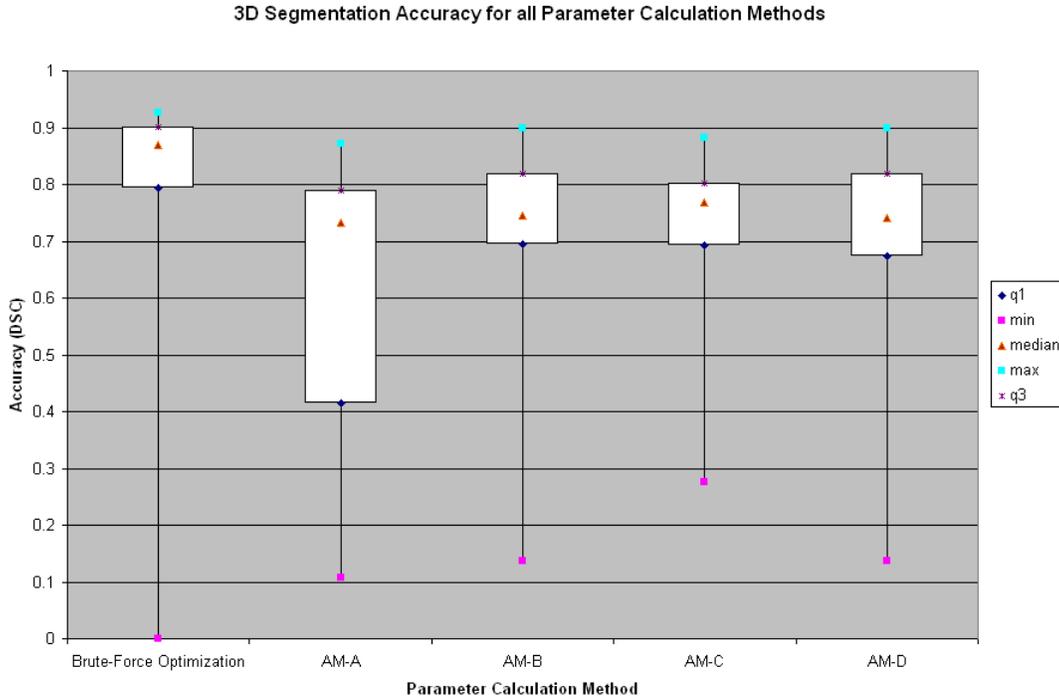


Figure 48: Box plot of the 3D segmentation accuracy, measured by DSC, for each of the parameter calculation methods. Data includes automated 3D segmentations of all ten MRI data sets, using each of the three initialization slices, compared to each of the three expert manual segmentations. Q1 and Q3 represent the first and third quartiles of the DSC data, respectively.

We observe that our brute-force parameter optimization method not only has the highest IQR, in terms of DSC, but also the smallest IQR of all parameter calculation techniques. This is convincing evidence that the original brute-force algorithm parameter optimization technique produces more accurate and reliable segmentations than the other methods tested. Furthermore, as shown in Figure ??, judicious selection of the slice used for parameter optimization eliminates the failures that produced the brute-force optimization outliers seen in Figure 48.

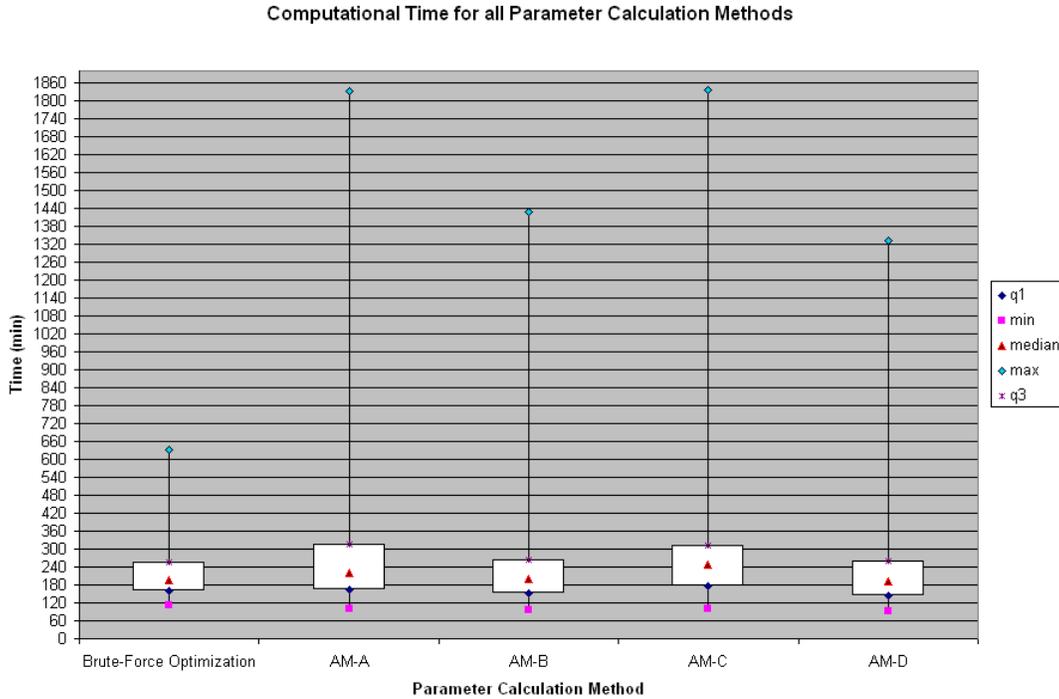


Figure 49: Box plot of the computation time of 3D segmentation, measured in minutes, for each of the parameter calculation methods. Data includes automated 3D segmentations of all ten MRI data sets, using each of the three initialization slices, compared to each of the three expert manual segmentations. Q1 and Q3 represent the first and third quartiles of the time data, respectively.

We next consider the anticipated increase in computational time when using the brute-force parameter optimization technique, as the alternate parameter calculation methods were designed to reduce the computational time of parameter determination. Figure 49 shows a box plot of time data collected for each of the parameter calculation techniques. Surprisingly, the brute-force optimization technique had comparable, or better, computational time performance relative to the alternative parameter calculation methods. The brute-force parameter optimization method shows a much lower maximum time (630 minutes, as opposed to 1333-1836 minutes for the other methods), while maintaining the second lowest median

time, the smallest IQR, and the lowest third quartile marker value. This behavior is likely to be because the alternative parameter calculation methods produce sub-optimal algorithm parameters, leading to excess calculation as spheres are grown larger than necessary.

We have shown that the brute-force parameter optimization method has time performance comparable to or better than any of the other parameter calculation methods devised, with less variability in computational time. More importantly, 3D segmentations produced using the brute-force method significantly surpassed those produced using other methods in terms of accuracy and reliability. We can conclude that brute-force algorithm parameter optimization remains the most effective technique for use with our S&S segmentation algorithm. We believe this result is because, given proper selection of a 2D initialization slice, brute-force optimization is the most reliable method of producing effective parameters.

5.2 ERROR VISUALIZATION

Validation data has been presented thus far in the form of the Dice Similarity Coefficient (DSC), a measure of overall agreement between segmentations, comparing our automated segmentations to expert manual segmentations. While this measure, as with most measures of segmentation accuracy, provides information on how accurate the segmentation was overall, it gives no insight into the local causes of any inaccuracy in the segmentation. Thus, a segmentation algorithm that produces a segmentation exactly one pixel beyond the boundary of an object on all sides and a segmentation algorithm that fails utterly on one corner of an object but is completely accurate elsewhere may produce the same DSC. Obviously, cases such as these would require different steps to be taken to correct the problem, due to their differing natures of failure. When assessing the accuracy of a segmentation algorithm, information about local error is at least as important as global information. In the two hypothetical case presented, knowledge of local error would tell us that the second algorithm failed only in a specific part of the object, and thus the failure can be attributed to the algorithm's response to some local feature of the object. In the case of the first algorithm, local error will be seen to be even throughout the object, and can be attributed to some general behavior of the algorithm along the entire object boundary.

To facilitate effective analysis of error in our Shells and Spheres segmentation algorithm, we require a method of assessing localized errors in the segmentation, preferably tied to an intuitive method of visualizing that error. To keep our error analysis as closely linked as possible to our algorithm itself, allowing knowledge of the nature of errors to be intuitively applied to algorithmic solutions, we looked to the unique nature of the sphere map itself. The most useful information in determining the cause of failure in a Shells and Spheres algorithm is which spheres are incorrect, and to what extent they are too large or small. Visualizing hundreds, or thousands, in some cases, of incorrectly grown spheres simultaneously would yield an output too complex and confusing for visual analysis. Conversely, visualizing only a few faulty spheres at a time would not give a global sense of the segmentation and its weak/strong points.

An effective compromise in visualization was reached, utilizing a display of only the central pixels of spheres, with each pixel’s color corresponding to its sphere’s error from its ideal radius. Given a gold standard manual segmentation, the Noiseless Sphere Map Optimization algorithm can be run on the binary mask, growing spheres centered at each pixel within the segment until it reaches the manually-defined boundary to produce a Gold-Standard Sphere Map (GSM). An automated S&S algorithm, applied to the same data set, produces an Automated Sphere Map (ASM). Since a sphere map can be represented simply as an image with a radius value at each pixel, these two images can be subtracted to obtain an Error Sphere Map (ESM), as

$$ESM = ASM - GSM. \tag{5.1}$$

The resulting ESM may have both positive and negative radius values, corresponding to the difference in radius (in the positive or negative direction), between each sphere in the ASM and its corresponding sphere in the GSM. For instance, a sphere grown two pixels too large in automated analysis would produce a value of 2 at its central pixel in the ESM, while a sphere that fell one pixel short of reaching the boundary in automated analysis would produce a corresponding value of -1 in the ESM. Of course, a correct sphere would have a corresponding value of 0 in the ESM.

Visualization of the ESM was performed in ITK-SNAP, which allowed us to not only show the semi-transparent pixels overlaid on the real image data, but also a 3D surface rendering of the ESM. We display this surface rendering intuitively as a heat map, with colors along a range of yellow to red showing increasing sphere growth past object boundaries, and colors in a range from light blue to purple indicating early sphere stoppage before object boundaries. Figure 50 shows an ESM for one of our RVOT data sets visualized in ITK-SNAP. The bottom-left quadrant displays the surface rendering, while the other three screens show axial, sagittal, and coronal views of the data set, with semi-transparent error colors overlaid.

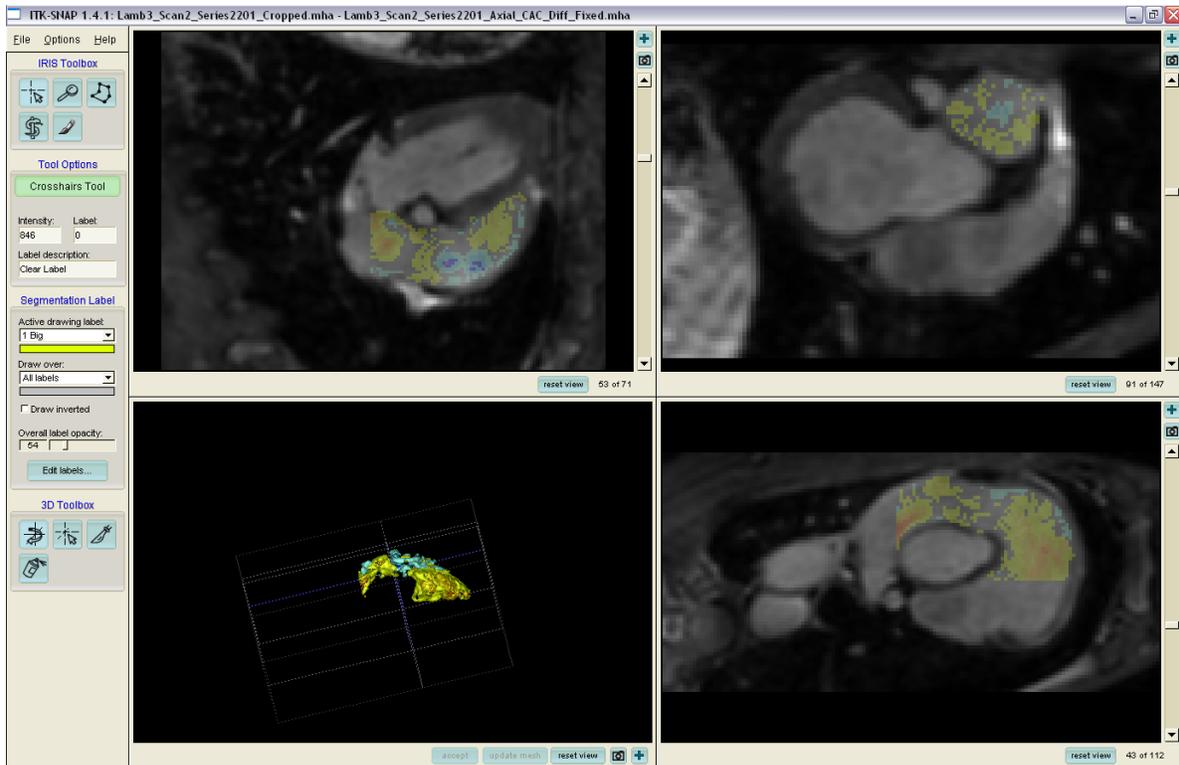


Figure 50: Screenshot of ITK-SNAP displaying an Error Sphere Map (ESM) overlaid on the original image data. The object segmented was an RVOT, bounded at pre-determined cutoff planes. the lower left quadrant displays a surface rendering of the ESM.

While this visualization gives the user a sense of the proportion of spheres with incorrect radii relative to the gold standard through the proportion of color appearing within the target object, so much data displayed concurrently in a visual environment may be overwhelming. In Figure 50 we can clearly see the RVOT, bounded by our pre-determined arbitrary bounding planes, outlined in the surface rendering window. Since spheres agreeing completely with the gold standard would produce no surface, this rendering gives the sense that a large proportion of spheres in our automatically generated sphere map were wrong. However, in this form the visualization fails to give a sense of the overall error. Since our segmentation is ultimately achieved by a flood-fill through neighboring medial spheres, segmentation accuracy will only be directly affected by spheres at the center of the object. Furthermore, the amount

of segmentation error is not simply proportional to the magnitude of the difference in radius of the automated sphere from the gold standard sphere. The segmentation error, measured in the number of pixels incorrectly included or excluded from the segmentation, will likely increase rapidly, depending on the shape of the object boundary, as a medial sphere's radius gets farther away from its ideal value. For example, a medial sphere growing one radius too large will cross an object boundary on either side by one pixel, causing a small handful of pixels (the exact number of dependent on the radius of the sphere) to be included in the segmentation that should not be. Were this sphere to grow another radius past the object boundary, the number of pixels outside the object boundary included in the sphere will more than double, assuming the local curvature of the object boundary is not extreme.

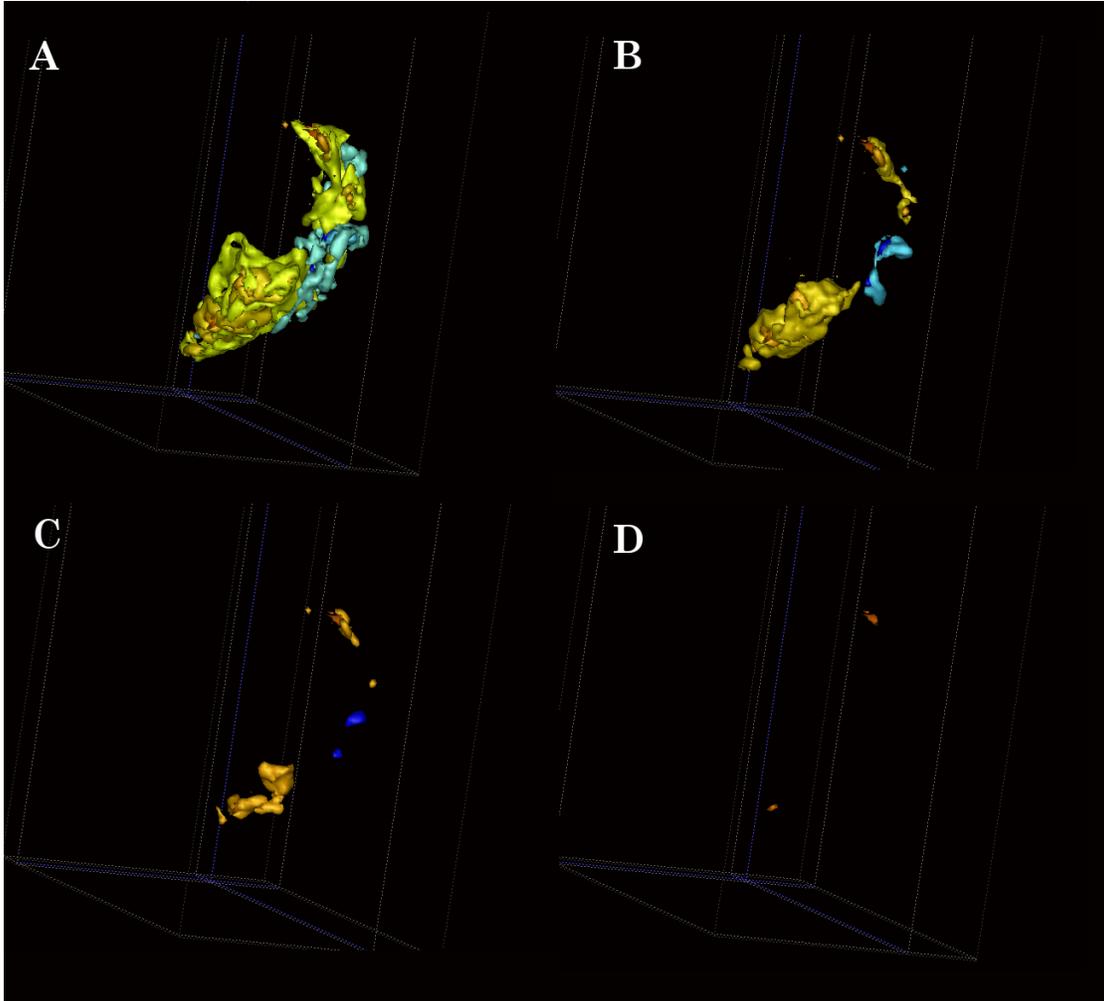


Figure 51: **A:** Surface rendering of the ESM of an RVOT. **B:** The same RVOT ESM, with spheres within 1 radius of the gold standard removed. **C:** The same RVOT ESM, with spheres within 2 radii of the gold standard removed. **D:** The same RVOT ESM, with spheres within 3 radii of the gold standard removed.

Figure 51 shows the surface rendering from Figure 50 in four states, successively removing spheres based on their magnitude of error. While Figure 51A shows all spheres in the ESM surface map, Figure 51B shows the surface after removing all spheres within one radius of the gold standard value from the visualization. This drastically reduces the colored area in the surface rendering, proving that a majority of incorrect spheres were ± 1 from the correct radius value, thusly generating little error in the segmentation produced. Figure 51C and Figure 51D go on to remove spheres with radii off by ± 2 and ± 3 , respectively, from the gold standard, giving a sense of the distribution of magnitude of error among all incorrect spheres. We see that a sphere with a magnitude of error greater than 3 was an exceedingly rare occurrence in this segmentation, indicating that the surface distance of our automated segmentation will rarely, if ever, exceed three pixels. We also see that the majority of high magnitude error was concentrated in the end of the RVOT leading to the RV. This information, which could be critical to creating corrective measures, could not have been detected using traditional segmentation error metric, such as those previously presented.

5.3 VALIDATION AGAINST OTHER TECHNIQUES

We have validated the use of our S&S segmentation system on cardiac MRI images by calculating agreement to a set of expert manual segmentations. To further argue for our system as contributing to the field of medical image segmentation, we now compare the S&S segmentation system to another commonly used computational segmentation technique, using a software tool created specifically for clinical use. ITK-SNAP was developed by Yushkevich et al. [36] and published in 2005 to be an inexpensive, clinically accessible software package offering cutting-edge semiautomated segmentation techniques in a polished, intuitive interface for clinical users, a vision which we share. One of the semiautomated segmentation algorithms implemented in ITK-SNAP is the *geodesic active contour* method designed by Castelles et al. [39, 40] This algorithm defines an evolving snake, which is guided by a pre-processed feature image of edges. The snake evolution in ITK-SNAP is implemented using a *level set method* [41, 11, 22], which is a recent standard for curve and surface evolution. Details on the theory and implementation of this active contour level set method can be found in [36].

5.3.1 Supervision

We begin our comparison with a discussion of the extent and nature of the supervision required by each algorithm. As previously presented, our S&S segmentation system required the user to perform two initialization steps at the onset of analysis:

- Select a particular slice from the 3D image containing the target object.
- Perform a 2D tracing of the target object on the chosen slice.

All algorithm parameters are then automatically calculated and optimized by the system, using the 2D tracing provided. No further user supervision/interaction is required, during or after the segmentation process. The requisite operations of target identification and 2D tracing require minimal user time and effort, and utilize only the anatomical knowledge inherent to a clinical professional.

Now we examine the active contour technique, which requires a very different means of supervision. A feature image is calculated from the input image in a preprocessing step. This feature image $g(I)$, where I is the input image, is a specialized gradient magnitude image calculated using three manually determined parameters:

- σ , the scale of gaussian blurring applied to the input image I before the gradient magnitude image is calculated
- κ , an edge contrast factor
- λ , an edge mapping exponent

Once the feature image $g(I)$ is calculated the user must position “bubbles”, or spherical seed regions, within the object to be segmented and specify their size. The user can place and adjust the size of as many bubbles as desired, though in our study we limited the number of bubbles to one, to ensure that we embedded no additional 3D spatial information into the active contour than is made available to the S&S algorithm. Once the bubble is placed and properly sized, three algorithm parameters must be manually adjusted to control the evolution of the contour by defining the front propagation equation. These parameters are:

- α , the balloon force term
- β , the curvature force term
- γ , the advection force term

An alternate experimental equation including three additional parameters is available in the software, but was not used in our study. Finally, the user must initiate evolution of the contour and supervise its growth, deciding how many evolutionary iterations the snake should perform before the segmentation is complete. Including the number of iterations for snake evolution, and the radius of the seed bubble, this active contour method requires the user to set eight parameters throughout its operation, a process which must be continually monitored from start to finish. While the ITK-SNAP software has been designed to provide visual feedback for the parameters governing the feature image $g(I)$, the user still must be knowledgeable of the ways in which various aspects of $g(I)$ may effect the evolution of the contour. The user must also have knowledge of the nature of level set contour evolution to

accurately predict the effect of parameters α , β , and γ . Such knowledge is outside the realm of expertise of the clinical professional, drawing on high-level mathematics, optimization, image analysis, and machine learning.

5.3.2 Experimental Methods

For comparison to our S&S segmentation system, the implementation of geodesic active contours in ITK-SNAP was used to segment the RVOT in each of the ovine MRI data sets from our previous validation study. Parameters were determined via manual experimental testing, using the preview functionality of ITK-SNAP. Figure 52 shows an example feature image $g(I)$ from one of our MRI data sets, its parameters manually adjusted, with the RVOT labeled.

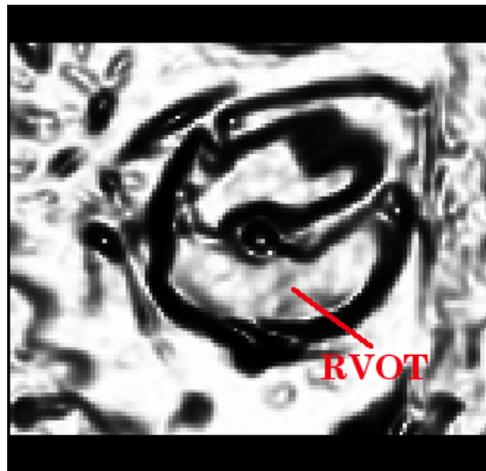


Figure 52: An example feature image $g(I)$ created via manual parameter adjustment using ITK-SNAP. The RVOT is labeled. (red)

The single seed bubble was manually placed in the center of the RVOT near the pulmonary valve, as determined by manual inspection. Bubble placement can be seen in Figure 53, where a bubble shown in green has been placed in the RVOT of a grayscale slice from the same data set used to create the feature image in Figure 52.

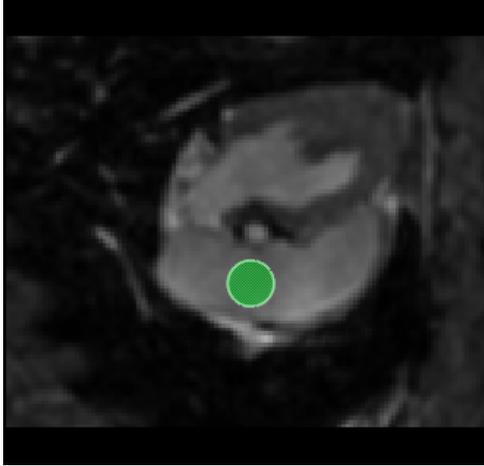


Figure 53: A slice of the grayscale MRI image used to create the feature image in Figure 52, with the seed bubble (green) for the active contour placed in the RVOT.

The user supervised the evolution of the contour, manually overseeing the number of iterations used to produce the final 3D segmentation. The next section will present the results of performing this active contour segmentation on each of our ovine MRI data sets, and a comparison to the segmentations produced by our Shells and Spheres algorithm.

5.3.3 Results

Segmentations of the RVOT in our 3D ovine MRI data sets were produced using the geodesic active contour method implemented in ITK-SNAP. As before, the DSC is used to show agreement to the manual 3D segmentations of the RVOT produced by our three experts. Results of this comparison can be seen in Figure 54, alongside the agreement data previously presented for our S&S segmentation method.

This graph shows the mean segmentation agreement to all expert segmentations over all ten ovine MRI data sets for the active contour method and our S&S method. The error bars were determined by the Standard Error of the Mean (SEM). Figure 55 shows the mean agreement value for each individual data set for both segmentation methods.

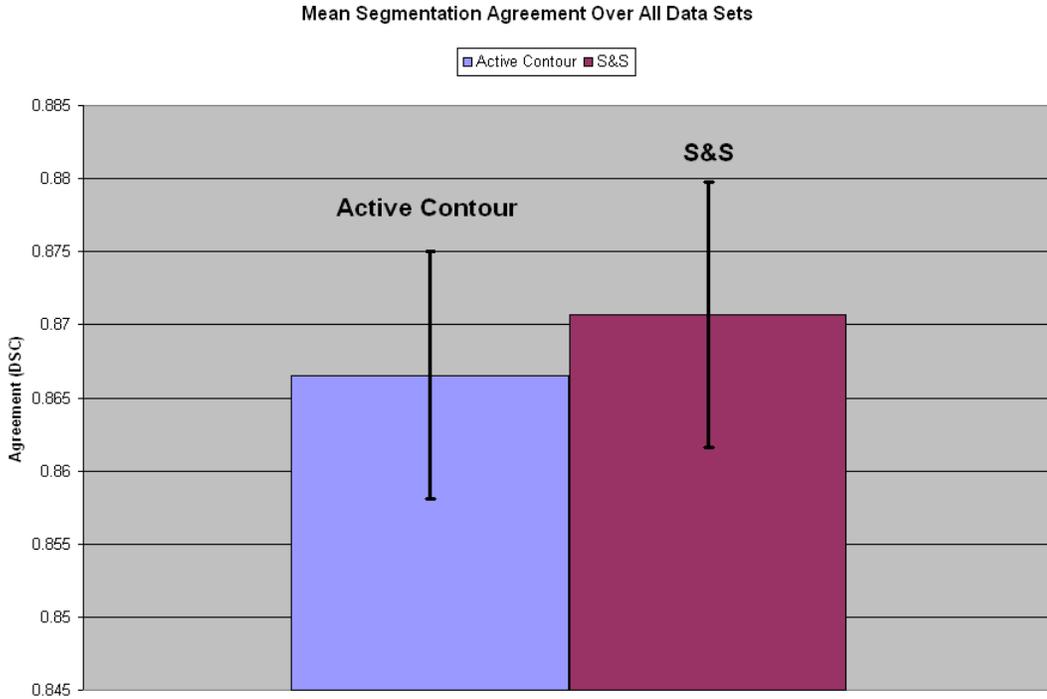


Figure 54: Bar graph showing the mean 3D segmentation agreement to all expert segmentations over all data sets for both the active contour method (blue) and our S&S method (maroon). Error bars show the standard error of the mean (SEM).

We can see that the S&S segmentation system performed slightly better than the active contour method, but the high degree of overlap of the respective SEM values indicates roughly equivalent performance. An independent-samples t-test showed that the two means were not significantly different (p-value = 0.741). We conclude that S&S can match this current clinical state of the art automated segmentation method, with significantly less supervision, as described. Furthermore, the minimal supervision required by the S&S algorithm is at the onset of analysis, unlike the constant supervision necessary for the active contour method. Additionally, our system is designed to require only skills and expertise inherent to the clinical professional, rather than expecting a medical professional to gain algorithmic or mathematical expertise to effectively perform automated 3D segmentations.

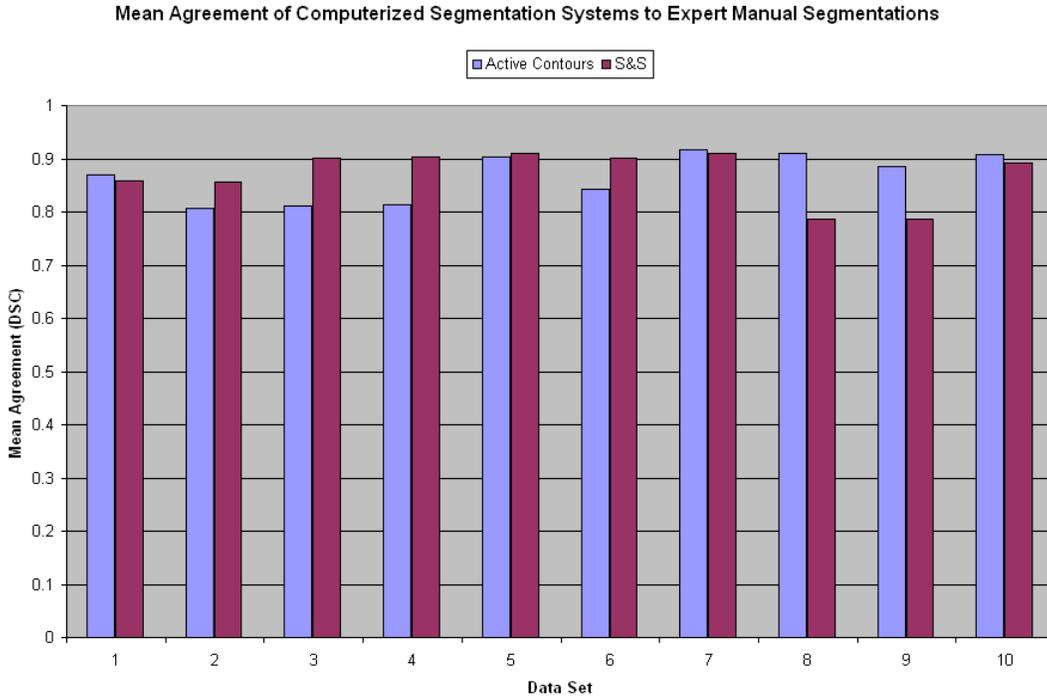


Figure 55: Bar graph showing the individual 3D segmentation agreement to expert segmentations for each data set for both the active contour method (blue) and our S&S method (maroon).

Intensity thresholding coupled with a flood-fill from a manually placed seed point was also explored as a common technique for comparison to our segmentation system, but the prevalence of partial-volume effects and tissue inhomogeneity in MRI images [9] made this method incapable of segmenting the RVOT in our data sets, due to “bleeding” of the flood-fill regardless of threshold parameters. Without a high degree of manual post-processing, this method produced a failed segmentation ($DSC < 0.70$ for all expert segmentations) on each of our MRI data sets. A discussion of the accuracy and reliability of this method is thus irrelevant, as it is merely a reflection of the skill of the user performing the necessary post-processing of the segmentation.

6.0 CONCLUSIONS AND FUTURE WORK

Our research has produced a validated, clinically accessible automated segmentation system requiring minimal supervision. We have developed unique methods of training a 3D system with sparse 2D data, automatically optimizing algorithm parameters. Our system produces n -dimensional segmentations complete with linked medial and boundary point sets. Our approach has been validated on a database of ten 3D ovine cardiac MRI data sets, through comparison to manual segmentations produced by three expert users. We have also developed a method of displaying local error unique to S&S. We have reported insights acquired through our experience in designing a clinically accessible automated image analysis system, our validation studies, and through the use of our segmentation system to produce segmentations for mechanical modeling of the RVOT. We present our conclusions in the following sections. The final section of this chapter will propose future work to be done on this research, outlining the author's theories on improvements to the segmentation system presented, new studies to be performed, potential novel algorithm designs using the Shells and Spheres framework, and expected fruitful directions for future research in clinical image segmentation.

6.1 CLINICALLY ACCESSIBLE SYSTEMS

The importance of specifically designing automated medical image analysis systems to be accessible to the clinical user cannot be overstated. There are numerous issues to consider when attempting to bring new technology into clinical practice, regardless of its potential to improve patient care, including cost of purchase and implementation, system reliability, acceptance by the clinical community, and necessary training time. Researchers have con-

cluded that initialization of systems is a pivotal research area, and have noted the difficult tradeoff between manual interaction and performance. [28, 9] We addressed these concerns by designing our Shells and Spheres segmentation system specifically for the clinical user, addressing each of the above challenges to introducing new technology into clinical practice.

The costs of implementing a new system include training personnel in its use, and such costs will be driven down by introducing systems that are intuitive and easily used by clinical professionals. A system that is comfortable to its users will produce more reliable results, by minimizing user errors and maximizing the likelihood that users will utilize the system effectively and appropriately. A system that makes use of an interface intuitive to the clinician and requires no input other than anatomical tracing will find easier acceptance among clinical professionals than those requiring more complex user interaction. It will also minimize training time for new users. Research shows that limiting supervision to the onset of analysis is preferable in automated segmentation systems, minimizing time and effort put forth by users and leaving them free to pursue other tasks unhindered while the system proceeds with the automated segmentation. [28]

Our S&S segmentation system follows the precepts set forth by Yushkevich et al., who cite a “lack of inexpensive user-friendly tools implementing semi-automatic [image segmentation] methods”. [36] We have adopted the interface of ITK-SNAP, which is both intuitive and robust, offering many of the user-interactive features required by our system in a well-polished, open source package. This interface enabled us to simplify the supervised portion of our segmentation procedure, allowing our users to easily select a 2D slice from a 3D data set and trace a simple 2D contour around the target anatomical object. All necessary data for tuning our 3D segmentation algorithm is derived from this action, with no technical expertise required from the user beyond recognition of the target anatomy.

Our segmentation framework and algorithm have been implemented using the open-source Insight Toolkit (ITK) [42], ensuring that our segmentation system will remain accessible and inexpensive.

6.2 VALIDATION OF THE SHELLS AND SPHERES FRAMEWORK FOR IMAGE SEGMENTATION

We have presented validation studies of our Shells and Spheres framework for image analysis, and the associated n -dimensional segmentation algorithm. Inter-rater reliability was calculated as the mean combinatorial Dice Similarity Coefficient (DSC) between the three expert manual segmentations for each 3D data set. These inter-rater reliability values were compared to the mean DSC of an automated S&S 3D segmentation to the three expert segmentations, and the difference between mean automated DSC and inter-rater reliability was reported. In 80% of our data sets the difference was below 0.1 DSC, and stayed below 0.14 DSC in the remaining 20%. The mean difference over all data sets was 0.05 DSC, indicating that our S&S segmentation system is nearly equivalent to manual segmentation.

Studies were conducted to determine the impact of the 2D slice used for initialization of our segmentation system, using a brute-force parameter optimization method. Of the three slice orientations tested, one orientation was notably superior to the other two, producing an average 3D segmentation accuracy 5% higher than other orientations. This orientation also yielded much lower variance in accuracy between data sets, proving it to also be the most reliable of those tested.

Additional methods of calculating algorithm parameters from given initialization information were explored, to test the necessity of the dual-parameter brute-force optimization technique. These were designed to relieve some of the computational burden of the brute-force optimization approach, by directly calculating one or both of the parameters from a priori data in a fraction of the time. Results indicated that the brute-force optimization method not only out-performed alternative methods in terms of accuracy, yielding a median accuracy at least 10% higher than any other method tested, but showed no increase in overall computational time over alternative methods. The total 3D segmentation time, including parameter calculation, when using the brute-force optimization system was comparable to the lowest time of the alternative methods, with significantly lower time variance between data sets. This surprising result emphasizes the effectiveness of the algorithm parameters calculated by the brute-force method, as the increased efficiency of our algorithm using these

parameters offset the added time of computation necessary for brute-force optimization. From these results, we can conclude that further work improving computational efficiency of the segmentation algorithm would have a greater impact on the total time required than speeding up the initial parameter optimization.

Our studies have validated the effectiveness of the Shells and Spheres framework and our particular segmentation algorithm for producing automated 3D segmentations with minimal user interaction and a training set of only a single 2D image slice. We have demonstrated the superiority of brute-force algorithm parameter optimization over other parameter calculation methods. We have also shown the large impact of the 2D slice chosen for initialization and training of our segmentation algorithm on 3D segmentation accuracy and reliability, and have identified an initialization slice that produces reliable results on all volumes in our 3D cardiac data library.

6.3 USING SEGMENTATIONS OF THE RIGHT VENTRICULAR OUTFLOW TRACT FOR MECHANICAL MODELING

We have delivered 3D segmentations produced by our automated system to collaborators to be used for mechanical modeling of the RVOT. In addition to automated segmentations of the RVOT from each of the ovine MRI data sets, we have also delivered the expert manual segmentations collected as part of our validation process. Segmentations were post-processed to create deliverables in the form of point clouds, containing linked pairs of boundary and medial points for each RVOT.

6.4 FUTURE WORK

Development of the Shells and Spheres framework and associated image analysis algorithms is an ongoing effort, with many avenues of further research to be pursued. This section will discuss the next steps in research envisioned by the author, proposing improvements to the existing algorithm, approaches for new algorithm design, and further studies to be conducted on the framework.

6.4.1 Algorithm Improvements

Our current image segmentation algorithm, presented in this dissertation, has a number of distinct limitations that should be addressed in future releases. In addition, the S&S framework itself may be improved by the addition of new statistical measures and features that may be useful for segmentation, or other automated image analysis tasks. In this section we will cover both levels of enhancement to our work. Our discussion will be loosely structured to follow the segmentation algorithm through its sequential steps, bringing to light shortcomings, enhancements, and directions for further research as appropriate.

Of the six steps taken by our 3D segmentation algorithm, the first step (VSS Gradient-Based Radius Approximation, Section 3.3.1) uses the greatest percentage of computational time. In many cases, it has been observed that this step requires as much computation time as the other five steps combined. This first step merely provides a rough estimate of the correct radius of our spherical operators, initializing the sphere map to a state in which an acceptable percentage of spheres contain a statistically significant sample of pixels from within their respective image objects. As such, the current iterative process of testing statistics at each step of sphere growth for all pixels is excessive, as only a percentage of spheres must be initialized for the algorithm to proceed effectively. Additionally, initialized spheres need not necessarily be set to the exactly correct radius. The complete impact of these initialized spheres on algorithm performance is complex, but as spheres will be further adjusted by subsequent steps, it is likely that a reasonable approximation of a percentage of correct sphere sizes is all that is necessary. Spheres whose radii were considered reliable

could be used to set neighboring spheres, if a reasonable confidence estimate could be devised. We envision a system that selects a percentage of “seed” pixels to initialize, applies with a confidence test to weed out uncertain spheres, and subsequently assigns initial values to the remaining pixels via a specialized interpolation method. The seed pixels may be selected using a static mask or by a monte carlo method. If a monte carlo method is used, it may be advantageous to weight the selection to include a greater percentage of pixels that are statistically likely to be within the target anatomical object, using the 2D tracing provided by the user as a statistical gold standard. Pixels within the target object are hypothesized to have a more dramatic impact on the eventual segmentation, and should be given heightened consideration. Furthermore, potential other novel methods of initializing the radius of our spherical operators may be envisioned, with the hope of providing a highly efficient means of radius estimation. A redesign of this first step to reduce computational complexity would greatly improve the speed of the algorithm, and sacrifice little or no segmentation accuracy if the efficacy of the initial radius estimation does not decrease.

In addition to its application to algorithm Step 1, we propose further exploration of the concept of using a subset of pixels, rather than the entire image, for analysis. This concept can be applied not only to sphere map initialization, but to the entire algorithm. The 3D segmentation ultimately produced is directly calculated from the set of spheres determined to be medial to the target object. Though other spheres in the image may greatly influence the evolution of these medial spheres and are necessary to label them as “medial”, the full population of non-medial spheres may not be necessary for effective sphere map optimization and segmentation. Maintaining throughout the algorithm those spheres selected by a monte carlo pixel selection method, weighted as proposed above to include a greater proportion of pixels likely to be within the target object, may drastically reduce computation time without adversely effecting segmentation results.

Moving on to other stages of the algorithm, we propose enhancements to the selection of outposts in algorithm step 3 (Section 3.3.3). Currently, secondary outposts are defined as pixels of radius 0 with an appropriate reflector count, directly adjacent to a primary outpost. An unintended effect of this operation is a blurring of object boundaries, by potentially labeling secondary outposts in all directions from a primary outpost, rather than exclusively

along the object boundary. This problem could be resolved by fitting the optimal plane (line in 2D) of the object boundary at a given outpost, and allowing this outpost to place its secondary outposts only near this plane. Recall that, by definition, each outpost must be included in the outer shell of a number of spheres. The tangent planes (or lines) of a those spheres could be used in the above fitting. Adding this constraint to outpost calculation could sharpen boundaries and reduce the number of secondary outposts incorrectly placed within image objects, reducing the necessity for correctional actions such as algorithm step 4 (Section [3.3.4](#)).

Another approach to enhancing the accuracy of outpost placement in our segmentation algorithm is to develop a statistical model of “boundary” pixels, which can be used to filter out outposts that are not likely to represent a true object boundary. All of our research thus far has used our spherical operators to model homogeneous regions, attempting to prevent spheres from crossing object boundaries. We hypothesize that sampling possibly non-spherical regions around potential outposts instead would allow us to verify their validity as object boundary pixels by comparing their local population statistics to those observed around known boundary pixels from the a priori initialization tracing. This preventative procedure may reduce the number of spurious outposts placed within homogeneous regions due to noise and tissue inhomogeneity.

A final improvement to our Shells and Spheres segmentation system envisioned by the author takes the form of an advancement of the output. Our algorithm optimizes a sphere map for the entire image, producing properly sized spheres within all image objects. Once a seed point is given, calculation of a segmentation of the object indicated is a swift, simple procedure. An iterative procedure could be developed which would automatically run segmentation flood-fill operations on all medial pixels successively, excluding those already within a segmentation to remove redundancy. This would produce a set of fuzzy segmentations of all objects in the image. Once medial/boundary point pairs were calculated, an atlas of known anatomical shape representations could be used to automatically label anatomical objects through the use of an appropriate shape matching procedure. Our system produces well-defined shapes, with robust boundary and medial information. Thus, matching these shapes to those contained in an atlas could be a relatively simple procedure. Shape matching, in far more complex circumstances, has been thoroughly investigated by other researchers. [43, 44, 45, 46, 47]

6.4.2 Further Research Studies

Beyond the algorithm improvements highlighted in the previous section, we propose a number of studies to improve our segmentation algorithm. We first propose exploration of various intelligent search schemes from the fields of artificial intelligence and computational opti-

mization that could replace the brute force approach to parameter optimization employed in our current system. This research is given low priority, due to the small percentage of total computation time attributed to parameter optimization. Improvements to the computation time of the segmentation algorithm will have a far greater effect in reducing the total time of segmentation at present. Additionally, we envision a study continuing our research into effective 2D slices to be used for our a priori manual initialization. Our research thus far has only tested three slice orientations around a specific anatomical point, and proven that the 2D slice used for initialization has a dramatic impact on the algorithm parameters calculated, and thus, on the 3D segmentation produced. A rigorous study, testing an exhaustive set of potential 2D initialization slices through a particular anatomical object in a given imaging modality would hopefully locate an even better 2D slice for initialization. Similar tests should be conducted for each particular anatomical target and imaging modality to which the algorithm is applied. The resulting information could be incorporated into an atlas of optimal 2D slices, which could be used to automatically select the optimal 2D initialization slice from a given 3D volume by finding the slice most closely matching that in the atlas.

6.4.3 Design of a New Segmentation Algorithm

Throughout our research we have maintained that our current segmentation algorithm is merely one of many potential algorithms that could make use of the Shells and Spheres framework. The current algorithm has evolved to improve both speed and accuracy through years of development, incorporating a variety of algorithm features and functions. Both the manual initialization procedure and 2D tracing prior were added to the system subsequent to its initial development. Because of this, we note that the algorithm, which was not originally designed to take advantage of a priori information of any kind, is not ideally suited to capitalize on all aspects of the given a priori data. The knowledge and experience gained through the development of our current algorithm will be invaluable in the design of a new segmentation algorithm, which is a high priority future research endeavor.

There are numerous directions that may be explored in the development of a new S&S segmentation algorithm. First, the reliance of our initial algorithm on global variance thresholds has proven to be a limitation, hampering the degree to which it can adequately account for intensity gradients within otherwise homogeneous regions or the optimization of a sphere map for multiple objects of different variance in the same image. Future algorithms should be designed with this complication in mind, striving to add local considerations for sphere growth. Conversely, we predict that further development of processes to increase the level of communication between spheres in the same object will allow us to maintain accuracy while reducing blanket measures such as global thresholds.

A brief study was presented in this dissertation exploring the validity of the normal assumption in our spherical sampling regions (Section 3.1.2.3). A secondary product of this study is information on the behavior of standard normality measures when applied to our spheres as they evolve. This small-scale study showed that the normality measures tested did, indeed, peak as spheres grew past object boundaries, implying that these measures have the potential to be used to detect boundaries. Using normality instead of VSS gradient to signal a sphere crossing a boundary would alleviate the errors caused by VSS gradients being canceled out when a sphere (usually a medial sphere) crosses multiple symmetric boundaries simultaneously, as well as potentially increase the accuracy of boundary detection and lessen the burden on subsequent steps. Normality measures may also be more robust in the face of image noise, and an approach using the peak in a signal would remove the “number of consistent observations” parameter inherent to our current VSS gradient initialization approach. Unfortunately, the peaks of the normality measures are consistently “late”, appearing only after the sphere had grown a number of radii past the boundary. This effect is seen because as a sphere grows past a boundary, it includes a small number of pixels from a new object population, while also adding more pixels from its own object population. A number of steps past this boundary are necessary for the new object pixels to “catch up”, gaining enough statistical clout to decrease the normality measure.

To alter an evolving sphere’s normality signal to produce a peak at the proper point, i.e. the radius at which the sphere first crosses an object boundary, the set of new pixels from the population on the other side of the boundary need to be given more weight in the normality

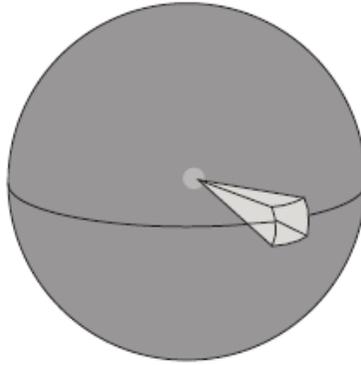


Figure 56: A truncated wedge (light) extending from the surface of a sphere (dark) to the center.

measurement. To accomplish this, the set of pixels sampled from within the sphere needs to be reduced, without reducing the number of pixels samples from across the boundary. One potential technique is to divide the outer shell of the sphere into multiple local regions, while maintaining the connection of pixels within these new sets internally to the core of the sphere. This approach to shape division, known as *truncated wedges*, was put forth by George Stetten in his dissertation. [48] Figure 56 shows a truncated wedge within a sphere. Using this approach, the surface area of the wedge containing the point of a sphere crossing an object boundary would contain only pixels from the new population. The set of pixels from the current population would contain only pixels within the wedge, a much smaller set than that of the entire sphere. Performing normality measures on the set of pixels within a truncated wedge would thus increase the proportion of pixels from across the boundary in the set on which normality is measured, and could cause the normality measure to detect a decrease in normality earlier, producing a peak in the signal at the correct point in a sphere's evolution. If normality measures can be made to reliably detect spheres crossing boundaries, a new algorithm developed incorporating this technique would remove many of the limitations of the current algorithm.

Among the many possible ways of performing image segmentation is to formulate the problem as a classification problem. A final proposed algorithm utilizing the S&S framework is such a classification algorithm. We consider the possibility of classifying spheres as “correct” or “incorrect”, based on a large set of spherical population features. Consider the space of all spheres in an image at all possible radii. A properly optimized sphere map, with each sphere sized to correctly reach the nearest boundary, represents a manifold in this space. Points along this manifold represent spheres that should be classified as “correct”, while points outside the manifold should be classified as “incorrect”. A large set of features could be explored to obtain an effective set for accurate classification. The classification itself is a two-class problem, lending itself to k-means or linear regression approaches, assuming an effective feature set. Though the ultimate difficulty in such approaches is finding a feature set that produces data clusters that can be discriminated, this reformulation of segmentation as a classification problem in the context of S&S reduces the number of classes to a known value of two, removing a great deal of complexity from other classification approaches to segmentation.

APPENDIX A

LIST OF NOTATION (TERMS)

<u>TERM</u>	<u>DESCRIPTION</u>	<u>PAGE</u>
Shells and Spheres	A framework for n -dimensional, variable-scale statistical image analysis, applied here to medial image segmentation ..	iv
a priori	A term referring to prior knowledge of an actual population, rather than current estimated knowledge	iv
Image Segmentation	The process of delineating separate objects within an image	1
Supervision	The amount of training data required by an automated or semiautomated software system	1
Sphere Map	A collection of spherical sampling regions, one centered at each image pixel, of variable radius	3
Variable-Scale Statistics	Statistics computed between two image operators, designed to be applied to operators of varying independent scales ..	3
Distance Map	A representation of an image, indicating the distance from each pixel to the nearest object boundary. Also known as a <i>distance transform</i> or <i>distance field</i>	3
Medial Spheres	Spheres centered along the medial manifold of an image object	3
Medial Manifold	Defined by Blum as “that locus of the centers of all circles completely enclosed by the boundary contour that touch the contour in more than one location.” [23]	3

Kernel	A fixed-size sampling region, representing a local set of pixels when applied to a particular location in an image; Used to calculate local statistical values in an image	4
Scale Space	The one dimensional space created by representing an image as a family of increasingly smoothed images	4
Fuzzy Set	A set defined as a pair (A, m) where A is a set and $m : A \rightarrow [0, 1]$. For each $x \in A$, $m(x)$ is the <i>grade of membership</i> of x to set A , representing the similarity of x to members of set A	11
Grade of Membership	$m(x)$ of fuzzy set (A, m) , within $[0, 1]$ representing the similarity of x to members of set A	11
Fuzzy-Connectedness	A semiautomated hybrid image segmentation technique, invented by Saha and Udupa et al. at the University of Pennsylvania	13
VSS gradient	A specialized statistical value based on the first-order moment of intensities within a sphere, corrected by a center of mass value to remove edge effects	23
Outpost	A boundary representation construct in the S&S segmentation algorithm representing a point alongside a boundary, designed to stop spheres from crossing into a new image object	36
Friendly Outpost	An outpost on the same side of a boundary as a given pixel x	43
Enemy Outpost	An outpost on the opposite side of a boundary as a given pixel x	43
Initialization Slice	The particular 2D slice used to calculate algorithm parameters for a given 3D segmentation	67

APPENDIX B

LIST OF NOTATION (ACRONYMS)

<u>ACRONYM</u>	<u>DESCRIPTION</u>	<u>PAGE</u>
3D	Three-dimensional	iv
MRI	Magnetic Resonance Imaging	iv
2D	Two-dimensional	iv
RVOT	Right Ventricular Outflow Tract	iv
TEPV	Tissue Engineered Pulmonary Valve	iv
S&S	Shells and Spheres	2
VSS	Variable-Scale Statistics	3
PV	Pulmonary Valve	3
ITK	Insight Toolkit	3
CT	Computed Tomography	13
IQR	Inter-Quartile Range	71
PA	Pulmonary Artery	73
AM-A	Alternate Method A for calculating algorithm parameters	76
IV	Variance of all pixels within the 2D target object	78
LSV	Variance of pixels within the largest sphere in the 2D target object	78
AM-B	Alternate Method B for calculating algorithm parameters	78
AM-C	Alternate Method C for calculating algorithm parameters	78
AM-D	Alternate Method D for calculating algorithm parameters	78
AA	Aortic Arch	83
GSM	Gold-Standard Sphere Map	101

ASM	Automated Sphere Map	101
SEM	Standard Error of the Mean	109

APPENDIX C

LIST OF NOTATION (VARIABLES)

<u>VARIABLE</u>	<u>DESCRIPTION</u>	<u>PAGE</u>
Ω	The domain of an image	10
$r(\mathbf{x})$	The radius of a sphere centered at pixel x	20
$S_r(\mathbf{x})$	A sphere of radius r centered at image pixel x	20
$H_r(\mathbf{x})$	A <i>shell</i> , or a set of all pixels whose distance to a central pixel x rounds to a given radius r	20
$\mu(\mathbf{x})$	The mean of the intensities of the set of pixels within the sphere centered at pixel x at its current radius	23
$\sigma^2(\mathbf{x})$	The variance of the intensities of the set of pixels within the sphere centered at pixel x at its current radius	23
$\sigma(\mathbf{x})$	The standard deviation of the intensities of the set of pixels within the sphere centered at pixel x at its current radius	23
$\mathbf{m}(\mathbf{x})$	The first-order moment of intensities of the set of pixels within the sphere centered at pixel x at its current radius	23
$\mathbf{c}(\mathbf{x})$	The center of mass of the set of pixels within the sphere centered at pixel x at its current radius	23
$\nabla f(\mathbf{x})$	The VSS gradient of the set of pixels within the sphere centered at pixel x at its current radius	24
$S^{-1}(\mathbf{x})$	The set of spheres containing pixel x	24
$\mu_\mu(\mathbf{x})$	The mean of the mean intensities for all the spheres in $S^{-1}(\mathbf{x})$	25

$\sigma_\mu(\mathbf{x})$	The standard deviation of the mean intensities for all the spheres in $S^{-1}(\mathbf{x})$	25
$z_\mu(\mathbf{x} \mathbf{y})$	The z-value, representing the likelihood of $S(\mathbf{x})$ belonging to the set $S^{-1}(\mathbf{y})$	26
g	A parameter representing the number of successive increases of VSS gradient required to deduce a true boundary in algorithm Step 1	38
α_s	A variance threshold used in Algorithm Step 2	39
β_s	A parameter governing α_s , representing the number of standard deviations away from the mean variance of all spheres in the image that the threshold is set	40
$K(\mathbf{x})$	The set of reflectors placed by a given sphere x	41
$K^{-1}(\mathbf{x})$	The set of spheres that have placed a reflector at pixel x ..	42
$\mathbf{k}(\mathbf{x})$	The reflectance at pixel x	42
P	The set of all outposts in an image	43
P'	The set of all primary outposts in an image	43
κ	The number of reflectors a pixel must contain to be labeled a primary outpost	43
P''	The set of all secondary outposts in an image	43
λ	The number of reflectors a pixel must contain to be labeled a secondary outpost	43
$E_r(\mathbf{x})$	The set of enemy outposts (those with reflectance facing \mathbf{x}) within the sphere of radius $r(\mathbf{x})$	43
γ	The number of outposts a sphere is allowed to contain after Step 3	44
α_g	A variance threshold used in Algorithm Step 4	45
β_g	A parameter governing α_g , representing the number of standard deviations away from the mean variance of all spheres in the image that the threshold is set	45

B	A dense set of boundary pixels, defined as all pixels of radius 0 or 1 in S&S algorithm Step 5	46
$\mathbf{s}(b)$	An orientation vector extending from pixel b , estimating the direction orthogonal to the boundary	46
M	The set of all medial pixels in an image	46
C	A set of connected medial pixels defining an object segmentation	48
$DSC(\beta_s, \beta_g)$	The similarity, as measured using the DSC, of an automated segmentation performed using parameters β_s and β_g to a corresponding gold standard segmentation	53
D	The DSC similarity measured at a given point during the parameter optimization process	53
D_{max}	The maximum DSC produced during parameter optimization	54
j	The increment by which parameters are adjusted in a brute force parameter optimization	54
β_{si}	The initial value of β_s in a range of parameter values over which brute force parameter optimization is performed ...	54
β_{sf}	The final value of β_s in a range of parameter values over which brute force parameter optimization is performed ...	54
β_{gi}	The initial value of β_g in a range of parameter values over which brute force parameter optimization is performed ...	54
β_{gf}	The final value of β_g in a range of parameter values over which brute force parameter optimization is performed ...	54
R	A set of spheres within a gold standard sphere map used to calculate the range of parameter values over which to optimize	63
ρ	The mean of the internal variance values of spheres within R	63

$\sigma(\rho)$	The standard deviation of the variance values of spheres in R	63
α_{gi}	The threshold value corresponding to a given value of β_{gi}	63
α_{gf}	The threshold value corresponding to a given value of β_{gf}	63
$\nu(\delta)$	The mean variance of all spheres in M for each radius increment δ above than their correct radius value	64
α_{si}	The threshold value corresponding to a given value of β_{si} .	64
α_{sf}	The threshold value corresponding to a given value of β_{sf}	64
$T(d, s)$	A 2D manual tracing where $d \in \{1 : 10\}$ is the data set and $s \in \{axial, sagittal, coronal\}$ is the slice orientation	69
$\theta_{d,s}$	A set of parameters optimized for a given slice orientation s from 3D image d	69
Θ	The set of all optimal algorithm parameters calculated in our validation study	69
$A(\theta_{d,s}, d)$	An automated 3D segmentation performed on 3D image d using parameters $\theta_{d,s}$	69
$M(u, d)$	A manual 3D segmentation of image d produced by expert user u	70
PA	Pulmonary Artery	83
ESM	Error Sphere Map	101
$g(I)$	A specialized gradient magnitude image calculated in a pre-processing step of the Geodesic Active Contours method used in our validation study, where I is the input image ..	107

BIBLIOGRAPHY

- [1] Malte Westerhoff, *Efficient Visualization and Reconstruction of 3D Geometric Models from Neuro-Biological Confocal Microscope Scans*, Ph.D. thesis, Fachbereich Mathematik u. Informatik, Freie Universitt Berlin, Berlin, Germany, January 2004.
- [2] Paul A. Yushkevich, Joseph Piven, Heather Cody, Sean Ho, and Guido Gerig, “Geodesic snakes for user-guided segmentation of 3d anatomical objects: Significantly improved efficiency and reliability,” Tech. Rep., University of North Carolina, 2004.
- [3] Paul A. Yushkevich, Joseph Piven, Heather Cody, Sean Ho, James C. Gee, and Guido Gerig, “User-guided level set segmentation of anatomical structures with itk-snap,” *IJ - 2005 MICCAI Open-Source Workshop*, July 2005.
- [4] P. E. Danielsson, “Euclidean distance mapping,” *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, Nov. 1980.
- [5] T. Lindeberg, “Scale-space theory in computer vision, kluwer, dordrecht,” *Monograph 1994*, 1994.
- [6] Tony Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.
- [7] S.P. Hoerstrup, R. Sodian, S. Daebritz, J. Wang, E.A. Bacha, D.P. Martin, A.M. Moran, K.J. Guleserian, J.S. Sperling, S. Kaushal, J.P. Vacanti, F.J. Schoen, , and Jr. J.E. Mayer, ,” *Circulation*, vol. 102(19 Suppl 3), pp. III44–9, 2000.
- [8] Women’s Heart Foundation, ,” http://www.womensheart.org/content/HeartSurgery/heart_valve_replacment.asp, August 2006.
- [9] D. Pham, C. Xu, and J. Prince, “Current methods in medical image segmentation,” *Ann. Rev. Biomed. Engng.*, vol. 2, pp. 315–337, 2000.
- [10] L. A. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, pp. 338–353, 1965.
- [11] D. Adalsteinsson and J. A. Sethian, “A fast level set method for propagating interfaces,” *Journal of Computational Physics*, vol. 118, no. 2, pp. 269–277, 1995.

- [12] L. D. Cohen and I. Cohen, “Finite-element methods for active contour models and balloons for 2-d and 3-d images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1131–1147, 1993.
- [13] Laurent D. Cohen, “On active contour models and balloons,” *CVGIP: Image Underst.*, vol. 53, no. 2, pp. 211–218, 1991.
- [14] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [15] Xiaofeng Ren and Jitendra Malik, “Learning a classification model for segmentation,” in *Proc. 9th Int’l. Conf. Computer Vision*, 2003, vol. 1, pp. 10–17.
- [16] Punam K. Saha and Jayaram K. Udupa, “Scale-based diffusive image filtering preserving boundary sharpness and fine structures,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 11, pp. 1140–1155, 2001.
- [17] P. K. Saha, J. K. Udupa, and Dewey Odhner, “Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation,” *Computer Vision and Image Understanding*, vol. 77, no. 2, pp. 145–174, Feb. 2000.
- [18] Jean Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Inc., Orlando, FL, USA, 1983.
- [19] Jean Serra, “Introduction to mathematical morphology,” *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 3, pp. 283–305, Sept. 1986.
- [20] Matús Straka, Alexandra La Cruz, Arnold Köchl, Milos Sránek, Meister Eduard Gröller, and Dominik Fleischmann, “3d watershed transform combined with a probabilistic atlas for medical image segmentation,” in *MIT 2003*, 2003, pp.
- [21] J. K. Udupa and S. Samarasekera, “Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation,” *Graphical Models in Image Processing*, vol. 58, no. 3, pp. 246–261, May 1996.
- [22] J. Sethian, “Level set methods and fast marching methods: Evolving interfaces in computational geometry,” 1998.
- [23] H. Blum and R. N. Nagel, “Shape description using weighted symmetric axis features,” *Pattern Recognition*, 10, pp. 167–180, 1978.
- [24] Tony Lindeberg, *Discrete Scale-Space Theory and the Scale-Space Primal Sketch*, Ph.D. thesis, Department of Numerical Analysis and Computer Science, KTH, Stockholm, Sweden, May 1991.
- [25] C. A. Cois, “Shells and spheres: A novel framework for variable-scale statistical image analysis,” M.S. thesis, Department of Computer Science, University

- of Pittsburgh, Pittsburgh, PA, August 2006, <http://etd.library.pitt.edu/ETD/available/etd-05162006-155257/>.
- [26] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, 3, and 4, pp. 591–611, 1965.
- [27] R. J. Tamburo, G. J. Siegle, G. D. Stetten, C. A. Cois, K. J. Rockot, J. M. Galeotti, C. F. Reynolds III, and H. J. Aizenstein, “Localizing amygdala structure differences in late-life depression,” in *Proceedings of the Fourth IEEE International Symposium on Biomedical Imaging (ISBI 2007)*, 2007 (in press).
- [28] J.C. Bezdek, L.O. Hall, and L.P. Clarke, “Review of mr image segmentation techniques using pattern recognition,” *Medical Physics*, pp. 1033–1048, 1993.
- [29] Gordon E. Moore, “Cramming more components onto integrated circuits,” pp. 56–59, 2000.
- [30] L. R. Dice, “Measures of the amount of ecologic association between species,” *Journal of Ecology*, vol. 26, pp. 297–302, 1945.
- [31] Guido Gerig, Matthieu Jomier, and Miranda Chakos, “Valmet: A new validation tool for assessing and improving 3D object segmentation,” *Lecture Notes in Computer Science*, vol. 2208, pp. 516–523, 2001.
- [32] V. Kumar and L. N. Kanal, “The cdp: A unifying formulation for heuristic search, dynamic programming, and branch-and-bound,” pp. 1–27, 1988.
- [33] A. Zijdenbos, B. Dawant, and R. Marjolin, “Morphometric analysis of white matter lesions in mr images: Methods and validation,” *IEEE TMI*, vol. 13(4), pp. 716–724, 1994.
- [34] D Goldberg-Zimring, IF Talos, JG Bhagwat, SJ Haker, PM Black, and KH Zou, “Statistical validation of brain tumor shape approximation via spherical harmonics for image-guided neurosurgery,” *Academic Radiology*, pp. 459–46, 2005.
- [35] PhD Kelly H. Zou, PhD Simon K. Warfield, MD Aditya Bharatha, MD Clare M.C. Tempny, PhD Michael R. Kaus, PhD Steven J. Haker, PhD William M. Wells, III, MD Ferenc A. Jolesz, and MD Ron Kikinis, “Statistical validation of image segmentation quality based on a spatial overlap index,” *Academic Radiology*, vol. 11, pp. 178–189, 2004.
- [36] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig, “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,” *Neuroimage*, 2006, To appear.
- [37] Michael W. Hansen and William E. Higgins, “Relaxation methods for supervised image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 949–962, 1997.

- [38] Yan Yang, *Image Segmentation and Shape Analysis of Blood Vessels with Applications to Coronary Atherosclerosis*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, Georgia, USA, March 2007.
- [39] V. Castelles, F. Catta, T. Coll, and F. Dibos, “A geometric model for active contours,” *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.
- [40] V. Castelles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International Journal Of Computer Vision*, vol. 22, pp. 61–79, 1997.
- [41] S. Osher and J. Sethian, “Fronts propagating with curvature speed: Algorithms based on hamilton-jacobi formulations,” *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
- [42] T.S. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, and R. Whitaker, “Engineering and algorithm design for an image processing api: A technical report on itk - the insight toolkit,” *In Proc. of Medicine Meets Virtual Reality*, pp. 586–592, 2002.
- [43] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz, “Symmetry descriptors and 3d shape matching,” 2004.
- [44] F.R. Schmidt, M. Clausen, and D. Cremers, “Shape matching by variational computation of geodesics on a manifold,” 2006, pp. 142–151.
- [45] Andrey Sharf and Ariel Shamir, “Feature-sensitive 3d shape matching,” 2004.
- [46] L.S. Davis, “Shape matching using relaxation techniques,” 1977, pp. 191–197.
- [47] L.S. Davis, “Shape matching using relaxation techniques,” *PAMI*, vol. 1, no. 1, pp. 60–72, January 1979.
- [48] George Stetten, *Medial Node Models for 3D Ultrasound*, Ph.D. thesis, University of North Carolina, Chapel Hill, Chapel Hill, North Carolina, USA, December 1999.