

COMPRESSION OF 4D MEDICAL IMAGE AND SPATIAL  
SEGMENTATION USING DEFORMABLE MODELS

YAN PINGKUN

NATIONAL UNIVERSITY OF SINGAPORE

2005

COMPRESSION OF 4D MEDICAL IMAGE AND SPATIAL  
SEGMENTATION USING DEFORMABLE MODELS

YAN PINGKUN

*(B.Eng. (Electronic Engineering), USTC)*

A THESIS SUBMITTED  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
DEPARTMENT OF  
ELECTRICAL & COMPUTER ENGINEERING  
NATIONAL UNIVERSITY OF SINGAPORE

2005

*This dissertation is dedicated to*  
*my beloved wife, Yuyu,*  
*and*  
*my parents*

# Acknowledgments

There are many people whom I wish to thank for the help and support they have given me throughout the course of my Ph.D. program. My foremost thank goes to my supervisor Dr. Ashraf Kassim. I thank him for his patience and encouragement that carried me on through all the difficult times, and for his insights and suggestions that helped to shape my research skills. His valuable feedback contributed greatly to my research work, definitely including this thesis. I also thank Dr. Kuntal Sengupta, who was my former co-supervisor. His visionary thoughts and energetic working style have influenced me greatly.

I would like to thank the rest of my thesis committee members: Dr. Surendra Ranganath and Dr. Sadasivan Puthusserypady. Their valuable discussions and suggestions helped me to improve the dissertation in many ways.

This work was mostly done using the data provided by the National University Hospital (NUH) of Singapore. I would like to thank Dr. Wang Shih Chang and Dr. Borys Shuter from the Department of Diagnostic Radiology at NUH for their kindly help.

Furthermore, I am thankful to Mr. Koh Kok Yan, his wife Leong Swee Ling and their lovely daughter for their kindness and help during these years in Singapore.

I would also like to take this opportunity to thank all the students and staffs in Vision & Image Processing Lab and Embedded Video Lab, whose presence and fun-loving spirits made the otherwise grueling experience tolerable. They are: Francis

Hoon, Jack Ng, Dr. Qiao Yu, Lee Wei Siong, Ng Zhi Rong, Wang Hee Lin, Hiew Litt Teen, Subramanian Ramanathan, Shen Weijia, Tan Eng Hong, Feng Wei, Wang Chao, Wang Yong, and Saravana Kumar. I enjoyed all the vivid discussions we had on various topics and had lots of fun being a member of this fantastic group.

Last but not least, I would like to thank my parents, my parents in law and my sister for always being there when I needed them most, and for supporting me through all these years. I would especially like to thank my wife Yuyu, who with her unwavering love, patience, and support has helped me to achieve this goal. This dissertation is dedicated to them.

# Contents

<b>Acknowledgments</b>	<b>i</b>
<b>Summary</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 4D Medical Image Compression . . . . .	3
1.2 Medical Image Segmentation . . . . .	5
1.3 Thesis Focus and Main Contributions . . . . .	6
1.4 Organization of the Thesis . . . . .	8
<b>2 Related Works: Medical Image Compression</b>	<b>10</b>
2.1 Introduction to Medical Image Compression . . . . .	11
2.1.1 Predictive Coding . . . . .	12
2.1.2 Transform Coding . . . . .	13
2.2 Lossless Compression Using Integer Wavelet Transform . . . . .	15
2.2.1 Integer Wavelet Transform . . . . .	15
2.2.2 Set Partitioning in Hierarchical Trees (SPIHT) . . . . .	17
2.3 Video Coding Framework . . . . .	19

<b>3</b>	<b>Four-Dimensional Medical Image Compression</b>	<b>22</b>
3.1	Introduction . . . . .	22
3.2	Motion Compensated 4D Lossy-to-Lossless Medical Image Compression . . . . .	24
3.2.1	Motion Compensation Algorithm . . . . .	26
3.2.2	Encoding/Decoding Frames . . . . .	29
3.3	Compression Performance and Discussions . . . . .	31
3.3.1	Lossless Compression Performance . . . . .	33
3.3.2	Progressive Compression Performance . . . . .	34
3.4	PSNR Fluctuations Under Lossy Compression . . . . .	38
3.4.1	Previous Works . . . . .	40
3.4.2	Error Prediction . . . . .	41
3.4.3	Experimental Results . . . . .	46
3.5	Summary . . . . .	48
<b>4</b>	<b>Related Works: Medical Image Analysis</b>	<b>50</b>
4.1	Introduction . . . . .	50
4.2	Parametric Deformable Models . . . . .	54
4.3	Geometric Deformable Models . . . . .	55
4.3.1	Front Evolution Theory . . . . .	56
4.3.2	Level Set Methods . . . . .	57
4.3.3	Geometric Deformable Models . . . . .	58
4.4	Minimal Path Deformable Models . . . . .	61
4.5	Medical Image Visualization . . . . .	63
4.5.1	Volume Rendering . . . . .	63
4.5.2	Surface Rendering . . . . .	64
4.5.3	Applications . . . . .	65

<b>5</b>	<b>Minimal Path Deformable Models</b>	<b>67</b>
5.1	Introduction . . . . .	67
5.2	Finding the Minimal Path . . . . .	68
5.2.1	Implicit Prior Shape Modeling . . . . .	70
5.2.2	Worm Algorithm . . . . .	74
5.2.3	MAP Shape Estimation . . . . .	77
5.3	Results and Discussions . . . . .	78
5.4	Summary . . . . .	83
<b>6</b>	<b>Capillary Geodesic Active Contour</b>	<b>84</b>
6.1	Introduction . . . . .	84
6.1.1	MRA Image Segmentation . . . . .	85
6.1.2	Capillary Action . . . . .	89
6.1.3	CURVES . . . . .	91
6.2	Modeling the CGAC . . . . .	92
6.2.1	Free Surface Energy . . . . .	93
6.2.2	Wetting Surface Energy . . . . .	94
6.2.3	Volume Constraint . . . . .	97
6.2.4	Evolution Equation . . . . .	98
6.3	Implementation . . . . .	99
6.3.1	Level Set Evolution Equation . . . . .	99
6.3.2	Numerical Implementation . . . . .	103
6.3.3	Toolkits . . . . .	104
6.4	Results and Discussions . . . . .	104
6.4.1	Illustration of Capillary force . . . . .	105
6.4.2	Segmentation Results of 3D MRA Images . . . . .	107
6.5	Summary . . . . .	113



<b>7</b>	<b>Conclusions</b>	<b>117</b>
7.1	4D Medical Image Compression . . . . .	117
7.2	Medical Image Segmentation . . . . .	118
7.2.1	Minimal Path Deformable Model . . . . .	119
7.2.2	Capillary Geodesic Active Contour . . . . .	120
7.3	Future Work . . . . .	121
7.3.1	Object Based Coding . . . . .	121
7.3.2	Vasculature Measurement . . . . .	121
7.3.3	Medical Image Segmentation with Prior Knowledge . . . . .	122
<b>A</b>	<b>Deriving Level Set Evolution Equation of CGAC</b>	<b>124</b>
	<b>Bibliography</b>	<b>126</b>
	<b>List of Publications</b>	<b>141</b>

# List of Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
4D	Four-Dimensional
CALIC	Content-based Adaptive Lossless Image Coding
CGAC	Capillary Geodesic Active Contour
CGMS	Capillary Geodesic Minimal Surface
CT	Computed Tomography
CURVES	Curve Evolution for Vessel Segmentation
DCT	Discrete Cosine Transform
DPCM	Differential pulse code modulation
DSR	Dynamic Spatial Reconstructor
DWT	Discrete Wavelet Transform
EZW	Embedded Zero-tree Wavelet
GAC	Geodesic Active Contour
GOF	Group of Frames
IID	Independent and Identical Distribution
LIP	List of Insignificant Pixels
LIS	List of Insignificant Sets
LOCO-I	LOW COMplexity LOSSless COMpression for Images
LSP	List of Significant Pixels
MIP	Maximum Intensity Projection
MRA	Magnetic Resonance Angiography
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
QF	Quantification Factor
ROI	Region Of Interest
SNR	Signal to Noise Ratio
SOT	Spatial Orientation Tree
SPIHT	Set Partitioning In Hierarchical Trees
VTK	Visualization ToolKit

# Summary

Medical imaging technologies have been extensively improved over the last several decades. Medical imaging, like *magnetic resonance imaging* (MRI), *computed tomography* (CT), *positron emission tomography* (PET) and *ultrasound*, has become an essential tool for doctors in diagnosing process for its convenience, noninvasive operation and efficiency. Medical images have been extensively obtained from scanning machines in hospitals.

One of the problems associated with the popularity of medical imaging is the huge data volume of the produced medical images. A very large memory space is needed to store patient data and normally these images are required to be kept for a long time. Furthermore, with the increasing popularity of telemedicine, great volume of medical images would be transmitted over internet with limited bandwidth. The problem becomes more acute for four-dimensional (4D) medical images, which consist of three-dimensional (3D) image sequences over time (3D+Time). In this thesis, a new motion compensated lossy-to-lossless 4D medical image compression scheme is proposed. Since strong temporal similarity exists in the 4D medical image, the 3D motion compensation strategy is employed to exploit the temporal redundancy among the volumetric frames. For legal and diagnostic reasons, lossless compression is required for medical image compression. Thus, 3D integer wavelet transform is applied on each volumetric frame after motion compensation to reduce 3D spatial redundancy and the produced integer coefficients are coded by 3D *set partitioning in hierarchical trees* (3D-SPIHT), which is an embedded coding scheme

and provides progressive coding performance in nature. Encouraging results are obtained by using the proposed motion compensated progressive lossy-to-lossless 4D compression algorithm.

Another problem associated with the increasing popularity of medical imaging is the tedious work of segmenting medical images to extract diagnostic information. Due to the large data volume, manual segmentation has become impractical. Thus, automated or semi-automated segmentation methods relying on the power of modern computers have been proposed. Deformable models have gained considerable success in medical image segmentation. However, they require careful initialization, which raises heavy workload when segmenting large quantity of images. In order to simplify the initialization, a minimal path deformable model based algorithm is designed. In this approach, the work of initialization is significantly simplified into one single mouse click to choose a starting point. A proposed “worm” algorithm is employed for detecting the minimal path, which consists of the actual object contour. To make the segmentation framework more robust, an implicit statistical shape model is incorporated into the potential map for evaluating paths.

Finally, 3D *magnetic resonance angiography* (MRA) segmentation is studied in this dissertation. Although existing MRA segmentation methods can extract the main structure of the vasculature, they do have difficulties in finding small vessels, which can provide critical information for navigating and positioning in brain surgery. Inspired by the capillary action, where fluid is “sucked” into thin tubes by surface tensions, a *capillary geodesic active contour* (CGAC) is modeled and constructed to extract tiny blood vessels from MRA images. Experimental results show that the CGAC can achieve more precise segmentation when compared with other state-of-the-art algorithms.

# List of Figures

1.1	Illustration of 4D data set and a 3D frame from the 4D cardiac CT image. . . . .	3
1.2	Usage of medical image compression system in medical imaging related applications. . . . .	4
1.3	Organization and development of ideas in this dissertation. . . . .	8
2.1	A typical prediction pattern in predictive coding. . . . .	12
2.2	Encoder and decoder block diagram of predictive coding. . . . .	13
2.3	Block diagram of transform coding. . . . .	14
2.4	The forward integer wavelet transform using lifting: First the Lazy wavelet, then alternating dual lifting and lifting steps. . . . .	16
2.5	The 2D spatial orientation tree superimposed on a map of wavelet transform coefficients. . . . .	18
2.6	Frames and motion compensation in video coding. . . . .	21
3.1	Overview of the proposed motion compensated lossy-to-lossless 4D medical image compression scheme. . . . .	25
3.2	Illustration of cube matching for 3D motion estimation. . . . .	26
3.3	The block diagram of the three-step 3D cube match algorithm. . . .	27
3.4	The point with minimum MSE at different positions: (a) when it is at a center of one side, 17 points out of 26 have been searched and only 9 more points will be checked; (b) when it is located at a mid-point of one edge, 11 points have been searched and only 15 points will be checked; (c) when it is at a corner, 7 points have been searched and 19 more points will be checked. . . . .	28

3.5	A frame of the (a) original image and (b) residual image after motion compensation. . . . .	30
3.6	Reordered bit stream for progressive transmission. . . . .	31
3.7	2D samples of 4D data set A of DSR images. The brightest region in the middle represents the left ventricle of a canine heart. . . . .	32
3.8	2D samples of 4D data set B of MRI images showing an enhanced human kidney cortex, spleen and liver obtained for a urography study. . . . .	33
3.9	Lossy coding results using our 4D compression scheme with wavelet filter (2, 2) (a) on 8th frame of set A and (b) on 3rd frame of set B at 1bit/voxel. PSNR results using all key frames method and JPEG-2000 on 2D slices are also included for comparison. . . . .	35
3.10	(a) The original 90 <sup>th</sup> slice of the 8 <sup>th</sup> volume of sequence A (cardiac data). Decoded results when encoded with (1, 1) filter at 0.5bit/voxel using (b) all key frames method and (c) our 4D compression method, respectively. Decoded results when encoded with (2, 2) filter at 0.5bit/voxel (d) using all key frames method and (e) our 4D compression method, respectively. . . . .	37
3.11	The original 4 <sup>th</sup> slice of the 2 <sup>nd</sup> volume of sequence B (4D MR urography study). Decoded results using all key frames at 0.1bit/voxel with (b) (3,1) filter and (d) (2, 2) filters, respectively. Decoded results using one key frame and two intermediate frames at 0.1bit/voxel with (c) (3,1) filter and (e) (2, 2) filters, respectively. . . . .	39
3.12	Block diagram of the inverse integer wavelet transform based on the lifting scheme. . . . .	42
3.13	PSNR values of reconstructed slices with different wavelet filters decoded at 1 bit/voxel. . . . .	48
4.1	The implicit level set curve is the black line superimposed over the image grid. The location of the curve is interpolated by the pixel values of a signed distance map. The grid pixels closest to the implicit curve are shown in gray. . . . .	57
4.2	Block diagram of medical image analysis scheme incorporated with visualization. . . . .	63
4.3	Samples of visualization results generated by using (a) volume rendering technique and (b) surface rendering technique. . . . .	65
5.1	Illustration of metrication error. . . . .	69

5.2	(a) A sample of CT cardiac image. (b) Edge detection result of the image. (c) Graph weighting map produced by applying distance transform on the edge map. . . . .	69
5.3	Samples of CT cardiac image over a cardiac cycle [1]. . . . .	71
5.4	Manual segmentation results of images shown in Fig. 5.3. . . . .	72
5.5	Extracted zero level set of the largest three modes of variation. . . .	74
5.6	Illustrating the use of the worm algorithm for the synthetic image in (a), which consists of sharp corners and two breaks. In (b), the worm stops at the intersection point. (c) The final contour detection result. . . . .	79
5.7	Illustration of segmentation process for CT cardiac image in (a) with two initial points, where $\alpha = 0.6$ for the epicardium wall and $\alpha = 0.2$ for the endocardium wall. (b) Detected edges. (c) Segmentation results without prior shape influence. (d) Initial shape estimates. (e)–(h) Intermediate segmentation results. (i) The final segmentation results. (j) Manual segmentation results. . . . .	80
5.8	Segmentation results on MR brain images. In both experiments, we set $\alpha = 0.8$ . The first row and the second row show the segmentation processes for data set one and data set two, respectively. First column: original images. Second column: edge maps. Third column: starting points and initial shape estimates. Fourth column: final segmentation results. Fifth column: Manual segmentation results. .	81
5.9	The effect of varying parameter $\alpha$ on the segmentation errors. . . .	82
6.1	Maximum intensity projection of a cerebral MRA data set. . . . .	86
6.2	Illustration of capillary action. (a) Capillary tube, (b) Surfaces of a three-phase system. . . . .	89
6.3	3D tubular surface in (a) is stretched to get 2D surface in (b). Minimizing lower area of the 2D surface through evolving the contact line. . . . .	95
6.4	The evolving direction of the contact line is the tangential sub of the surface normal direction. . . . .	97
6.5	Illustration of the magnitude variation of $1 - \cos^2 \theta$ with respect to the value of angle $\theta$ . . . . .	100
6.6	Illustration of various parameter settings for the sigmoid function $f$ . (a) Effects of varying $a$ under $b = 0.5$ ; (b) Effects of varying $b$ under $a = 0.05$ . . . . .	102

6.7	Illustration of the effects on varying capillary force coefficient $\alpha$ . (a) Generated cylinder. (b) Initialization of the algorithm. (c) $\alpha = 0$ . (d) $\alpha = 0.25$ . (e) $\alpha = 0.5$ . (f) $\alpha = 0.75$ . . . . .	106
6.8	Samples of MRA data set A. Bright regions and points are blood vessels. . . . .	108
6.9	MIP of the region of interest of cerebral MRA data set A. . . . .	109
6.10	MRA segmentation results of the CURVES algorithm with different view points. . . . .	110
6.11	MRA segmentation results of the proposed CGAC algorithm with different view points. . . . .	111
6.12	MIP of the 3D MRA image B. . . . .	112
6.13	Segmentation results of MRA image B using the fuzzy connectedness method [2]. . . . .	112
6.14	Segmentation results of MRA image B using the CGAC method. . .	113
6.15	MIP of the MRA image C. . . . .	114
6.16	Segmentation results of the 3D MRA image C using our CGAC method. . . . .	114
6.17	MIP of the MRA image D. . . . .	115
6.18	Segmentation results of cerebral MRA data set D using the CGAC. .	116



# List of Tables

3.1	Lossless performance of different integer wavelet filters. (The data is given in bits per voxel.) . . . . .	34
3.2	1D simulation results (with theory prediction values in parentheses)	47

# Chapter 1

## Introduction

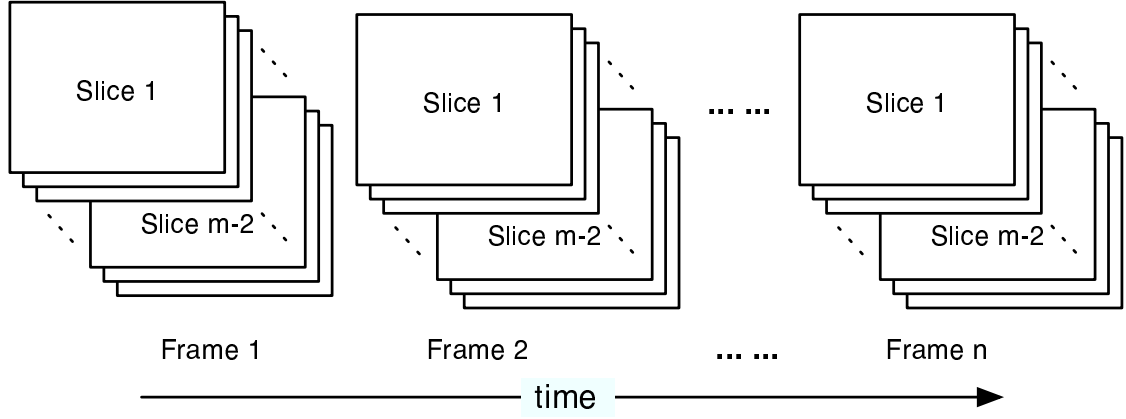
The discovery of X-rays in 1895 heralded a new era in the practice of medicine. It is a milestone in the development of diagnostic techniques. With X-rays, it became possible to visualize the internal parts of the body without painful and often dangerous surgery. Since then, medical imaging utilizing the transmission of radiant energy (*e.g.*, X-rays, gamma rays, radio waves, or ultrasound waves) through the body to produce images without subjective sensations has been widely studied. In medical imaging processes, a beam of radiation passing through the body is absorbed and scattered by structures in the beam path to varying degrees, depending on the composition of these structures and on the energy level of the beam. The differential absorption and scatter pattern by tissues within the body are recorded by a detector to produce an image of the tissues. Since a variety of sources of radiant energy are available that can be administered at levels selected and/or controlled to readily penetrate and be absorbed to some degree by all bodily tissues, radiographic images can be produced for every body organ ranging in density from bone to lung. The images produced from radiant emanations passing through parts of the body provide a direct recording of internal, unseen structures.

Over the years, there have been numerous improvements in the basic tomographic methodology. These advances have been spurred by the development of

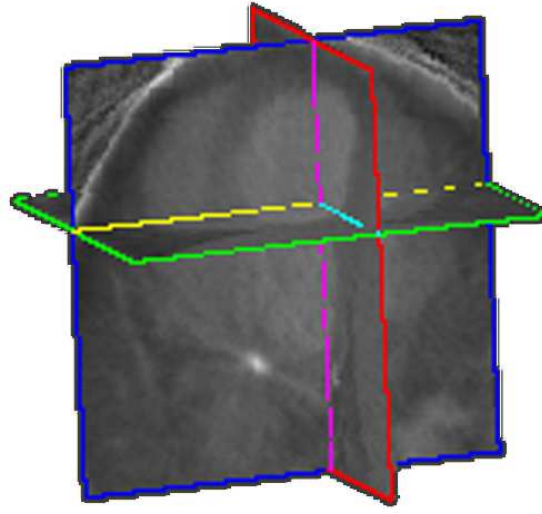
more sophisticated and powerful instruments and techniques using a variety of energy forms that have broadened and refined applications of medical imaging. The physician is now provided with significant capabilities for noninvasive examination of internal structures of the body with accuracy and specificity not ever before available. In the past several decades, medical imaging has become an essential tool in medical diagnostic processes due to its noninvasive operation and high spatial resolution. Modern computers have made possible the development of several new imaging modalities that use different sources of radiant energy to elucidate different properties of body tissues. These methods permit significant potential for providing greater specificity and sensitivity in clinical diagnostic and basic investigative imaging procedures than ever before possible. Medical imaging modalities like *magnetic resonance imaging* (MRI) and *computed tomography* (CT) have revolutionized the diagnostic capabilities of radiologists and they are considered to be among the most important advances in medical science.

Along with the improvement of medical imaging resolution and sensitivity, multi-dimensional imaging systems are rapidly developed at the same time. Three-dimensional (3D) and four-dimensional (4D) medical images have been extensively obtained from scanning machines. Three-dimensional image is composed by a stack of two-dimensional (2D) slices and 4D image consists of 3D image sequences over time (3D+Time) (see Fig. 1.1(a)). A 3D frame of 4D data set is shown in Fig. 1.1(b). With the help of 3D and 4D medical imaging techniques, doctors can observe a specific organ in 3D space directly and clearly, and even watch its activity continuously over a period of time.

The increasing popularity of medical imaging has led to rapid improvement of techniques for medical image processing. One of the primary issues addressed in this thesis is efficient compression of 4D medical images. Another key issue discussed in this dissertation is labeling and characterizing organs in 3D medical images through segmentation as well as visualization of the results.



(a) Illustration of 4D data set



(b) A 3D frame of 4D medical image

Figure 1.1: Illustration of 4D data set and a 3D frame from the 4D cardiac CT image.

## 1.1 4D Medical Image Compression

In order to preserve high spatial resolution of medical images, large numbers of pixels/voxels are required to represent a medical image, where voxel is the basic element of volumetric image just like pixel of 2D image. Hence, the size of medical image is usually very large. Since extensive amounts of medical images are being produced by medical imaging techniques, this leads to a major memory storage problem, which is further exacerbated when dealing with 3D or 4D image data sets. Typically, even a few seconds of volume cardiac image sequences can consume a few hundred mega-bytes of storage space. In addition, with the increasing popularity of

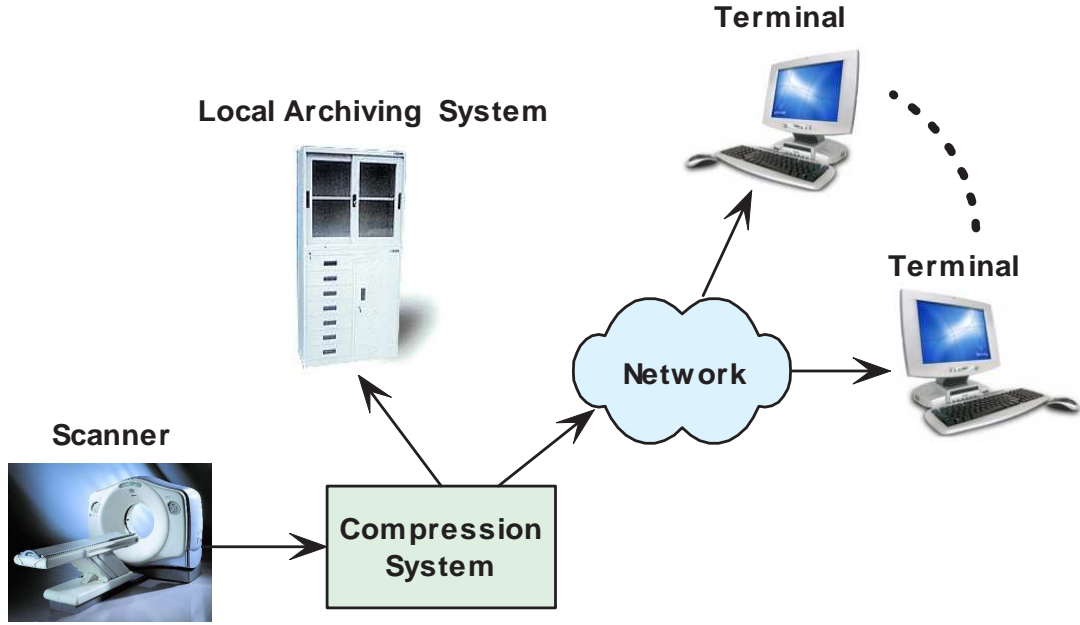


Figure 1.2: Usage of medical image compression system in medical imaging related applications.

telemedicine, large amount of data sets need to be transmitted over channels with limited bandwidth. Thus, it is highly desirable to have an effective way to compress and accommodate the rapid growth of medical imaging and to reduce storage and bandwidth costs of medical information systems as shown in Fig. 1.2. Efficient medical image compression can not only significantly enhance the performance of medical image archiving and communication systems but also may be considered an enabling technology for telemedicine.

The 4D medical image consists of a sequence of 3D spatial volumetric frames over time. Except spatial redundancy inside each frame, these sequences have significant temporal coherence among their frames, which can be used to compress 4D medical images more effectively. However, most current research in medical image compression focuses on the compression of 2D or 3D images only [3–5]. There has been little work done on compression of 4D medical images, which requires many new issues to be explored [6, 7]. In this dissertation, we describe how 4D medical image data can be compressed using the 3D motion estimation algorithm and lossy to lossless volumetric image compression techniques. Three-dimensional

motion prediction can effectively exploit the temporal redundancy inside the 4D medical image. Similar to 2D motion prediction in video compression scheme, 3D motion estimation is essential for 4D compression. The compression ratio can be increased significantly by exploiting the temporal redundancy effectively.

## 1.2 Medical Image Segmentation

The digital revolution and the rapid growing processing power of the modern computer in combination with medical imaging modalities have helped doctors to achieve more accurate diagnosis and surgery. It also helps people to better understand the complex human anatomy and its behavior to a certain extent. However, it is not enough to use either computers or medical scanning techniques alone to gain insight into medical images. The art of extracting boundaries, surfaces, and segmented volumes of these organs in the spatial and temporal domains is expected. This art of organ extraction is *segmentation*. Computer algorithms for the segmentation of anatomical structures and other regions of interest are becoming a key component in assisting and automating specific radiological tasks. A large number of algorithms have been proposed for biomedical imaging applications such as the quantification of tissue volumes [8], diagnosis [9], localization of pathology [10], study of anatomical structure [11], treatment planning [12], partial volume correction of functional imaging data [13], and computer integrated surgery [14, 15].

In this dissertation, we first present a method for applying the minimal path deformable model [16] to obtain organ contours. Segmentation is realized through finding the minimal path, which is obtained by using an “intelligent worm” algorithm. The algorithm requires a very simple initialization compared to other deformable models and has been used to segment medical images. With the *internal energy* and *external energy* defined, the worm can avoid local minima and join disconnected parts of the object contour. The prior knowledge of the shape is

incorporated into the segmentation process to achieve more robust segmentation by constructing the statistical prior shape model. Segmentation of 3D *magnetic resonance angiography* (MRA) image is studied in this dissertation as well. The proposed algorithm, called the capillary geodesic active contour (CGAC), models capillary action where the liquid can climb along the boundaries of thin tubes. The CGAC, whose implementation is based on level set, is able to segment thin vessels and has been applied for verification on synthetic volumetric images and real 3D MRA images. When compared with other state-of-the-art MRA segmentation algorithms, our experiments show that the introduction of capillary force can facilitate more accurate segmentation of blood vessels.

### 1.3 Thesis Focus and Main Contributions

This thesis presents new and novel methodologies for compressing 4D medical image data sets and segmenting 3D medical images. Three major contributions presented in this dissertation are as follows:

- 1) **Motion compensated lossy-to-lossless 4D medical image compression scheme:** A new lossy-to-lossless 4D medical image compression scheme is introduced. In previous works, 2D and 3D medical image compression has been widely studied, however, there are few works on 4D medical image compression. These methods are not able to compress 4D medical images losslessly and efficiently because not all the redundancy is removed. Our scheme efficiently exploits the temporal redundancy between adjacent 3D frames due to a new 3D fast cube matching algorithm. The resulting 3D key and residual frames are encoded by a revised version of 3D *set partitioning in hierarchical trees* (3D-SPIHT) for progressive decoding of the whole data set. Hence, both temporal and 3D spatial redundancies are exploited. Compared

with existing compression techniques, our scheme is able to achieve much higher compression ratio of 4D medical images than existing schemes.

- 2) **Minimal path deformable model:** A new framework featuring shape priors for segmentation of medical images by extracting organ contours is introduced. For segmentation, initialization is a tedious process, especially when dealing with 3D images. Our proposed scheme greatly simplifies the initialization of the deformable model by selecting one starting point. Object boundaries are delineated by detecting a minimal path, *i.e.*, a path with the minimal combined energy. Graph searching strategy is employed to find the minimal path in a weighted graph, which is obtained by applying distance transform on the edge map of the image. The prior shape knowledge is incorporated into the segmentation process to achieve more robust segmentation by constructing the statistical prior shape model. The estimated shapes of objects of interest are implicitly represented in a weighted map of the image. Accordingly, a maximum *a posteriori* estimator is proposed to get shape estimates. Our segmentation framework overcomes the shortcomings of traditional deformable models and has been successfully applied to segment various medical images.
- 3) **Capillary geodesic active contour:** In particular, a new *capillary geodesic active contour* (CGAC), is formulated and introduced to extract vasculature from MRA images. Our model is derived from the capillary action, which is considered as an energy minimization process. The incorporated capillary force adapts the evolving surface into very thin branches of blood vessels and obtains more accurate segmentation results than existing MRA segmentation techniques as demonstrated in our experiments. The CGAC can achieve more details of vasculature. Our approach is geometric in nature and topology free due to that implicit representation of the evolving surface is used.



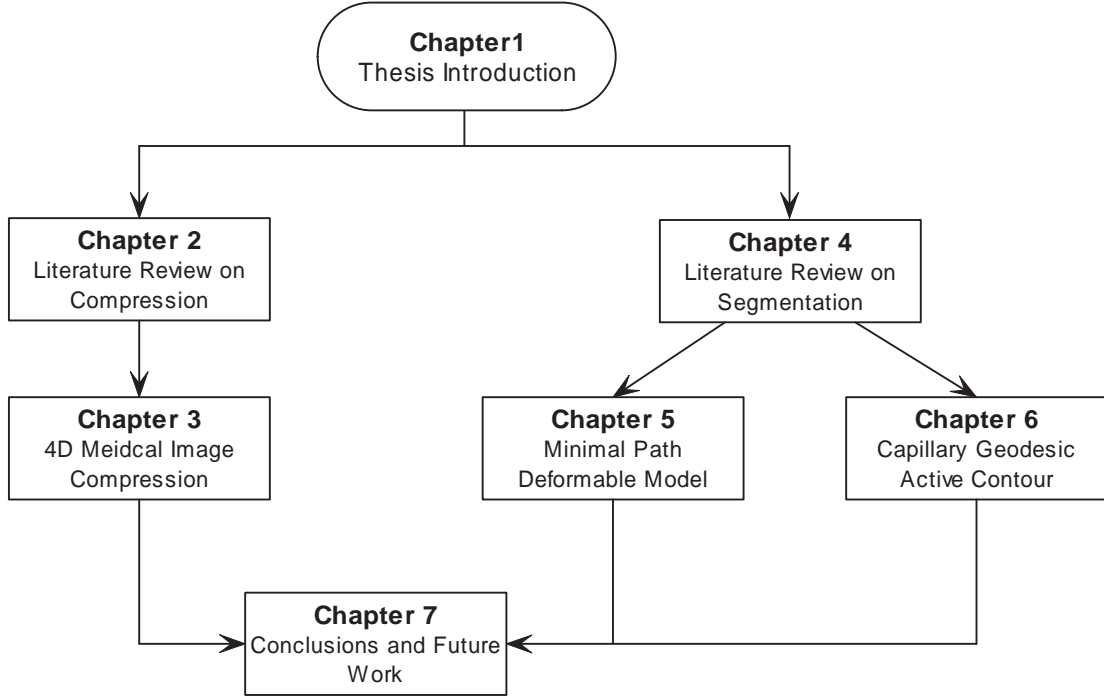


Figure 1.3: Organization and development of ideas in this dissertation.

## 1.4 Organization of the Thesis

The thesis is divided into two parts. In the first part, 4D medical image compression is discussed and 3D medical image segmentation and visualization is presented in the second part. Fig. 1.3 shows the organization and development of the ideas presented in this dissertation.

Chapter 2 provides a comprehensive literature review for the field of medical image compression. In Chapter 3, a motion compensated progressive lossy-to-lossless 4D medical image compression scheme is presented. A new fast 3D cube matching algorithm is proposed to exploit the temporal redundancy among 3D frames of 4D data set. Both its lossless compression and lossy compression performances are presented and discussed in detail.

In Chapter 4, we present a literature review of medical image segmentation as well as visualization using computer graphics techniques. New segmentation methods for medical image segmentation based on deformable models are reported

in Chapter 5 and 6. We present a minimal path deformable model in Chapter 5, which greatly simplifies the initialization task in segmentation compared with other deformable models based segmentation techniques. In Chapter 6, segmentation of vasculature from MRA is studied. A capillary geodesic active contour is reported for extracting thin vessels from MRA. Experiments are presented to demonstrate the ability of the algorithms.

Finally, in Chapter 7, we present our conclusions and indicate future research directions.

## Chapter 2

# Related Works: Medical Image Compression

The goal of data compression is to represent an information source (*e.g.* a data file, a speech signal, an image, or a video signal) as accurately as possible using the fewest number of bits. Data compression is achieved by reducing redundancies among the data. In particular, compressing still images needs reducing spatial redundancy among the pixels of an image. While for coding videos, both temporal redundancy between the successive pictures and spatial redundancy inside each picture are exploited. Furthermore, variable length coding techniques can be used to reduce the redundancy between the compressed data symbols to achieve higher compression ratio. In this chapter, we provide a brief survey on the various techniques developed for medical image compression. The basics of medical image coding are summarized in Section 2.1. In Section 2.2, the integer wavelet transform based compression scheme and video coding scheme, which are useful for 4D medical image compression, are presented.

## 2.1 Introduction to Medical Image Compression

Existing image compression algorithms can be classified into lossy and lossless techniques according to compression quality, which indicates the quality of the reconstructed image [3, 4, 17]. Lossless compression, also known as bit-preserving or reversible compression, involves exact reconstruction of the original data, *i.e.*, the data which is reconstructed from the compressed data is numerically identical to the original data [5, 18]. Obviously, lossless compression is desirable since no information is compromised. However, only limited compression can be obtained using lossless compression. On the other hand, in lossy compression (also known as irreversible compression), the reconstructed data contains degradations relative to the original [6, 7, 19]. As a result, much higher compression can be achieved as compared to lossless compression. In general, more compression is obtained at the expense of higher distortion. The precision of the data is compromised by quantizing the compressed data within a pre-specified number of bits. This introduces losses to the compression but results in cost-savings in terms of the number of bits required to represent a certain image.

In medical image compression, although lossy compression is sometimes acceptable, lossless compression is preferred [4, 17, 20, 21]. Since lossless compression does not degrade the image, it does not hinder accurate diagnosis. Lossy compression techniques could lead to errors in diagnosis, as they introduce artifacts even though the visual quality is excellent. Furthermore, there exist several legal and regulatory issues that favor lossless compression in medical applications.

Lossless compression schemes often consist of two distinct and independent components: modeling and coding. The former is concerned with the “understanding” of the source data, and is related to other knowledge based areas of computing such as machine learning and categorization techniques. In this step, spatial redundancy is reduced. In contrast, coding is a tightly specified task of ef-

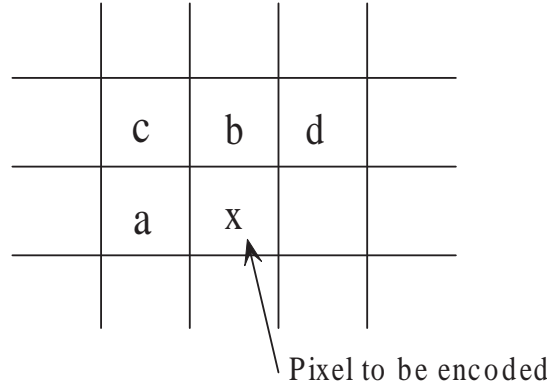


Figure 2.1: A typical prediction pattern in predictive coding.

ficiently representing a single symbol as a code, usually in binary form, given a set of estimated symbol probabilities. The redundancy among the symbols from the modeling step is reduced here. According to the modeling and coding techniques used, lossless compression algorithms can be categorized into predictive coding and transform coding [4, 22].

### 2.1.1 Predictive Coding

Predictive coding techniques, from simple methods like *differential pulse code modulation* (DPCM) to advanced ones such as the *low complexity lossless compression for images* (LOCO-I) [23] and the *context-based adaptive lossless image coding* (CALIC) [24], have obtained considerable success for lossless compression of 2D images. The LOCO-I has even been adopted as the core algorithm of the lossless image compression standard JPEG-LS due to its good performance and low complexity.

The motivation of predictive coding techniques is to remove redundancy between neighboring pixels by predicting the value of the current pixel on the basis of past pixels in some fixed order (say, raster order going row by row, left to right within a row as shown in Fig. 2.1). If we denote the current pixel by  $x$  and its predicted value by  $\hat{x}$ , then only the prediction error,  $e = \hat{x} - x$ , needs to be encoded.

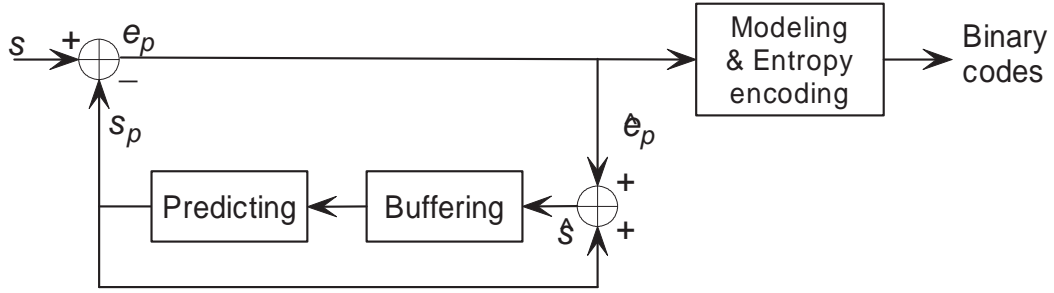


Figure 2.2: Encoder and decoder block diagram of predictive coding.

If the prediction is reasonably accurate then the distribution of prediction errors is concentrated near zero and has a significantly lower average bit rate than coding the original image directly. Thus, an optimal predictor is supposed to minimize the prediction error.

If the residual image consisting of prediction errors is treated as an *independently and identically distributed* (IID) source, then it can be coded efficiently using standard variable-length entropy coding techniques such as Huffman coding or arithmetic coding. Unfortunately, even after applying the most sophisticated prediction techniques, generally the residual image has ample structure which violates the IID assumption. Hence, in order to encode prediction errors efficiently, *error modeling* or *bias cancellation* is inserted before entropy coding. In this step, the prediction error at each pixel is encoded with respect to a conditioning state or context, which is arrived at from the values of previously encoded neighboring pixels. Fig. 2.2 shows a general predictive encoder. A symbol  $s$  is subtracted by the predicted value  $s_p$  and the residual  $e_p$  is modeled and encoded by using entropy coding. The decoder is the reverse of the encoder.

### 2.1.2 Transform Coding

Transform coding techniques have been widely used in image compression. The JPEG image compression standard is based on the *discrete cosine transform* (DCT) and *wavelet transform* has been adopted by the JPEG-2000 standard [25].

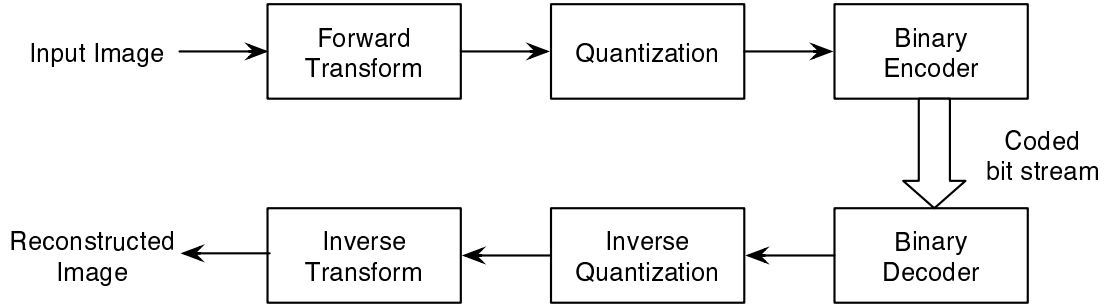


Figure 2.3: Block diagram of transform coding.

In transform coding, compression is achieved by transforming the image, projecting it on a basis of functions, and quantizing and encoding the coefficients (see Fig. 2.3). An optimal transform should be able to minimize the correlation among resulting coefficients, so that scalar quantization can be employed without losing too much in coding efficiency compared to vector quantization and to compact the energy into as few coefficients as possible. In addition, because of the nature of the image signal and the mechanisms of human visual system, the transform used for compression must accept nonstationarity and be well localized in both the space and frequency domains. Since the wavelet transform satisfies all of these conditions, it has become the most popular transform for image compression [26]. In addition, the introduction of *integer wavelet transform* [27] makes it possible to compress medical images losslessly with the power of wavelet transform coding.

The basic idea of the wavelet transform is to represent any arbitrary function  $f$  as a superposition of wavelets. Any such superposition decomposes  $f$  into different scale levels, where each level is then further decomposed with a resolution adapted to the level. Since wavelet transform owns very good localization in both the space and frequency domains, wavelets based image coding schemes can yield good compression results. This has been demonstrated by compression schemes like the *embedded zero-tree wavelet* (EZW) [28] and the *set partitioning in hierarchical trees* (SPIHT) [29] for effective reordering and coding of the wavelet coefficients into scalable and rate controllable data bit-streams. The SPIHT image coding scheme incorporates desirable features such as simple prioritization and segregation

of significant data into ordered bit planes.

## 2.2 Lossless Compression Using Integer Wavelet Transform

Although both predictive coding and transform coding can realize lossless image compression, predictive coding algorithms are difficult to be extended to 3D for volumetric image compression. In contrast, if an orthogonal transform basis is used, the wavelet transform can be naturally extended to 3D or even higher dimensions. In addition, the introduction of integer wavelet transform has made it possible to use wavelet transform compression scheme for medical image compression [27]. The performance of the wavelet based medical image compression algorithms has been demonstrated on 2D [20, 30] and 3D data [3, 5, 19, 21, 31]. Another advantage of wavelet transform compression is that it can provide *progressive lossy-to-lossless* image compression, which offers better image quality with increasing bit rate until the original image is recovered [3, 31]. Progressive coding is naturally supported when embedded coding techniques like EZW [28] or SPIHT [29] are used to code transform coefficients.

In the rest of this section, we first provide a brief introduction to integer wavelet transform in Section 2.2.1. In Section 2.2.2, the state-of-the-art SPIHT coding algorithm is reviewed.

### 2.2.1 Integer Wavelet Transform

In most cases, traditional wavelet transform produces floating-point coefficients and the use of finite-precision arithmetic for computation results in a lossy compression scheme. Part of the error comes also from the limited precision of quantization



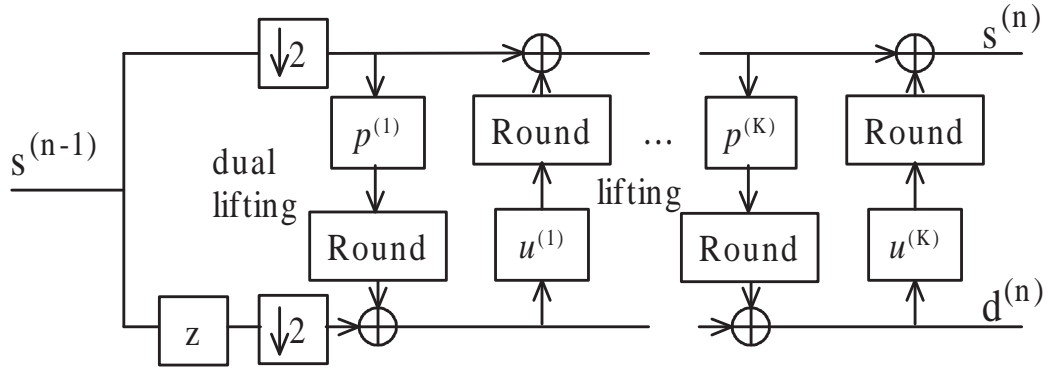


Figure 2.4: The forward integer wavelet transform using lifting: First the Lazy wavelet, then alternating dual lifting and lifting steps.

for compression purpose. In order to realize lossless compression, integer wavelet transform [27], which maps integers to integers, is then used. Integer wavelet transforms were used to be difficult to construct. However, the construction becomes very simple with lifting scheme [32]. Lifting is a flexible technique that has been used for easy construction and implementation of wavelets. In particular, many traditional non-integer wavelet transforms can be approximated by an integer version using lifting if rounding operation is added to each lifting step before adding or subtracting.

Computing the wavelet transform using lifting steps consists of several stages. The idea is to first compute a trivial wavelet transform (the Lazy wavelet or polyphase transform) [32], and then improve its properties using alternating lifting and dual lifting steps, see Fig. 2.4. Let  $x[n]$  be a discrete time input signal. The Lazy wavelet only splits the signal into its even and odd indexed samples:

$$s_{1,l}^{(0)} = s_{1,2l} = x[2n] \quad (2.1)$$

$$d_{1,l}^{(0)} = s_{1,2l+1} = x[2n + 1] \quad (2.2)$$

A dual lifting step consists of applying a filter to the even samples and subtracting

the result from the odd ones. In integer wavelet transform, it becomes

$$d_{1,l}^{(i)} = d_{1,l}^{(i-1)} - \left\lfloor \sum_k p_k^{(i)} s_{1,l-k}^{(i-1)} + \frac{1}{2} \right\rfloor \quad (2.3)$$

A primal lifting step does the opposite: applying a filter to the odd samples and subtracting the result from the even samples. After being incorporated with rounding operation, it is represented as

$$s_{1,l}^{(i)} = s_{1,l}^{(i-1)} - \left\lfloor \sum_k u_k^{(i)} d_{1,l-k}^{(i)} + \frac{1}{2} \right\rfloor \quad (2.4)$$

where the coefficients  $p_k^{(i)}$  and  $u_k^{(i)}$  are computed by factorization of a polyphase matrix (see [32] for details). Although integers are transformed to integers in (2.3) and (2.4), coefficients  $p_k^{(i)}$  and  $u_k^{(i)}$  are not necessarily integers.

Since it is written using lifting steps, the transform is invertible and the inverse transform is obtained conveniently by reversing the lifting steps and flipping the signs as

$$s_{1,l}^{(i-1)} = s_{1,l}^{(i)} + \left\lfloor \sum_k u_k^{(i)} d_{1,l-k}^{(i)} + \frac{1}{2} \right\rfloor \quad (2.5)$$

$$d_{1,l}^{(i-1)} = d_{1,l}^{(i)} + \left\lfloor \sum_k p_k^{(i)} s_{1,l-k}^{(i-1)} + \frac{1}{2} \right\rfloor \quad (2.6)$$

### 2.2.2 Set Partitioning in Hierarchical Trees (SPIHT)

The SPIHT [29] is an embedded image coding scheme that codes wavelet transform coefficients to produce embedded bit-streams. A *spatial orientation tree* (SOT) structure is used to group and order wavelet transform coefficients into sets and subsets in a hierarchical manner as illustrated in Fig. 2.5 to facilitate coefficient prediction and coding order.

If  $w$  is an absolute maximum of the wavelet transform coefficients, the entire encoding process would have up to  $N = \lfloor \log_2 |w| \rfloor$  steps. Each encoding step

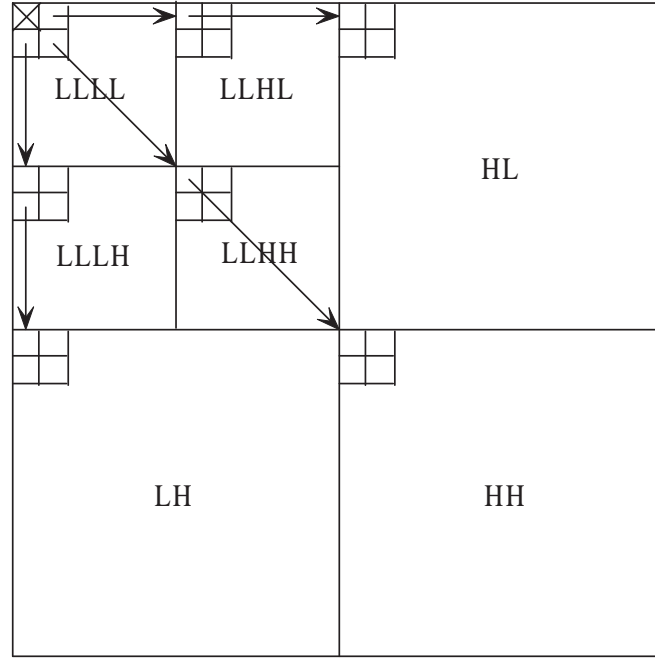


Figure 2.5: The 2D spatial orientation tree superimposed on a map of wavelet transform coefficients.

consists of a sorting pass and a refinement pass. The absolute value of the wavelet transform coefficients are compared to a threshold  $2^n$  to determine their significance at each bit plane  $n$ , where  $n \in \{0, 1, \dots, N\}$ . There are two types of significance tests, one for nodes and the other for sets. A node  $x_{i,j}$  is significant if the magnitude of the *discrete wavelet transform* (DWT) coefficient at location  $(i, j)$  is greater than the threshold. A set of nodes  $\mathcal{X}_{i,j}$  is significant if all its members have magnitudes that are greater than the threshold.

Ordering nodes and sets of wavelet transform coefficients is important for significance testing in the SPIHT to achieve efficient coding. Three ordered lists are maintained in the algorithm. Namely, *list of insignificant pixels* (LIP), *list of significant pixels* (LSP) and *list of insignificant sets* (LIS). The LIP and LIS are initialized with only root nodes in the  $LL$  subband of the SOT structure while the LSP is initially empty. During the sorting pass, the pixels in the LIP, which are insignificant in the previous pass, are tested. Significant nodes from LIP will be moved to the LSP. Significant sets in the LIS will be partitioned into subsets for further testing.

In the refinement pass, the  $n^{th}$  bit of the wavelet coefficient associated with each node in the LSP is coded. At the decoder, the wavelet coefficient values are refined as the bits are received. The encoding process continues to decrease  $n$  and alternates between sorting pass and refinement pass until any termination condition is satisfied. The encoding process can be stopped when the size of bit stream has reached the maximum or when the least significant bit plane has been processed. The output of the encoder is an embedded bit stream, which allows the decoder to reconstruct the image, of various quality at any point in the stream. The bit stream can be further encoded using coders like arithmetic encoding algorithm to get higher compression ratio [17].

## 2.3 Video Coding Framework

Since 4D image data can be represented as multiple 3D frames, *i.e.*, dynamic 3D image data, it is possible to code these 3D images independently on a 3D-frame by 3D-frame basis. However, four-dimensional medical image is normally temporally smooth and such 3D methods do not exploit the redundancy among voxels in different frames, where voxel is the basic element of volumetric image just like pixel of 2D image. This situation is very similar to that of video coding. Hence, it is natural to introduce the successful video coding framework into 4D medical image compression.

A segment of video can be considered as a sequence of 2D pictures. Since these pictures are taken continuously, in addition to the high degree of spatial redundancy in each 2D image, high degree of temporal redundancy between consecutive pictures is expected as well [33]. To exploit this, a two-stage process is employed: first stage deals with the temporal redundancy via interframe coding, while the second stage deals with the spatial redundancy via image coding techniques.

Temporal redundancy between successive frames can be reduced by finding

and coding the differences between them. For static parts of the image sequence, temporal differences will be close to zero, and hence are not coded. Those parts that change between the frames, either due to illumination variations or to motion of objects, result in significant differences, which need to be coded. This may consume a large amount of bit rates. It is noted that image changes due to motion can be significantly reduced if the motion of the object can be estimated. The differences are then considered as motion compensated image. To carry out motion compensation, the motion of the moving objects has to be estimated first. This is called *motion estimation*.

The commonly used motion estimation technique in standard video *codecs* (coders/decoders) is the block matching algorithm. In a typical block matching process, a frame is divided into blocks of  $M \times N$  pixels or, more usually, square blocks of  $N^2$  pixels. Then, for a maximum motion displacement of  $w$  pixels per frame, the current block of pixels is matched against a corresponding block at the same coordinates but in the previous frame, within the square window of width  $N + 2w$ . The best match on the basis of a matching criterion yields the displacement. To locate the best match by full search,  $(2w+1)^2$  evaluations of the matching criterion are required. This will significantly increase the encoder's computational complexity. Therefore, a number of fast search methods for motion estimation have been introduced to reduce the computational complexity of block matching algorithm [34, 35]. The basic principle of these methods is that the number of search points can be reduced, by selectively checking only a small number of specific points, assuming that the distortion measure monotonically decreases towards the best matched point.

In video coding schemes, frames in a sequence are marked as I, P or B frames and coded respectively using three different algorithms, as illustrated in Fig. 2.6 [33]. I frames (intra images) are self-contained and coded using an image coding technique. I frames are used as random access points in coding streams and they

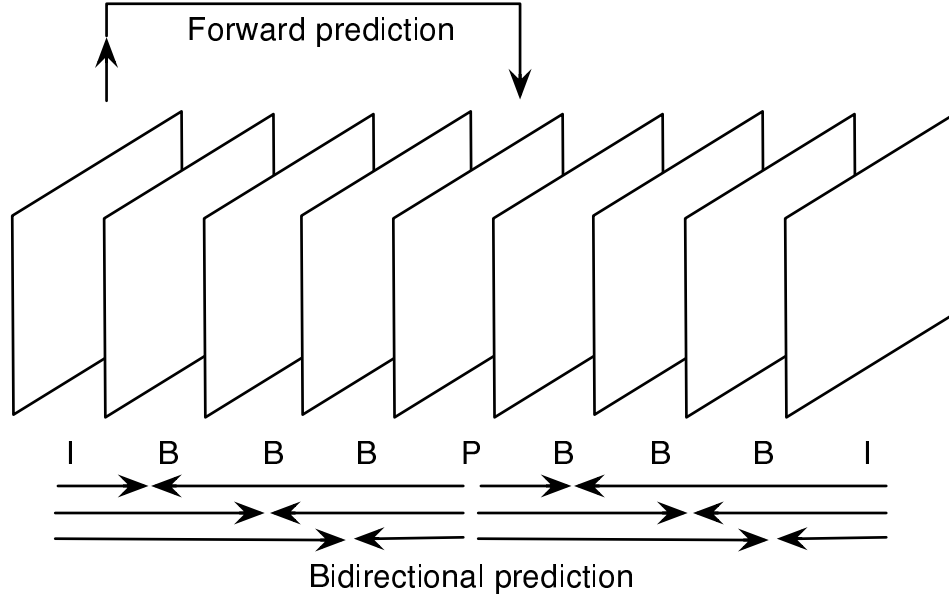


Figure 2.6: Frames and motion compensation in video coding.

normally give the lowest compression. P frames (predicted images) are motion compensated with reference to a previous frame (I or P) and the resulting residual images are coded using image coding techniques. Compressed bit rate of P frames is significantly lower than that of I frames. B frames can use forward, backward motion compensation, or simple interpolation. Thus, a block in the current frame (B frame) can be replaced by a matched block from the past reference frame, or from the future reference frame, or by the average of two blocks (see Fig. 2.6). Thus, the B frames contain only necessary information and have the highest compression ratios.

## Chapter 3

# Four-Dimensional Medical Image Compression

### 3.1 Introduction

In the last decade, the application of medical imaging has been on the rise. Among these applications, 4D imaging techniques are gaining popularity because of their ability to provide sequences of 3D volumes over time, which is very helpful when watching the cardiac cycle of a beating heart or the functioning process of some medicine in 3D space. An associated problem with the 4D medical image is that it requires large medium for storage and excessive bandwidth for transmitting. Hence, compression is needed for these images. A number of techniques have been proposed for efficient compression and transmission of 2D and 3D medical image, however, the field of 4D medical image compression has received relatively little attention.

Since 4D image data can be represented as multiple 3D frames, it is possible to code these 3D images independently on a 3D-frame by 3D-frame basis. However, such 3D methods do not exploit the dependencies that exist among voxel values

in different frames. Four-dimensional medical data is normally temporally smooth and a better approach is to consider the whole set of frames as a single 4D data set. Zeng *et al.* [6] and Peter *et al.* [7] proposed a method based on 4D discrete wavelet transform that can utilize dependencies in all four dimensions. However, they are lossy compression schemes in nature which are not suitable for medical image compression. In addition, they deem the spatial redundancy and the temporal redundancy as the same, which may hinder efficient exploiting these redundancies. Another problem associated with existing 4D wavelet transform compression methods is that all 3D volume images need to be decoded even if only one of them is to be viewed [6, 7]. Menegaz *et al.* [18] proposed a solution for this problem in 3D with a new compression scheme, which encodes volumetric images by using the 3D wavelet transform but decodes 2D slices independently. The technique reported in [18] can be extended to 4D but at the expense of higher overheads due to additional information needed for data addressing, which may reduce the compression efficiency.

To effectively compress 4D medical images, instead of extending 3D wavelet coding to 4D, a motion compensated lossy-to-lossless compression scheme is presented in this dissertation. Our scheme exploits the redundancies in the time dimension by using 3D motion compensation. A new fast 3D cube matching algorithm is proposed for this purpose. The resulted 3D frames are coded using a 3D lossy-to-lossless image compression algorithm. The lossy-to-lossless compression scheme can offer better image quality with increasing bit rate until the original image is recovered [3, 31]. In the proposed method, consecutive 3D volumetric images are divided into 3D key and 3D intermediate frames (*i.e.* volumetric images), which is similar to video coding algorithms [33]. The 3D key frames are used as the reference to predict the intermediate frames and the prediction errors are represented in the 3D residual frames. The proposed fast 3D cube matching algorithm exploits the redundancy that exists in 4D data sets for efficient compression and simplified computations. The resulting 3D key and residual frames are decomposed



into subbands by using the reversible Integer Wavelet Transform [27]. A modified 3D version of the Set-Partitioning In Hierarchical Trees (3D-SPIHT) [36, 37] coding scheme is used to encode the subband coefficients. The SPIHT algorithm is an efficient technique for coding 2D wavelet coefficients [29, 36] among others such as the Embedded Zerotree Wavelet (EZW) [28]. The extension of the SPIHT algorithm for coding 3D integer wavelet transform coefficients is also straightforward, especially when all the three dimensions are spatial. In order to realize progressive coding for all the frames, the bit stream from the normal SPIHT is reordered. The compression scheme is described in detail in the following sections.

## 3.2 Motion Compensated 4D Lossy-to-Lossless Medical Image Compression

Fig. 3.1 shows an overview of our motion compensated 4D lossy-to-lossless medical image compression system. Since 4D medical image consists of a sequence of 3D frames over time, just like video which is composed of a sequence of 2D temporal frames, the concepts of video compression are applied in coding 4D medical images. In this scheme, temporal redundancy between neighbor frames is detected and reduced by 3D motion estimation. Spatial redundancy inside each 3D frame is removed by 3D integer wavelet transform and a modified version of 3D SPIHT.

To code a group of  $n$  3D frames, the first frame is selected as the 3D key frame and the remaining intermediate frames are predicted using 3D motion estimation. Details about the motion estimation process are given in the following sections. Subtracting the 3D predicted frame from the corresponding intermediate frame results in the 3D residual frame. Thus, as illustrated in Fig. 3.1, a group of  $n$  3D image frames are processed at the encoder to produce a single 3D key frame, motion vectors and  $n-1$  3D residual frames that need to be coded. The key and

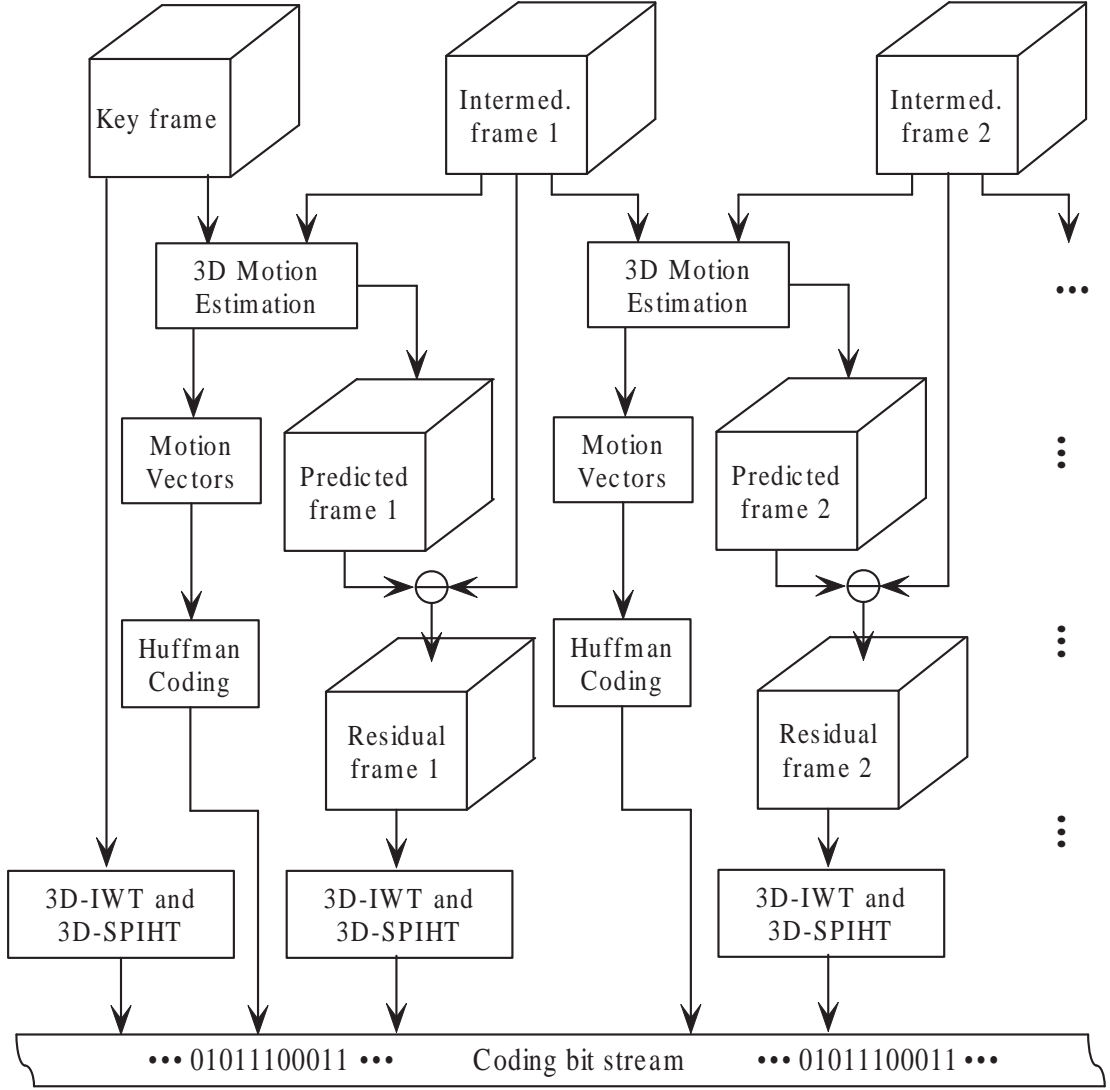


Figure 3.1: Overview of the proposed motion compensated lossy-to-lossless 4D medical image compression scheme.

residual frames are coded using 3D integer wavelet transform and 3D-SPIHT. The resulting bit stream is arithmetic coded. The motion vectors are coded by entropy coding and transmitted separately. At the decoder end, assuming that the key and residual frames are received correctly, the intermediate frames are reconstructed one after another. The key frame and the associated motion vectors are used to reconstruct the first predicted frame, which is then added to the first residual frame to reconstruct the first intermediate frame. The reconstructed first intermediate frame is then used together with its associated motion vectors and the second residual frame to reconstruct the second intermediate frame and the process continues

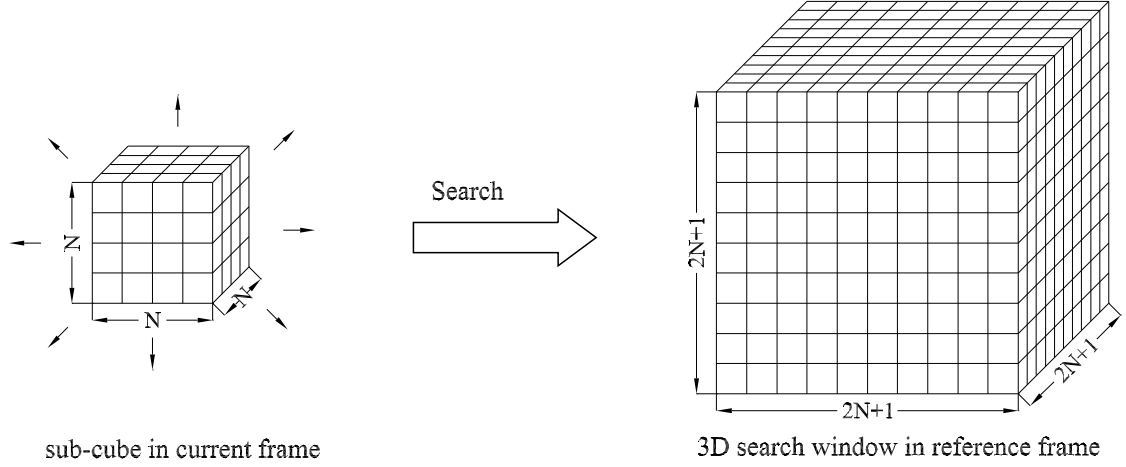


Figure 3.2: Illustration of cube matching for 3D motion estimation.

until all the  $n-1$  intermediate frames are reconstructed.

### 3.2.1 Motion Compensation Algorithm

Our 3D cube matching algorithm is an extension of the 2D block matching proposed in [34, 35], which has been extensively used in video compression. It is used in the motion compensation process to exploit the redundancy between 3D frames. In 3D cube matching, the current frame is divided into  $N \times N \times N$  sub-cubes and each sub-cube is matched inside a 3D search window in the 3D reference frame as shown in Fig. 3.2. Here, the first frame is used as the key frame and motion estimation is carried out on subsequent intermediate frames. Thus, when estimating 3D motion, the  $(n + 1)$ th frame is divided into  $N \times N \times N$  sub-cubes. The best match of each sub-cube is searched in a 3D window of the  $n$ th frame.

The motion vector  $(u, v, w)$  is essentially the difference between the position  $(i, j, k)$  of the sub-cube in current frame and the position  $(i', j', k')$  of the matched sub-cube in the reference frame. The matched sub-cube is defined as the one that has the minimum *Mean Square Error* (MSE) with the current sub-cube. The MSE,

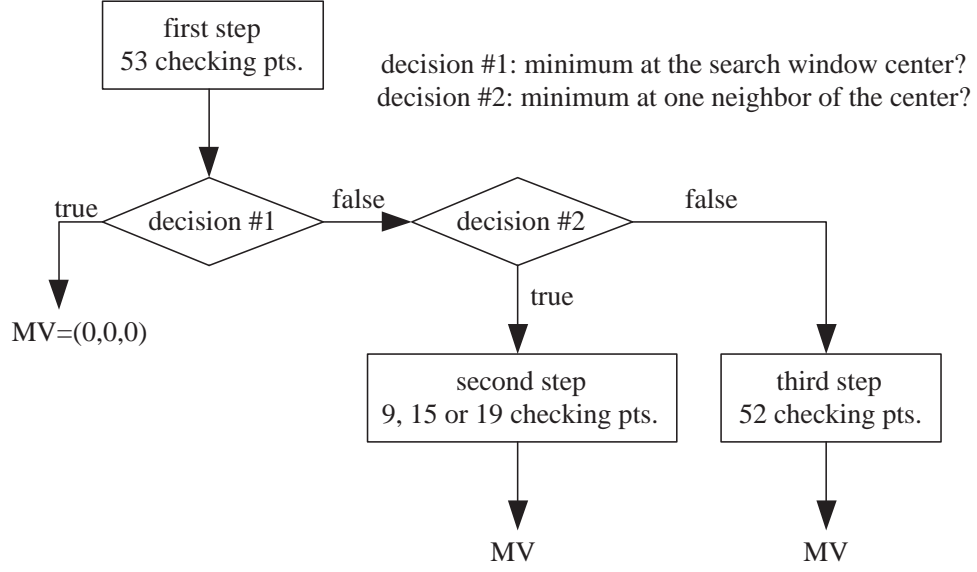


Figure 3.3: The block diagram of the three-step 3D cube match algorithm.

$E(u, v, w)$ , is calculated as follows:

$$E(u, v, w) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} [f_p(i+u, j+v, k+w) - f_c(i, j, k)]^2 \quad (3.1)$$

where  $-N \leq u, v, w \leq N$  and  $f_p(\cdot)$  and  $f_c(\cdot)$  refer to the sub-cubes in the reference frame and the current frame (to be compared), respectively. The MSE is minimized to obtain the estimates of motion vector  $(u, v, w)$ .

The 3D predicted frame can be constructed by filling its sub-cube at position  $(i, j, k)$  with the sub-cube at position  $(i+u, j+v, k+w)$  in the 3D reference frame, where  $(u, v, w)$  is the corresponding motion vector. The 3D residual frame is the difference between the 3D predicted frame and the actual 3D intermediate frame (see Fig. 3.1).

If the full search method is used, the entire search window consisting of  $(2N+1)^3$  points need to be checked. To reduce the number of points to be searched and therefore the computational complexity, a fast 3D cube match strategy is used [17], which is extended from 2D center-biased search algorithms [34, 35]. Fig. 3.3 illustrates the three-step fast 3D cube match algorithm based on a  $15 \times 15 \times 15$



- (a) If the minimum MSE in the first step occurs at the center of the search cube, the search is stopped. (This is called *first-step-stop*.)
  - (b) If the point with minimum MSE in the first step is one of the 26 points at  $\delta=1$  from the center, the global minimum is assumed to be near or at this point ( $O'_c$ ). Hence, in the second step, only the 26 points at  $\delta=1$  from  $O'_c$  will be checked before the search is stopped. However, some of these 26 points have been checked in the first step as shown in Fig. 3.4. Therefore, the number of points to be checked is less than 26. In Fig. 3.4(a), total of 18 points have already been checked including the point with minimum MSE and so only 9 more points (in black) will be checked. In Fig. 3.4(b), 15 points will be checked, while 19 points are to be checked in Fig. 3.4(c). (This is called *second-step-stop*.)
  - (c) If the point with minimum MSE in the first step is one of the 26 points at  $\delta=4$ , the point is marked as  $O''_c$  and the algorithm proceeds to step 3.
- 3) In the third step, the MSEs of the 26 points at  $\delta=2$  from the new center point  $O''_c$  are compared. The search is then performed on the 26 neighbor points of the one with minimum MSE and stopped.

In practice, the search is usually performed within an area of size  $15 \times 15 \times 15$ , and so  $N=7$ . For such a choice, the full search will check 3375 points, while our fast 3D cube matching algorithm only checks 105 points in the worst case, thus leading to a speed-up of more than 32. A residual frame, obtained by applying fast motion estimation on its original frame (see Fig. 3.5(a)), is shown in Fig. 3.5(b).

### 3.2.2 Encoding/Decoding Frames

As explained earlier, a group of  $n$  3D frames are processed to produce a single 3D key frame, motion vectors and  $n-1$  3D residual frames. The motion vectors

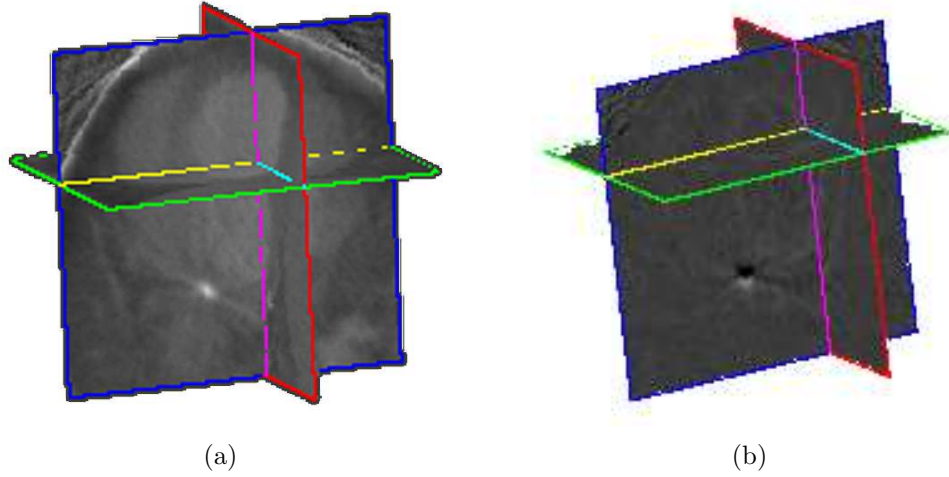


Figure 3.5: A frame of the (a) original image and (b) residual image after motion compensation.

are coded using the Huffman encoder. The 3D key and residual frames are first separately transformed using the 3D integer wavelet transform and then coded using the 3D-SPIHT scheme [37]. The decoding and reconstruction process is simply the reverse of the encoding process. The original  $n$  3D frames can be reconstructed perfectly (*i.e.*, lossless reconstruction) using the entire bit-stream.

Progressive lossy-to-lossless decoding, where the quality of the reconstructed 3D frames improves as more bits are decoded, can be easily achieved by reordering the bit-stream (as shown in Fig. 3.6) and correct decoding of the motion vectors. The coded bit stream of the key frame is normally larger than that of a residual frame. Hence, there are two segments of the key frame bit stream and one segment from each residual frame in a group as shown in Fig. 3.6. With this arrangement, the bit streams of 3D key frame and 3D residual frames can be decoded simultaneously. Hence, both the 3D key frames and the 3D residual frames can be reconstructed progressively.

To reconstruct the  $m_{th}$  frame in a group, the key frame and the preceding  $m-1$  residual frames must be decoded and the predicted frames derived from motion vector information. Each predicted frame is constructed by replacing its sub-cube at position  $(i, j, k)$  with the sub-cube at position offset by the motion vector in the

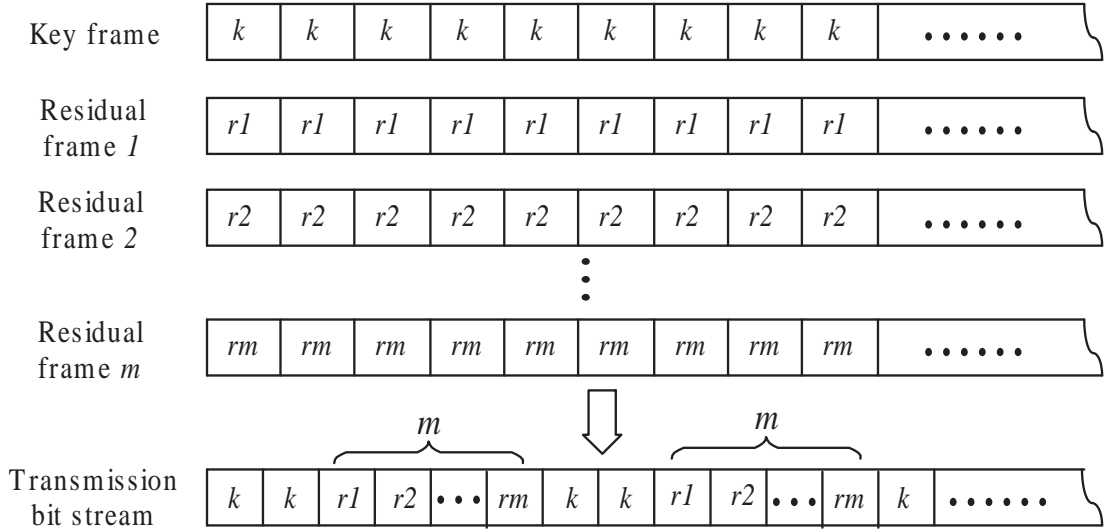


Figure 3.6: Reordered bit stream for progressive transmission.

reference frame. The residual data is then added to the predicted frames to yield the desired coded frame.

### 3.3 Compression Performance and Discussions

Two sets of 4D medical data sequences (A and B) are used in our experiments. Sequence A is a real sequence of cardiac CT volumes obtained using the *Dynamic Spatial Reconstructor* (DSR) [1] in a canine experiment. The data is stored in 16 bits/voxel. Sequence B is a set of 8 bits/voxel MRI images showing an enhanced human kidney cortex, spleen and liver obtained for a urography study. Sequence A consists of 15 frames of  $128 \times 128 \times 107$  voxels and sequence B has 33 frames of  $256 \times 256 \times 12$  voxels. Samples of data set A and B are shown in Fig. 3.7 and Fig. 3.8, respectively.

In the first experiment, all volumetric frames in each data set are treated as key frames and coded independently by using 3D integer wavelet transform and 3D-SPIHT [37]. We call this method as the all key frames method which is essentially pure 3D compression. The second experiment involves the proposed 3D motion



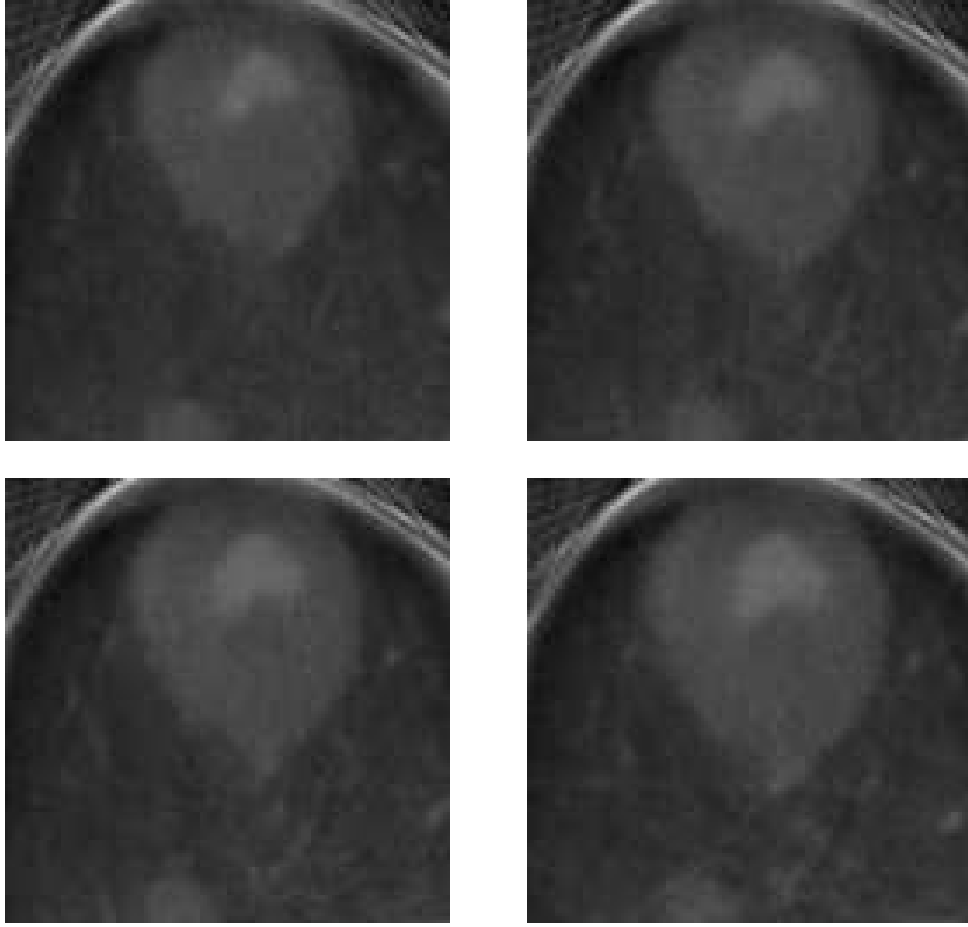


Figure 3.7: 2D samples of 4D data set A of DSR images. The brightest region in the middle represents the left ventricle of a canine heart.

compensation scheme, which uses key and predicted frames to reduce inter-frame redundancy. Each key frame is followed by two motion predicted frames. The resulting residual frames are transformed using 3D integer wavelet transform and coded using 3D-SPIHT.

In this dissertation, we use the notation  $(N, \tilde{N})$  to indicate integer wavelet filters, where  $N$  and  $\tilde{N}$  represent the number of vanishing moments of the analysis and the synthesis high-pass filters, respectively [27].

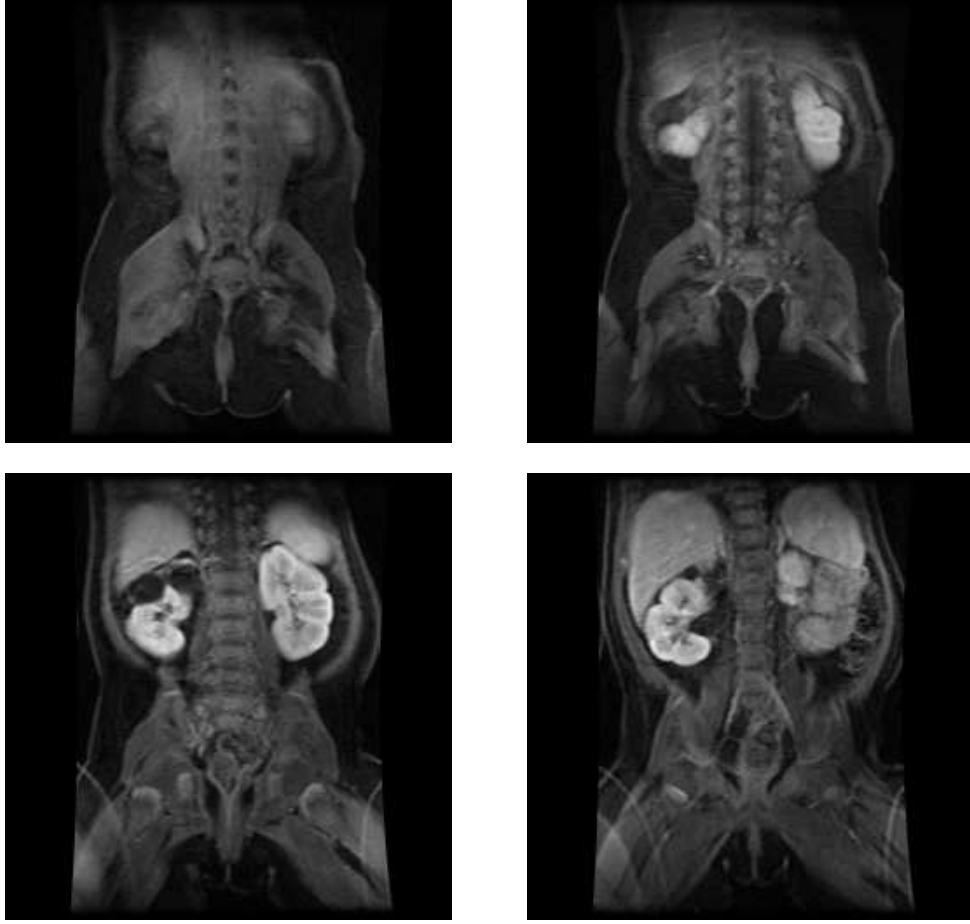


Figure 3.8: 2D samples of 4D data set B of MRI images showing an enhanced human kidney cortex, spleen and liver obtained for a urography study.

### 3.3.1 Lossless Compression Performance

Table 3.1 summarizes the lossless coding performance results using various integer wavelet transform filters. In our experiments, every three frames are grouped together to form a *group of frames* (GOF) and 3-level dyadic wavelet decomposition is applied.

The experiments show that our motion compensated 4D medical image compression scheme results in a much lower coding bit rate (by more than 25%) than the all key frames method. As the two algorithms use same wavelet filters and coding strategy in the lossless compression mode, the 3D motion compensation strategy is clearly responsible for the lower coding bit rate. The compression ratio of our scheme in Table 3.1 for lossless compression can be further increased if a

Table 3.1: Lossless performance of different integer wavelet filters. (The data is given in bits per voxel.)

Wavelet filters	All key frames (based on 3D-SPIHT)		Key and predicted frames (motion compensated 4D compression scheme)	
	Sequence A	Sequence B	Sequence A	Sequence B
(1, 1)	7.1091	2.0497	5.1388	1.4339
(2, 2)	7.1093	2.0695	5.1447	1.4148
(2, 4)	7.1363	2.0784	5.2018	1.4251
(2+2, 2)	7.2419	2.1523	5.2939	1.4815
(3, 1)	7.1261	1.8598	5.2079	1.3856
(4, 2)	9.3671	3.8668	7.8757	3.6827
(4, 4)	9.4076	3.9000	7.6381	3.7631

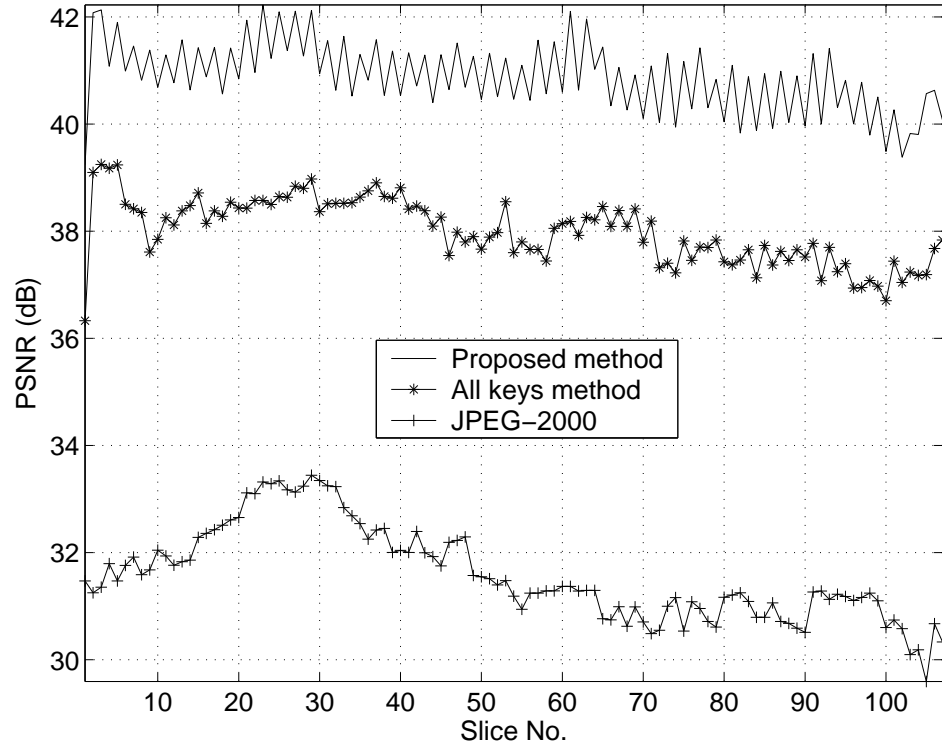
larger GOF is used.

Table 3.1 also shows that there is no single transform filter that performs best for both data sets. Different filters perform differently for each data set. However, the wavelet transform filters with fewer vanishing moments perform better than those with more vanishing moments, which is consistent with the previous 3D results in [3, 21]. This is because of the degradation in the approximation power of the wavelet basis due to the rounding operations in each lifting step while implementing integer wavelet transform through the lifting scheme, which is aggravated with the increasing of the filter length as shown in [38].

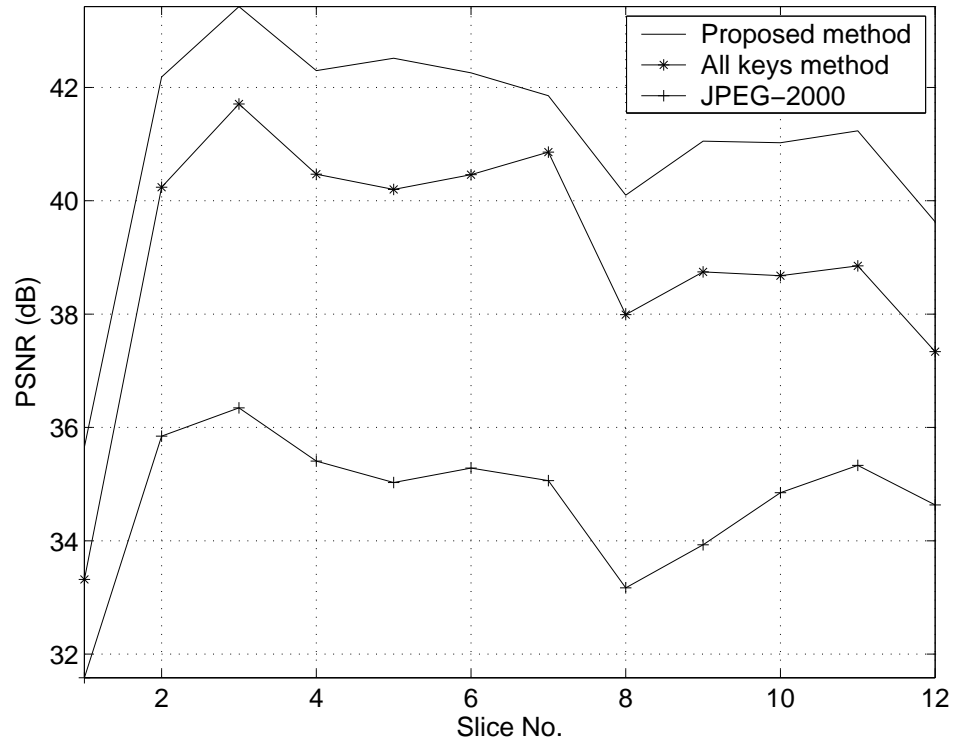
A C++ implementation of our algorithm on a Pentium 2.4-GHz PC takes about 8 seconds for lossless encoding/decoding each 3D frame of sequence A and 2 seconds for sequence B.

### 3.3.2 Progressive Compression Performance

The lossy compression performance of our algorithm is also evaluated. Since medical images are usually viewed slice by slice, the *peak signal to noise ratio* (PSNR) is calculated on each slice to evaluate the quality of the reconstructed image, which



(a) Lossy performance on data set A



(b) Lossy performance on data set B

Figure 3.9: Lossy coding results using our 4D compression scheme with wavelet filter (2, 2) (a) on 8th frame of set A and (b) on 3rd frame of set B at 1bit/voxel. PSNR results using all key frames method and JPEG-2000 on 2D slices are also included for comparison.

is defined as

$$PSNR = 10 \log_{10} \left( \frac{I_{\max}^2}{MSE} \right), \quad (3.2)$$

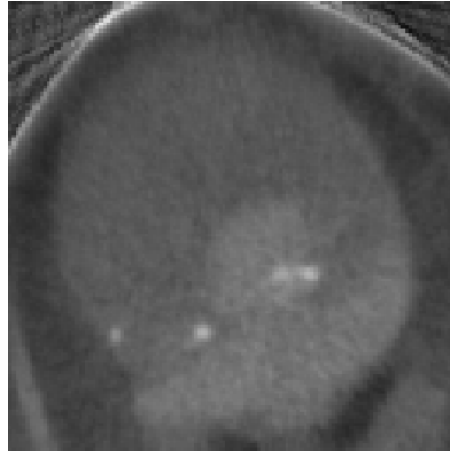
where

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - p(i, j)\|^2. \quad (3.3)$$

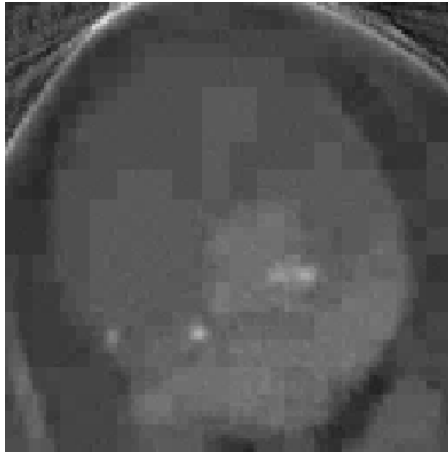
Fig. 3.9 shows plots of the PSNR versus slice number. Our algorithm is compared against the all key frames method and JPEG-2000, the *discrete wavelet transform* (DWT) based 2D image compression standard, in the lossy mode with the coding bit rate set at 1bit/voxel. Fig. 3.9 clearly shows that our proposed motion compensated compression scheme performs better than JPEG-2000 and the all key frames method. Compared with the JPEG-2000, which operates in 2D, the PSNR gain is not achieved by using integer wavelet transform. On the contrary, the rounding operations of integer wavelet transform degrades the performance on lossy image compression [38]. Our 4D compression scheme outperforms JPEG-2000 (2D) and the all key frames method (3D) in that 4D compression exploits both temporal and spatial redundancies in all four dimensions.

In Fig. 3.9(a), the PSNR fluctuations between adjacent slices are noticeable in the motion compensated 4D medical image compression scheme but minor effects are produced by JPEG-2000. The PSNR fluctuations in our scheme are caused by the 3D wavelet transform coding, which will be discussed later in Section 3.4, while the small PSNR changes with JPEG-2000 are due to content differences between slices. This phenomenon is not obvious in Fig. 3.9(b) because the number of slices within each frame is much smaller. Although the fluctuations may not be desired, they do not degrade the performance of our algorithm as the PSNR values in each slice are still much higher than that of JPEG-2000.

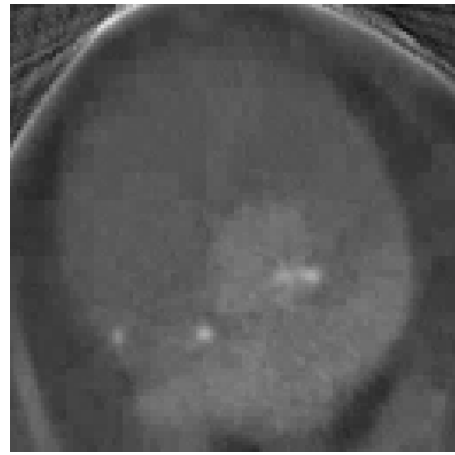
Fig. 3.10 shows the reconstructed slices of sequence A based on the all key frames method and our motion compensated 4D compression scheme using (1, 1) and (2, 2) filters with the coding bit rate set at 0.5bit/voxel. Due to its short



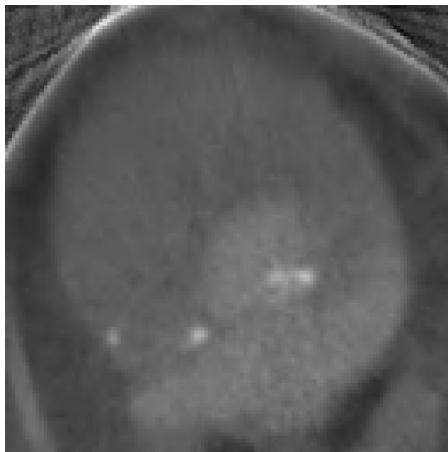
(a) Original slice



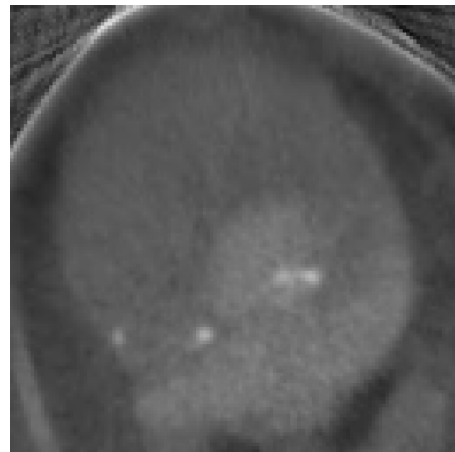
(b) PSNR 32.6dB



(c) PSNR 34.0dB



(d) PSNR 36.3dB



(e) PSNR 36.8dB

Figure 3.10: (a) The original 90<sup>th</sup> slice of the 8<sup>th</sup> volume of sequence A (cardiac data). Decoded results when encoded with (1, 1) filter at 0.5bit/voxel using (b) all key frames method and (c) our 4D compression method, respectively. Decoded results when encoded with (2, 2) filter at 0.5bit/voxel (d) using all key frames method and (e) our 4D compression method, respectively.

length, the  $(1, 1)$  wavelet filter which is also known as the Haar wavelet, produces block artifacts at high compression (see Fig. 3.10(b)). Some slight blurring can be observed when the  $(2, 2)$  filter is used. Taking into account the experimental results in Table 3.1 and Fig. 3.9, the wavelet filter  $(2, 2)$  is recommended as the default wavelet filter for our motion compensated lossy-to-lossless 4D medical image compression scheme because it works well in both lossless and lossy modes. In addition, as shown in Fig. 3.10(c) and (e), the artifacts become less perceivable when the motion compensated 4D compression scheme is applied.

Fig. 3.11 illustrates the reconstructed visual quality of data set B using the two methods decoded at 0.1bit/voxel with 3-level integer wavelet transform, respectively. It is evident that our compression scheme produces better quality decoded images. With the 3D motion compensation, the redundancy between frames is exploited, resulting in lower bit-rates for the predicted frames. Hence, our algorithm enables higher quality images to be reconstructed at the same bit rate compared with those using the all key frames method. In other words, lower bit rate is needed by our algorithm to achieve same compression quality of the all key frames method.

### 3.4 PSNR Fluctuations Under Lossy Compression

In spite of advantages brought by integer wavelet transform based 3D compression scheme, obvious PSNR fluctuations between reconstructed neighboring slices can be observed in the lossy decoding mode as described in Section 3.3.2. These unwanted oscillations could lead to critical problems on diagnosis. They are caused partly by the differences among slice contents and more heavily by the cyclostationary reconstruction errors in the inverse integer wavelet transform procedure, which are due to the quantization errors in the compression process. In the rest of this section,



(a) Original slice



(b) PSNR 29.4dB



(c) PSNR 30.5dB



(d) PSNR 31.9dB



(e) PSNR 33.1dB

Figure 3.11: The original 4<sup>th</sup> slice of the 2<sup>nd</sup> volume of sequence B (4D MR urography study). Decoded results using all key frames at 0.1bit/voxel with (b) (3,1) filter and (d) (2, 2) filters, respectively. Decoded results using one key frame and two intermediate frames at 0.1bit/voxel with (c) (3,1) filter and (e) (2, 2) filters, respectively.



we first review related works on reconstruction error analysis and then present a statistical model for analyzing errors produced by integer wavelet transform based compression system.

### 3.4.1 Previous Works

PSNR fluctuations associated with wavelet based codecs due to the cyclostationary properties of the inverse wavelet transform have aroused researcher's interests. It is also noticed that using the integer wavelet transform instead of the discrete wavelet transform degrades the performances of the lossy codecs [39] and causes more obvious oscillations. To better understand this phenomenon, there is a need to quantify these effects of integer wavelet transform in a theoretical framework.

Signoroni and Leonardi [40] model the reconstruction errors of discrete wavelet transform in 3D coding scheme as additive noises passing through linear systems. However, the reconstruction process of integer wavelet transform is more complex and can not be modeled as linear transformation, due to the rounding operations in each step of the lifting scheme for implementing integer wavelet transform [27]. Therefore, the error patterns are different and the model in [40] is not applicable for integer wavelet transform. Reichel et al. [38] compared the performance of discrete wavelet transform and integer wavelet transform based schemes under both low quality and high quality decoding conditions. They expressed integer wavelet transform as discrete wavelet transform plus the addition of rounding noise. Thus, the performance of integer wavelet transform is analyzed still by using the same model as discrete wavelet transform but with taking into account the rounding noises. In this dissertation, we propose a statistical model to analyze the cyclostationary PSNR fluctuations in 3D integer wavelet transform based compression schemes directly through investigating quantization errors and reconstruction errors under lossy compression.

### 3.4.2 Error Prediction

In the implementation of integer wavelet transform through lifting scheme, it is possible to replace the rounding operations with the addition of random floating noise to integers because every number can be represented by its nearest integer plus a floating remainder as

$$\text{round}(x) = x + \delta, \quad (3.4)$$

where  $\delta$  is the residual error. In integer wavelet transform based lossless compression, since rounding errors from the synthesis part counteract those from the analysis part, perfect reconstruction can be assured. However, if the coding/decoding is lossy, the quantization noise introduced by the compression algorithm can propagate in the lifting network and make the rounding noises deviating from the expected values. Therefore, the rounding errors may increase the distortion of signals instead of decreasing it as before. As shown in Fig. 3.12, the reconstructed data with their corresponding reconstruction errors from upper and lower channels, respectively, are mixed together after upsampling and unit delay. Such downsampling and upsampling operations can lead to a periodically time varying system in lossy compression mode [41]. This is the reason that the reconstructed data has cyclostationary PSNR fluctuations. Since the oscillations are caused by the cyclostationary reconstruction errors, we model the reconstruction errors first. Reconstruction errors are composed by the propagating quantization noises in compression stage and the induced rounding noises in inverse transform stage.

#### Compression noise

The compression noise is introduced by the quantization in data compression. In embedded lossy-to-lossless compression, the bit plane coding method is employed and the quantization step  $\Delta$  takes value of powers of two, i.e.  $\Delta = 2^Q$ , where

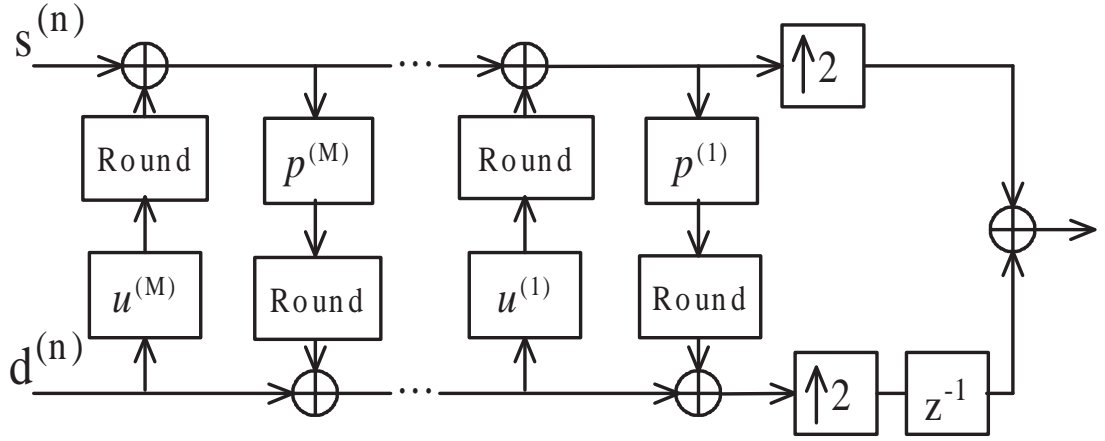


Figure 3.12: Block diagram of the inverse integer wavelet transform based on the lifting scheme.

$Q$  is *quantization factor* (QF). Assuming that the distribution of integer wavelet coefficients over a quantization interval is uniform, the mean of the quantization error is zero. Since all the coefficients are integers in integer wavelet transform, the variance of quantization error can be expressed as:

$$\sigma_{\Delta}^2 = \sum_{i=-\Delta+1}^{\Delta-1} \frac{i^2}{2\Delta} = \frac{(\Delta-1)(2\Delta-1)}{6} \quad (3.5)$$

There is no quantization error if  $Q \leq 0$ , because all the numbers are integers.

### Rounding noise

In lossy compression, rounding noise in the inverse lifting scheme is no longer able to kill its counterpart in forward transform, because of the presence of compression noise. Thus, if the input data is uniformly distributed in  $[0,1)$ , the mean of the rounding noise is zero. Its variance may be  $1^2/12$  because rounding a floating number is equivalent to quantizing it with scale one. However, the inputs  $d_m$  at each step of integer wavelet transform are all integers and the filter noise with rational coefficients can only take limited values like

$$\delta[n] = \frac{d_m}{D_m}, \quad \text{and} \quad -\frac{D_m-1}{2} \leq d_m \leq \frac{D_m-1}{2} \quad (3.6)$$

where  $D_m$  is a common denominator for all coefficients of the filter bank at step  $m$  of the lifting process. Therefore, the variance of rounding noise at the  $m_{th}$  step can be calculated as

$$\sigma_\delta^2 = \begin{cases} \frac{1}{12} \left( 1 - \frac{1}{D_m^2} \right), & D_m \text{ is odd,} \\ \frac{1}{12} \left( 1 + \frac{2}{D_m^2} \right), & D_m \text{ is even.} \end{cases} \quad (3.7)$$

Then the variance of the noise after  $m$  rounding operations can be described as

$$\begin{aligned} \sigma_R^2 = E \Big\{ & \left( \text{Round} \left( \sum_k c_{m,k} i[n-k] \right) \right. \\ & \left. - \text{Round} \left( \sum_k c_{m,k} (i[n-k] + \zeta[n-k]) \right) \right)^2 \Big\} \end{aligned} \quad (3.8)$$

where  $c_m$  represents filter coefficient as  $u_m$  or  $p_m$  in Fig. 3.12 and  $\zeta[n]$  is the error that comes with input signal. Replacing the rounding operation according to (3.4), we have

$$\sigma_R^2 = E \left\{ \left( \delta_1[n] - \sum_k c_{m,k} \zeta[n-k] - \delta_2[n] \right)^2 \right\} \quad (3.9)$$

where  $\delta_1[n]$  and  $\delta_2[n]$  are noises produced by the two rounding operations, respectively. They are computed by

$$\begin{cases} \delta_1[n] = \frac{d_{m_1}}{D_{m_1}} \\ \delta_2[n] = \frac{d_{m_1}}{D_{m_1}} + \frac{d_{m_2}}{D_{m_2}} \end{cases} \quad (3.10)$$

In lossless compression mode,  $\zeta[n]=0$  and then  $\delta_1[n]=\delta_2[n]$ , so the variance of rounding noise equals to zero according to (3.9). Assuming that  $\delta_1[n]$  is independent on  $\zeta[n]$ , Eqn. (3.9) can be simplified into

$$\sigma_R^2 = \sigma_{\delta_1}^2 - \sigma_{\delta_2}^2 + \sum_k c_{m,k}^2 \sigma_{\zeta,k}^2. \quad (3.11)$$

### Reconstruction error of 1D transform

After modeling the compression noise and rounding noise, the reconstruction error of one-dimensional (1D) transform can be computed. For the simplest one level transform case, the cyclostationary periodicity of the error is two because different errors from the upper and lower channels are mixed together as shown in Fig. 3.12. Since the lifting scheme is used, it is convenient to compute the reconstruction error at the basic transform block level. At the  $m$ th lifting segment, where  $m \in [1, 2, \dots, M]$ , the variance of reconstruction error in channel  $d$  can be computed recursively as

$$\begin{aligned} \sigma_{d,m}^2 = & \sigma_{\zeta_{d,m+1}}^2 + 2E \left\{ \zeta_{d,m+1}[n] \sum_k c_{m,k} \zeta_{\bar{d},m+1}[n-k] \right\} \sigma_{\delta_{d,m_1}}^2 \\ & - \sigma_{\delta_{d,m_2}}^2 + \sum_k c_{m,k}^2 \sigma_{\zeta_{\bar{d},m+1,k}}^2 \end{aligned} \quad (3.12)$$

where  $\sigma_{\zeta,M+1} = \sigma_{\Delta}^2$ . The error variance  $\sigma_0^2$  of the upper channel and  $\sigma_1^2$  of the lower channel can be obtained by calculating  $\sigma_{0,1}^2$  and  $\sigma_{1,1}^2$ , respectively. Given the quantization error, the variance of reconstruction error is mainly subject to the filter coefficients and the number of lifting segments. According to (3.12), filters with smaller coefficients will get lower error variance and hence lead to higher PSNR value on the reconstructed signals. Filters with fewer lifting steps will perform better than those with more steps. In addition, filters having similar coefficients in the lifting and dual lifting steps can produce less fluctuations. For simplicity, Eqn. (3.12) may also be rewritten as

$$\sigma_j^2 = H_j (\sigma_{\zeta_{0,M+1}}^2, \sigma_{\zeta_{1,M+1}}^2), \quad j = 0, 1 \quad (3.13)$$

where  $H_j$  is the system function that can be computed using (3.12).

Since one level transform can yield two different error patterns, the 1D  $L$  level transform will give out cyclostationary reconstruction error with periodicity of  $2L$ .

Its variance is defined by

$$\sigma_j^2 = H_{b_{L-1}}(\sigma_\Delta^2, H_{b_{L-1}}(\sigma_\Delta^2, \dots H_{b_0}(\sigma_\Delta^2, \sigma_\Delta^2)) = \tilde{H}_j^L(\sigma_\Delta^2) \quad (3.14)$$

where  $j=0,1,\dots,2^L-1$  can be decomposed into

$$j = \sum_{l=0}^{L-1} b_l 2^l \quad (3.15)$$

where  $b_l$  is binary with value 0 or 1.

### Three-dimensional $L$ level transforms

Since separable wavelet transform is normally used in image compression from 2D to 4D, the variance of reconstruction error can be calculated separately as

$$\sigma_{j_0,j_1,j_2}^2 = \tilde{H}_{j_0}^L \left( \tilde{H}_{j_1}^L \left( \tilde{H}_{j_2}^L (\sigma_\Delta^2) \right) \right). \quad (3.16)$$

If the same wavelet filter is used in each direction, which is normally the case [3,5,17],  $\sigma^2$  is a  $2^L \times 2^L \times 2^L$  symmetric matrix. As the volumetric image is usually viewed slice by slice, PSNR is calculated on each slice. Then the PSNR of the  $j$ th slice is described as

$$PSNR_j = 10 \times \log_{10} \left( \frac{(2^d - 1)^2}{\text{aMSE}_j} \right) \quad (3.17)$$

where  $d$  is the voxel depth and aMSE is the *average mean square error*, which is

$$\text{aMSE}_j = \frac{1}{2^{2L}} \sum_{j_1=0}^{2^L-1} \sum_{j_2=0}^{2^L-1} \sigma_{j,j_1,j_2}^2 \quad (3.18)$$

The index  $j$  can be in any direction, because matrix  $\sigma^2$  is symmetric.

### 3.4.3 Experimental Results

Both simulations and compression results of medical images are used to evaluate the proposed model. Several wavelet filters presented in [27] are adopted in the experiments for their good performance in medical image compression [3, 17, 39]. Due to the complex structure and presence of noise in medical images, wavelet filters with shorter length normally have better compression performance. Thus, wavelet filters with length 8 or shorter are used in the experiments. Three level transform is used in our experiments because it is one of the most commonly used transform levels in image compression applications.

#### Simulation results

Synthetic images with uniformly distributed random integers between 0 and 255 as their voxel intensities, which exactly matches the hypothesis of the proposed model, are used in simulation. Table 3.2 shows the 1D simulation results obtained through averaging over 10,000 images. Column 3 of Table 3.2 gives out the mean PSNR value and PSNR fluctuations are characterized in Column 4. The performance of the wavelet filters is evaluated by the mean and variance of the PSNR fluctuations. A good filter is expected to have high mean PSNR and smooth fluctuations.

Table 3.2 shows that our model predicts the reconstruction error successfully, because the theoretically predicted values are very close to the experimental results. These wavelet filters perform consistently under different quantization factors (QFs). Although their mean and variance of the reconstruction error increase with larger QF, their performance ranks remain unchanged. The (1,1) filter has the worst performance due to its lowest mean and highest variance of PSNR as shown in Table 3.2. This is caused by its large coefficients which can enhance the errors greatly according to (3.12). However, there is no such a good filter that owns both highest mean PSNR and smallest oscillations (see Table 3.2). For example,

Table 3.2: 1D simulation results (with theory prediction values in parentheses)

Filter	QF	Mean (dB)	Variance (dB)
(1,1)	Q=1	45.25 (45.18)	8.30 (8.22)
(2,2)		<b>47.39 (47.38)</b>	0.93 (0.95)
(3,1)		46.95 (47.18)	2.00 (2.07)
(2+2,2)		47.30 (47.21)	<b>0.59 (0.60)</b>
(2,4)		47.35 (47.29)	0.73 (0.75)
(1,1)	Q=2	39.64 (39.63)	10.16 (10.55)
(2,2)		<b>41.13 (41.08)</b>	4.04 (4.07)
(3,1)		40.62 (40.60)	7.14 (7.07)
(2+2,2)		40.89 (40.87)	<b>2.90 (2.77)</b>
(2,4)		40.96 (40.90)	3.30 (3.24)
(1,1)	Q=3	33.70 (33.69)	10.50 (9.98)
(2,2)		<b>34.51 (34.55)</b>	5.45 (5.69)
(3,1)		34.24 (34.20)	10.37 (10.68)
(2+2,2)		34.39 (34.35)	<b>4.42 (4.35)</b>
(2,4)		34.45 (34.43)	4.90 (4.94)

although filter (2,2) has the highest mean PSNR, its variance is much higher than that of filter (2+2,2). The reason is that, as implied by (3.12), the one more lifting step in filter (2+2,2) smoothes its PSNR fluctuations but increases the average errors.

### Experimental results on medical images

Three-dimensional volumetric frames of the 4D data set A, as described in Section 3.3, are used in the experiments. Coefficients of 3D integer wavelet transform are coded by a 3D-SPIHT [5, 17] based coding scheme incorporated with arithmetic coding.

It is clear in Fig. 3.13 that filter (2,2) outperforms the other two filters, which is consistent to the results in Table 3.2 as expected. However, the distance between the curves is much greater than the average differences in Table 3.2. One of the reasons is that the voxel intensities in real images are not uniformly distributed, which breaks the hypothesis of the modeling. Furthermore, adjacent voxels in medical image usually have strong similarities, while they are assumed independent



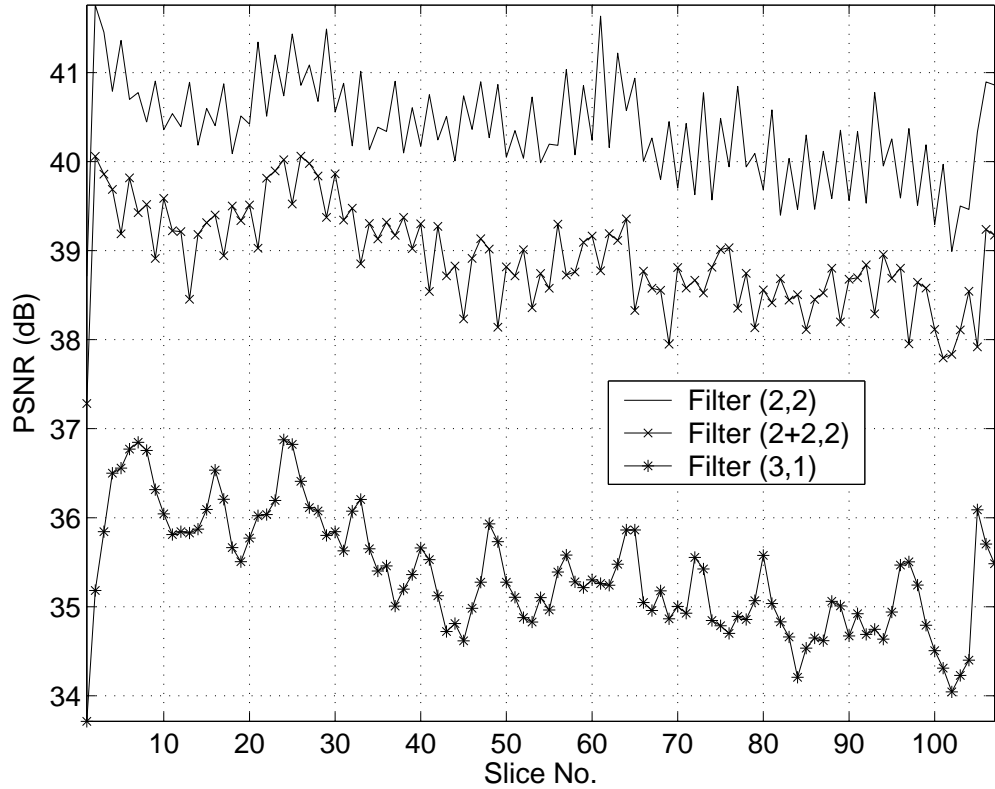


Figure 3.13: PSNR values of reconstructed slices with different wavelet filters decoded at 1 bit/voxel.

in the modeling process. Fig. 3.13 also shows the PSNR oscillations of filters (2,2) and (3,1) are more noticeable than those of filter (2+2,2), which verifies the results in Table 3.2. Therefore, it is reasonable to say that the proposed model can be applied on analyzing the performance of wavelet filters. To get high performance in lossy compression, one needs to use short filters with small coefficients. To reduce the PSNR fluctuations, filters with similar coefficient scales in lifting and dual lifting steps should be considered. Furthermore, other techniques like that in [42] can be designed according to the analysis in this thesis to get better performance.

### 3.5 Summary

In this chapter, a motion compensated lossy-to-lossless 4D medical image compression scheme is presented. Both temporal and spatial redundancies inside 4D images are exploited to achieve higher compression performance. For medical image

compression, lossy compression is usually not acceptable for diagnostic purposes. Our proposed compression scheme is progressively lossy-to-lossless, which offers the flexibility of using any truncated version of the embedded bit stream to decode a lossy version for preview and utilizing the entire bit stream to generate the lossless version whenever needed. Doctors may use the lossy mode in applications like telemedicine for fast searching and browsing of 4D medical data and the lossless mode for diagnosis. The compression performance under both lossless and lossy compression is evaluated and compared with other state-of-the-art techniques. PSNR fluctuations are observed when evaluating the progressive compression performance of our scheme, which are caused by quantization in compression and inverse wavelet transform in reconstruction. The effect is then analyzed and quantified in a theoretical framework.

## Chapter 4

# Related Works: Medical Image Analysis

In this chapter, a brief survey of various medical image analysis works is presented, starting with an introduction to medical image segmentation techniques in Section 4.1. In Section 4.2, we provide some background about parametric deformable models, which have been successfully used in medical image segmentation. Geometric deformable models and minimal path deformable models are introduced in Section 4.3 and 4.4, respectively. In Section 4.5, we present an overview of medical image visualization techniques.

### 4.1 Introduction

Image segmentation has been an important research topic for some decades and a number of methods have been proposed. Image segmentation is essentially a process of pixel/voxel classification, wherein the image pixels/voxels are segmented into subsets by assigning the individual pixels to classes.

Methods for performing medical image segmentation vary widely depending on the specific application, imaging modality and other factors. For example, the requirements for segmentation of brain tissue are different from those for segmentation of the liver. General imaging artifacts such as noise, partial volume effects, and motion can also have significant consequences on the performance of segmentation algorithms. Furthermore, each imaging modality has its own characteristics and could produce images, which are quite different looking for the same tissue. There is currently no single segmentation method that yields acceptable results for every kind of medical image. Methods do exist that are more general and can be applied to a variety of data [43–45]. However, those methods that are specific for particular applications can often achieve better performance by taking into account the specific nature of the image modalities.

Segmentation techniques are labeled as low level and high level techniques [43,46]. Low level segmentation techniques only consider local information such as edge, gradient and intensity. Edge detection [47], region growing [48], and watersheds [49] are among the most widely used low level techniques. In edge detection methods, an edge detector is first applied to an image to identify boundary elements by detecting intensity discontinuities. Then, an edge linking algorithm is used to link the boundary elements together to obtain a parameterized curve or surface representation. Region growing is a region-based technique that usually starts with a set of seed points and grows regions starting from these points by merging neighboring pixels or voxels that share similar properties. The concept of watersheds is based on visualizing a 2D image in three dimensions: two spatial coordinates versus gray levels. Watershed based algorithms suppose that a hole is punched in each regional minimum and that the entire topology is flooded from below by letting water rise through the holes at a uniform rate. When the rising water in distinct catchment basins is about to merge, a dam is built to prevent the merging. The flooding will eventually reach a stage when only the tops of the dams are visible above the water line. These dams are the boundaries extracted

by a watershed segmentation algorithm. Further information about these classical boundary mapping methods can be found in most image processing and computer vision textbooks [50, 51].

One limitation of these low level methods is that they only consider local information, so that incorrect assumptions may be made during the boundary integration process and infeasible object boundaries could be generated. As a result, these methods usually require considerable amounts of expert intervention. In addition, these methods usually generate results that are constrained by the resolution of the images and do not necessarily lead to accurate results. To address these problems, *a priori* knowledge, like the approximate shape, size or location of the object in images, may be incorporated into segmentation process and this leads to high level techniques. An impressive example of these advanced methods is the *deformable models* [43, 52].

In recent years, segmentation techniques that combine deformable models with local edge extraction have achieved considerable success in medical image segmentation [43, 44]. Deformable models are capable of accommodating the often significant variability of biological structures. Furthermore, different regularizers can be easily incorporated into deformable models to get better segmentation results for specific types of image. Deformable models [43] are referred by different names in the literature. In 2D segmentation, deformable models are usually referred as snakes [46, 52], active contours [53, 54], balloons [55], and deformable contours [56]. They are usually referred as active surfaces [57] and deformable surfaces [58, 59] in 3D segmentation.

Deformable models were first introduced into computer vision by Kass *et al.* [52] as “snakes” or active contours, and are now more well known as *parametric deformable models* due to their explicit representation as parameterized contours in a Lagrangian framework. By designing a global shape model, boundary gaps are easily bridged, and overall consistency is more likely to be achieved. Parametric

deformable models are commonly used when some prior information of the geometrical shape is available, which can be encoded using preferably, a small number of parameters. They have been used extensively but their main drawback is the inability to adapt to topology [43, 46]. *Geometric deformable models* are represented implicitly as a level set of a higher-dimensional, scalar level set functions and evolve in an Eulerian fashion [60, 61]. Geometric deformable models were introduced more recently by Caselles *et al.* [62] and by Malladi *et al.* [63]. A major advantage of geometric deformable models over parametric deformable models is topological flexibility due to their implicit representation. During the past decade, tremendous efforts have been put into various medical image segmentation applications based on level set methods [64]. Many new algorithms have been reported to increase the precision and robustness of level set methods. For example, Chan and Vese [65] proposed an active contour model that can detect objects whose boundaries are not necessarily defined by gray level gradients.

Although these existing methods are very powerful in solving object segmentation problems, they have some drawbacks also. For instance, deliberate initialization is required by most existing methods when used for segmentation. Currently, manual initialization is the most commonly used method for deformable models. This becomes impractical when dealing with large numbers of medical images. In addition, initialization may be tedious when dealing with several adjacent objects. Another main drawback is that current deformable models are computationally complex. They involve a large number of iterations before achieving an optimal result.

Cohen *et al.* [66] address the initialization problem. Their *minimal path deformable model* requires only two end points and the final contour will be between these two points. The work is further extended in [67] by incorporating a *fast marching* method [61]. Han *et al.* [68] developed a minimal path finding algorithm based on [66]. It eliminates the “metrication errors” by a “wriggling” process. How-

ever, the algorithm easily gets into local minima during the “wriggling” process. Details about the deformable models are presented in the following sections.

## 4.2 Parametric Deformable Models

The parametric deformable models were first introduced by Kass *et al.* [52]. In their formulations, image segmentation is posed as an energy minimization problem. The energy functional is minimized when the contours are smooth and reside on the object boundaries.

In the original parametric deformable models [52], the energy is defined as

$$E(\mathcal{C}) = \int_0^1 \left( \alpha \left| \frac{\partial \mathcal{C}}{\partial s} \right|^2 + \beta \left| \frac{\partial^2 \mathcal{C}}{\partial s^2} \right|^2 - f(\mathcal{C}(s)) \right) ds \quad (4.1)$$

where  $\mathcal{C}(s)$  is a curve and parameterized by  $s \in [0, 1]$ . Here,  $\alpha$  and  $\beta$  are real positive constants, and  $f$  is a driven force derived from the image. The object contour is approached by  $\mathcal{C}(s)$  when the associated energy  $E(\mathcal{C})$  is minimized. The first two terms at the right side of (4.1) define the internal deformation energy, which characterizes the deformation of a stretchy, flexible contour (or surface in 3D). Parameter  $\alpha$  controls the ‘tension’ of the contour while  $\beta$  controls its ‘rigidity’. The last term  $f(\mathcal{C}(s))$  at the right side of (4.1) defines the external force, which is a function derived from the image so that it takes on its smaller values at the features of interest, such as boundaries.

In the above model, the external image force (last term of (4.1)) is significant only in the immediate vicinity of the object boundaries. Elsewhere, the model’s evolution is driven dominantly by the internal forces (the first two terms of (4.1)), which leads to shrinking and smoothing of the contour. As a result, the original model requires an initial guess close to the actual object boundaries, or at least

located outside of the object, so that the shrinking forces move the contours close to the boundaries where the external force dominates.

The problem of finding a parameterized contour  $\mathcal{C}(s)$  that minimizes energy  $E(\mathcal{C})$  is known as a variational problem. In accordance with the calculus of variation, the contour  $\mathcal{C}(s)$  which minimizes the energy functional  $E(\mathcal{C})$  satisfies the following Euler-Lagrange equation

$$\frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathcal{C}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathcal{C}}{\partial s^2} \right) + \nabla f(\mathcal{C}(s)) = 0 \quad (4.2)$$

This vector valued partial differential equation expresses the balance of internal and external forces when the contour rests at equilibrium. In practice, the minimization is performed by considering the curve as a function  $\mathcal{C}(s, t)$  of time  $t$  as well as  $s$ . We note that adding a time directive term of  $\mathcal{C}$  is equivalent to applying gradient descent algorithm to find the local minimum of Eqn. (4.1) [58]. Then, the partial derivative of  $\mathcal{C}$  with respect to  $t$  is then set equal to the left hand side of (4.2) as follows

$$\vec{\mathcal{C}}_t = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathcal{C}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathcal{C}}{\partial s^2} \right) + \nabla f(\mathcal{C}(s)). \quad (4.3)$$

When the solution  $\mathcal{C}(s, t)$  stabilizes, the term  $\vec{\mathcal{C}}_t(s, t)$  vanishes and we achieve a solution of Eqn. (4.2). A numerical solution to (4.3) can be found by discretizing the equation and solving the discrete system iteratively [52].

### 4.3 Geometric Deformable Models

Geometric deformable models are based on the theory of *front evolution* and are implemented using the level set numerical method [60, 61]. In this section, we briefly review the main theory and major results of geometric deformable models.



### 4.3.1 Front Evolution Theory

Let  $\mathcal{C}(q, t)$ , defined as  $\{x(q, t), y(q, t)\}$  in 2D and  $\{x(q, t), y(q, t), z(q, t)\}$  in 3D, denote a family of closed contours generated by evolving an initial contour  $\mathcal{C}_0(q) = \mathcal{C}(q, 0)$ , where  $t$  parameterizes the family and  $q$  parameterizes the given contour. The basic result from the front evolution theory is that the geometric shape of the contour is determined by the normal component of the evolution velocity, while the tangential component affects only the parameterization. Hence, the evolution equation can be written as

$$\vec{\mathcal{C}}_t(q, t) = F(\mathcal{C}(q, t))\vec{\mathcal{N}}(\mathcal{C}(q, t)) \quad (4.4)$$

with

$$\mathcal{C}(q, 0) = \mathcal{C}_0(q)$$

where  $F(\mathcal{C}(q, t))$  is a scalar function that often depends on the curvature  $\kappa$  of the contour (for surfaces, both mean and Gaussian curvatures can be used), and  $\vec{\mathcal{N}}(\mathcal{C}(q, t))$  is the unit normal vector (chosen to be either the inward or the outward normal) along the contour  $\mathcal{C}(q, t)$ .

The Lagrangian approach to the above evolution equation involves discretizing the contour into a set of elements (*e.g.*, nodes connected by lines or triangles) and updating the node positions using a numerical approximation to (4.4) as the parametric deformable models in Section 4.2. In such an approach, frequent adjustment of the node spacing is required in order to preserve data fidelity and reduce numerical approximation errors. Computationally complex approaches may also be required for self-intersection avoidance. Level set methods are then proposed to solve this problem in another way [60, 61].

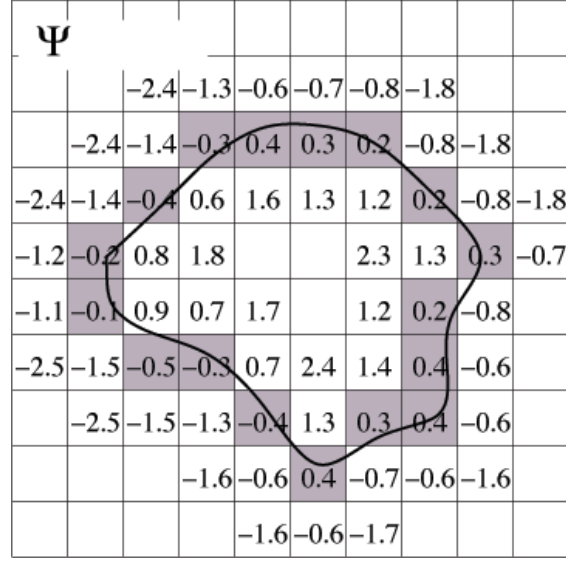


Figure 4.1: The implicit level set curve is the black line superimposed over the image grid. The location of the curve is interpolated by the pixel values of a signed distance map. The grid pixels closest to the implicit curve are shown in gray.

### 4.3.2 Level Set Methods

Level set methods are numerical techniques for analyzing and computing interface motion [60, 61, 63]. Since level set methods use implicit representation for the evolving curves, they allow for topological changes to occur without any additional computational complexity. When solving a problem, the level set method increases the dimensionality of the problem to a higher one. For example in curve evolution problems, instead of evolving the 2D curves, the method evolves a 3D surface and the 2D objective curve is embedded as its zero level set. Then different topologies of the 2D curve can be represented by a constant topology of the 3D surface.

The level set technique developed by Osher and Sethian [60] represents the contour  $\mathcal{C}(q, t)$  implicitly as the zero level set of a smooth, Lipschitz-continuous scalar function  $\Psi(x, t)$ , as shown in Fig. 4.1. Function  $\Psi(x, t)$  is also known as the *level set function*, where  $x \in \mathbb{R}^2$  in 2D and  $x \in \mathbb{R}^3$  in 3D. The implicit contour at any time  $t$  is given by

$$\mathcal{C}(\cdot, t) = \{x | \Psi(x, t) = 0\}. \quad (4.5)$$

Although there are many choices of the level set function, in practice, the signed distance function is preferred for its stability in numerical computations (see Fig. 4.1). The fast marching method proposed in [69] provides an efficient algorithm for constructing the signed distance function from a given initial contour. The initialization can also be as simple as one or several seed points. We use signed distance functions as the representation of function  $\Psi(x, t)$  for all of our experiments in this dissertation.

By differentiating  $\Psi(x, t) = 0$  with respect to  $t$  and substituting (4.4), the following associated equation of motion for the level set function  $\Psi(x, t)$  can be derived:

$$\frac{\partial \Psi(x, t)}{\partial t} = F(x, t) |\nabla \Psi(x, t)| \quad (4.6)$$

with

$$\Psi(\mathcal{C}_0(p), 0) = 0,$$

where  $|\nabla \Psi|$  denotes the norm of the gradient of  $\Psi$ . Above equations illustrate that evolving a curve  $\mathcal{C}$  is equivalent to updating its signed distance function  $\Psi$  according to speed function  $F$ . The evolving curve  $\mathcal{C}$  is then obtained as the zero level set of  $\Psi$ . The formulation is analogous for the case of surface evolving in 3D.

### 4.3.3 Geometric Deformable Models

Caselles *et al.* [62] and Malladi *et al.* [63] applied the above theory to the problem of image segmentation by multiplying the contour velocity by a “stopping” term  $g(|\nabla I(x)|)$  that is a monotonically decreasing function of the gradient magnitude of the image  $I$  (or its smoothed version). In this way, they arrived at the following evolution equation

$$\frac{\partial \Psi(x, t)}{\partial t} = g(|\nabla I(x)|)(c + \kappa(x, t)) |\nabla \Psi(x, t)|, \quad (4.7)$$

where  $c$  is a constant inflation or deflation (depending on its sign) speed term, similar to the function of balloon force in [55], which aims to keep the contour moving in the proper direction, and  $\kappa(x, t)$  is the mean curvature of the level set of  $\Psi(x, t)$  that passess through the point  $x$ , which can be easily computed from the spatial derivatives of  $\Psi(\cdot, t)$  (see [61]). We note that, in [62, 63], the above formulation is originally derived for planar curves, however, the same form applies to surfaces as well. In the remainder of this thesis, all of the equations apply to both curves and surfaces unless stated otherwise.

However, the model described in Eqn. (4.7) does not arise from the minimization of an energy function as in parametric deformable models. Thus, regularization of smoothness and rigidity might not be assured. To address this, Caselles *et al.* [53, 70] and Kichenassamy *et al.* [54] derived another geometric deformable model, called the *geodesic active contour* (GAC) model. The basic idea is to consider the object boundary detection as a problem of geodesic computation in a Riemannian space, according to a metric  $g(x)$  induced by the given image  $I$ .

The GAC model can be formally described in 2D as follows. The task of finding a curve that best fits the object boundary can be posed as a minimization problem over all closed planar curves  $\mathcal{C}(q) : [0, 1] \rightarrow \mathbb{R}^2$  [53, 54, 62]. Starting from the parametric deformable model first introduced by Kass *et al.* [52], in which the curve  $\mathcal{C}$  associated with an energy given by Eqn. (4.1) Caselles *et al.* [53] obtained a minimal path representation of the energy. The objective function is

$$E(\mathcal{C}) = \int_0^1 g(|\nabla I(\mathcal{C}(q))|) \left| \frac{\partial \mathcal{C}(q)}{\partial q} \right| dq \quad (4.8)$$

where  $g : [0, \infty] \rightarrow \mathbb{R}^+$  is a strictly decreasing function such that  $g(r) \rightarrow 0$  as  $r \rightarrow \infty$ . To minimize this objective function by steepest descent, consider  $\mathcal{C}$  to be a function of time  $t$  as well as spatial parameter  $q$ . The Euler-Lagrange equations

yield the curve evolution equation

$$\vec{C}_t = g(I)\kappa\vec{N} - (\nabla g \cdot \vec{N})\vec{N} \quad (4.9)$$

where  $\kappa$  is the Euclidean curvature, and  $\vec{N}$  is the unit normal vector of the contour. Theoretically, function  $g$  can be any uniform decreasing function, but the performance may not be optimal for some choices since the local minimization is obtained using this method.

This GAC model can be readily cast within the level set framework. This yields an equivalent contour evolution process implemented using the following level set function evolution equation

$$\frac{\partial \Psi(x, t)}{\partial t} = g(x)\kappa(x, t)|\nabla \Psi(x, t)| + \nabla g(x) \cdot \nabla \Psi(x, t). \quad (4.10)$$

The extension of GAC to 3D is straightforward and has been done in [70], where 3D object segmentation is realized through finding the minimal surface.

There are many other extensions of the basic geometric deformable model (*e.g.*, [71, 72]), which were designed either to improve the overall performance of the original model or to adapt to particular applications. In this thesis, we consider a very general framework of geometric deformable models summarized by the following evolution equation [61]:

$$\begin{aligned} \frac{\partial \Psi(x, t)}{\partial t} = & F_{prop}(x, t)|\nabla \Psi(x, t)| + F_{curv}(\kappa(x, t))|\nabla \Psi(x, t)| \\ & + \vec{F}_{adv}(x, t) \cdot \nabla \Psi(x, t), \end{aligned} \quad (4.11)$$

where  $F_{prop}(x, t)|\nabla \Psi(x, t)|$  is an expansion or contraction force or speed (“force” and “speed” are used interchangeably in literature);  $F_{curv}(\kappa(x, t))|\nabla \Psi(x, t)|$  is the part of the force that depends on the intrinsic geometry, *i.e.* the mean curvature  $\kappa(x, t)$ ; and  $\vec{F}_{adv}(x, t) \cdot \nabla \Psi(x, t)$  is an advection force that passively transports

the contour. The right-hand side of (4.11) can arise from the gradient descent minimization of an energy function as in the GAC model (4.10), where  $F_{prop}(x, t) = cg(x)$ ,  $F_{curv}(\kappa(x, t)) = \kappa(x, t)g(x)$ , and  $\vec{F}_{adv}(x, t) = \nabla g(x)$ .

## 4.4 Minimal Path Deformable Models

The derivation of minimal path deformable models is similar to that of GAC. Actually, minimal path deformable models can be considered as a branch of GAC but using different approaches for implementation. Since they are increasingly investigated and developed in medical image segmentation, they are discussed separately here.

The derivation of the minimal path deformable models is as follows. In the original parametric deformable model [52], the energy functional  $E(\mathcal{C})$  is defined as

$$E(\mathcal{C}) = \int_0^1 \left( \alpha \left| \frac{\partial \mathcal{C}}{\partial s} \right|^2 + \beta \left| \frac{\partial^2 \mathcal{C}}{\partial s^2} \right|^2 - f(\mathcal{C}(s)) \right) ds \quad (4.12)$$

where  $\mathcal{C}(s)$  is a curve and parameterized by  $s \in [0, 1]$ . Here,  $\alpha$  and  $\beta$  are real positive constants, and  $f$  is an edge map of the image. Solving the problem in (4.12) amounts to finding, for a given set  $\alpha$ ,  $\beta$  and  $\gamma$ , the curve  $\mathcal{C}$  that minimizes the energy functional  $E(\mathcal{C})$ . A problem with the original deformable models is the need to select parameters that control the trade-off between smoothness and proximity to the object. Inappropriate parameters can make the minimization process difficult or even impossible. Caselles *et al.* [53] proved that curve smoothing is obtained by setting  $\beta = 0$ , which reduces (4.12) to

$$E(\mathcal{C}) = \alpha \int_0^1 \left| \frac{\partial \mathcal{C}}{\partial s} \right|^2 - \gamma \int_0^1 f(\mathcal{C}(s)) ds. \quad (4.13)$$

As we will see later, the regularization effect on the active contours comes from curvature based curve flows, which can be obtained only from the other terms in

Eqn. (4.12). This will allow to achieve smooth curves in the proposed approach without having the high order smoothness given by  $\beta \neq 0$  in energy-based approaches. Moreover, the second order smoothness component in (4.12), assuming an arc-length parametrization, appears in order to minimize the total squared curvature. The use of curvature driven curve motions as smoothing term was proved to be very efficient in previous literature [53, 63]. Therefore, curve smoothing will be obtained also with  $\beta = 0$ , having only the first regularization term.

If  $g: [0, +\infty[ \rightarrow \mathbb{R}^+$  denotes a strictly decreasing function such that  $g(r) \rightarrow 0$  as  $r \rightarrow +\infty$ , Eqn. (4.13) can be changed into a general energy functional by replacing the edge map  $f$  with  $g(|\nabla I|)^2$ :

$$E(\mathcal{C}) = \alpha \int_{\Omega} \left| \frac{\partial \mathcal{C}}{\partial s} \right|^2 ds + \gamma \int_{\Omega} g(|\nabla I(\mathcal{C})|)^2 ds \quad (4.14)$$

where  $\nabla I$  is gradient of the image  $I : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$ . Minimizing (4.14) is equivalent to a problem of geodesic computation in a Riemannian space [53]:

$$E(\mathcal{C}) = \int_{\Omega} g(|\nabla I(\mathcal{C})|) \left| \frac{\partial \mathcal{C}}{\partial s} \right| ds \quad (4.15)$$

Noting that the Euclidean length of the contour  $\mathcal{C}$  is given by

$$L(\mathcal{C}) = \int_{\Omega} |\mathcal{C}_s| ds \quad (4.16)$$

and the object contour is approached by curve  $\mathcal{C}$  so the energy functional  $E(\mathcal{C})$  is minimized. Therefore, the problem of image segmentation is transformed into a search for the global minimal path weighted by  $g(|\nabla I(\mathcal{C}(s))|)$ . Thus, object contours can be delineated by evaluating the minimal paths instead of minimizing the energy  $E(\mathcal{C})$  in (4.12). Methods based on this approach are known as *minimal path deformable models*, which have lower computational complexity in high order gradients and do not involve minimizing the corresponding Euler-Lagrange equation.

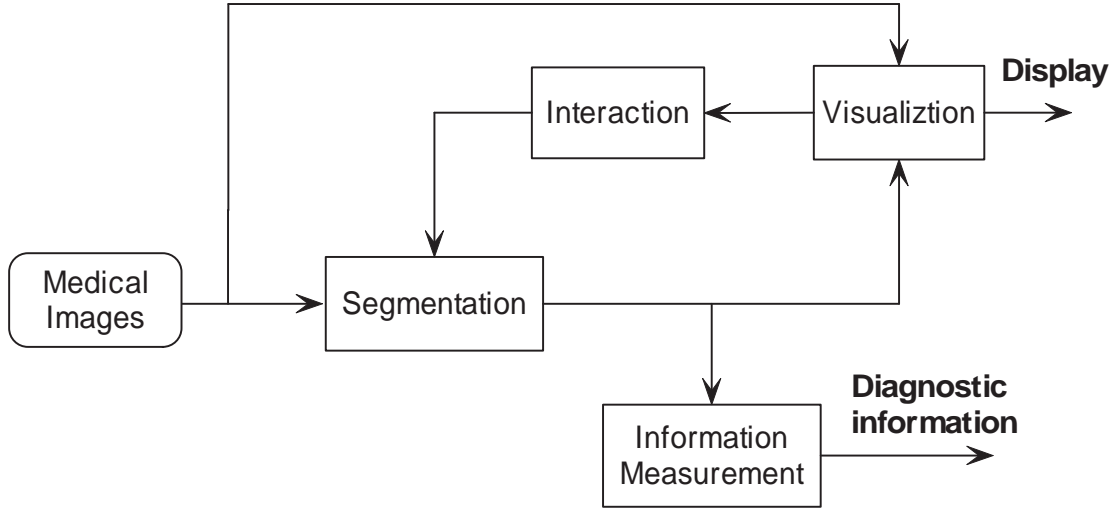


Figure 4.2: Block diagram of medical image analysis scheme incorporated with visualization.

## 4.5 Medical Image Visualization

To facilitate diagnosis, 3D or 4D medical images and their corresponding segmentation results are usually displayed using computer graphics techniques with sufficient ability for interaction. *Visualization* is the technique of exploring, transforming and viewing data as 2D images to gain understanding and insight into the data. Since 4D images can be visualized as a sequence of 3D images, the focus here is on 3D volumetric image visualization. Visualization techniques have characteristically been divided into two different types: *volume rendering* and *surface rendering* [73]. Both techniques can produce visualization of 3D volumetric images, but the methods involved in these techniques are different and each has its own pros and cons. A medical image analysis scheme integrated with visualization is shown in Fig. 4.2.

### 4.5.1 Volume Rendering

Volume rendering techniques based on ray-casting algorithms have generally become the method of choice for visualization of volumetric images. These methods



provide direct visualization of the volumetric images without the need for prior surface or object segmentation. They preserve the values and context of the original image data as well. Volume rendering techniques allow for the application of various rendering algorithms during the ray-casting process. Surface extraction is not necessary as the entire volume image is used in this rendering process. This provides the capability to visualize the actual data in the volumetric images and to make voxel value based measurements for the rendered image. The rendered image can be dynamically determined by changing the ray-casting conditions during the rendering process.

However, the volume of 3D medical image data sets is normally characteristically large. This leads to significant computational complexities of volume rendering techniques, which places high demands on the implementation systems. This is particularly true when the rendering process must provide sufficient details for visualizing some structures. Furthermore, given the discrete voxel-based nature of the volumetric images, there is no direct connection to other geometric objects, which may be desired for inclusion in the rendering or for output of the rendered structure to other devices.

### **4.5.2 Surface Rendering**

Surface rendering techniques inherently require the extraction of object contours that define the surface of the structure to be visualized. An algorithm then places surface patches or tiles at each contour point. The surface is rendered visible after removing hidden surface and shading according to light sources. The advantage of this technique lies in the relatively small amount of contour data, resulting in fast rendering speeds. In addition, standard computer graphics techniques can be applied, such as shading model. The technique can take advantage of particular graphics hardware to speed up the geometric transformation and the rendering

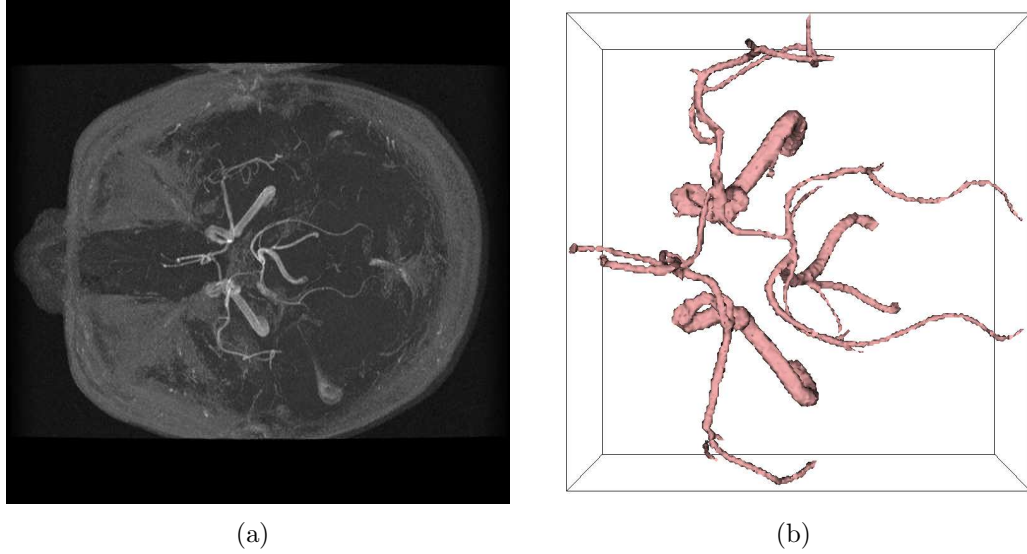


Figure 4.3: Samples of visualization results generated by using (a) volume rendering technique and (b) surface rendering technique.

processes [73]. Furthermore, other analytically defined structures can be easily superposed with the rendered structures.

The disadvantages of this technique are largely based on the need to discretely extract the surfaces defining the structures to be visualized. The segmentation can be a difficult task as highlighted in previous sections. After the segmentation is done, only isosurfaces are kept and visualized, while other information is not presented in the rendering process. This prohibits any further interactive and dynamic determination of the surface to be rendered, as the decision has been made during contour extraction, *i.e.* segmentation. Finally, the discrete nature of the surface patching technique makes it prone to sampling and aliasing artifacts on the rendered surface.

### 4.5.3 Applications

Since both techniques, volume rendering and surface rendering, can produce visualization of volumetric image, selection between these two approaches is often predicated on the particular nature of the data, the application to which the visu-

alization is being applied, and the desired result of the visualization. In this work, volume rendering is used to directly visualize the original data because no prior processing is required. For example, *maximum intensity projection* (MIP) [74] is employed to display MRA data sets instead of displaying each 2D slice of the data as shown in Fig. 4.3(a). The MIP is a kind of volume rendering technique but not a segmentation technique, although vasculature is observable in it. When doing the MIP, the ray-casting simply picks up the maximum value encountered without any prior segmentation. According to its nature of requiring extracted object contours, surface rendering is selected for visualizing segmentation results. Fig. 4.3(b) shows the extracted vasculature from the same MRA data set as in Fig. 4.3(a), which is visualized using surface rendering technique.

# Chapter 5

## Minimal Path Deformable Models

### 5.1 Introduction

In this chapter, we present a new minimal path deformable model incorporated with prior shape knowledge for medical image segmentation. Object boundaries are delineated by detecting a minimal path, *i.e.*, a path with the minimal energy. An intelligent “worm” [16] is used to extract object boundaries under the influence of shape priors.

When segmenting or localizing an anatomical structure, prior knowledge is usually very helpful. The incorporation of specific prior information into deformable models has received great attention. Several methods of incorporating prior shape information into boundary determination have been developed. Cootes *et al.* [75] propose an *active shape model* to construct a statistical shape model from a set of training images for image segmentation. The model is built by using the point positions after finding point correspondences across shapes. It is further extended to consider texture information for image segmentation in [76]. Staib and Duncan [77] incorporate global shape information into the segmentation process by using an elliptic Fourier decomposition of the boundary and placing a Gaussian prior on the

Fourier coefficients. In [78], Leventon *et al.* incorporate statistical shape influence into the evolution process of geodesic active contours [53] by embedding each shape in the training dataset as the zero level set of level set map. Chen *et al.* [79] proposed a variational method that minimizes an energy functional depending on the information of the image gradient and the shape of interest. In [80], Rousson *et al.* further derive an implicit representation to incorporate prior knowledge into the segmentation process. More recently, Xie *et al.* [81] utilized both texture and shape priors when defining their energy functional and segmentation is achieved through minimizing the functional.

In our algorithm, an image is considered as a regular graph, which enables the use of the great power of graph based segmentation methods [82]. To evaluate the paths, a weighted graph is designed. The shape priors are represented implicitly by the zero level sets of their signed distance maps to guide the segmentation. Since the implicit representation is in a very similar form as the weighted graph of the image, the prior shape knowledge is naturally incorporated into the segmentation process as the other weighted map. To avoid being captured by local minima, our algorithm uses an intelligent “worm” to look for the minimal path. In general, existing minimal path finding algorithms need two or more initialization points, whereas our algorithm requires only a single starting point to get a contour. Furthermore, the computational complexity is significantly reduced.

## 5.2 Finding the Minimal Path

The shortest path finding problems are well studied and many excellent algorithms, like Dijkstra’s method and dynamic programming [83], have been proposed. These graph searching algorithms can be used for image segmentation, but suffer from *metrication errors* (see Fig. 5.1) for several reasons [66]. If pixels are thought of as nodes, a raster image can be considered as a regular graph with unit weights on

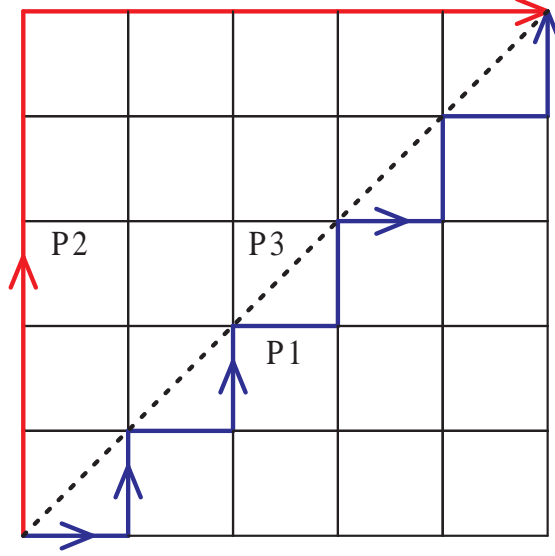


Figure 5.1: Illustration of metrication error.

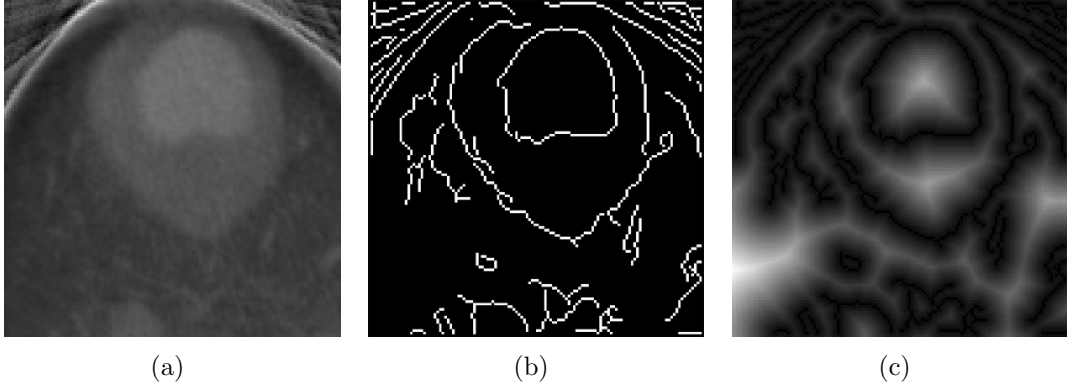


Figure 5.2: (a) A sample of CT cardiac image. (b) Edge detection result of the image. (c) Graph weighting map produced by applying distance transform on the edge map.

every link. In addition, the distance between two points is restricted to be a *city block* distance in graph searching algorithms. Under such an assumption, both P1 and P2 are shortest paths in Fig. 5.1. However, by refining the graph grids, only P1 can approach the ideal shortest path P3 based on Euclidean distances.

To solve this problem, raster images should be transformed into some properly weighted graphs. Since our objective is to extract organ contours, which are normally defined by edges, we expect that the graph nodes have lower weights when they are near the edges and higher weights when far away. Thus, the Euclidean distance transform [84] is applied on the edge map of the image to assign weights

to the graphs (see Fig. 5.2). For each pixel in the transformed map, the Euclidean distance transform assigns a number that is the distance between that pixel and the nearest nonzero pixel of the edge map. In 2D, the Euclidean distance between points  $(x_1, y_1)$  and  $(x_2, y_2)$  is:

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

With this measure, if the graph in Fig. 5.1 is weighted as the Euclidean distance map to the starting point, paths like P2 can be excluded by the shortest path finding algorithms while paths like P1 will be kept. Therefore, with refinement of grids, the shortest paths under Euclidean distance measure will be the expected result.

### 5.2.1 Implicit Prior Shape Modeling

Prior shape knowledge is very useful when segmenting organs from medical images. To incorporate this prior information into the segmentation process, we consider a probabilistic approach, and compute a prior on shape variation with a set of given training instances. To build the shape model, we choose a representation of shapes, and then define a probability density function over the parameters of the representation.

Suppose that we have a training set  $\mathbf{C} = \{\mathcal{C}_i \mid i = 1, 2, \dots, n\}$  of  $n$  registered shapes. To avoid the point correspondence problem in [75], an implicit representation of shapes is desired. Given the selected optimization framework in our algorithm, it will be convenient to add this shape prior information into the segmentation process if the shapes can be represented as some form of Euclidean distance. One way is to represent a curve  $\mathcal{C}$  by its Euclidean distance transform map  $\Phi : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$ , where  $\mathcal{C} = \{(x, y) \mid \Phi(x, y) = 0\}$ . However, the prior shape model is difficult to derive under this representation due to the discontinuity

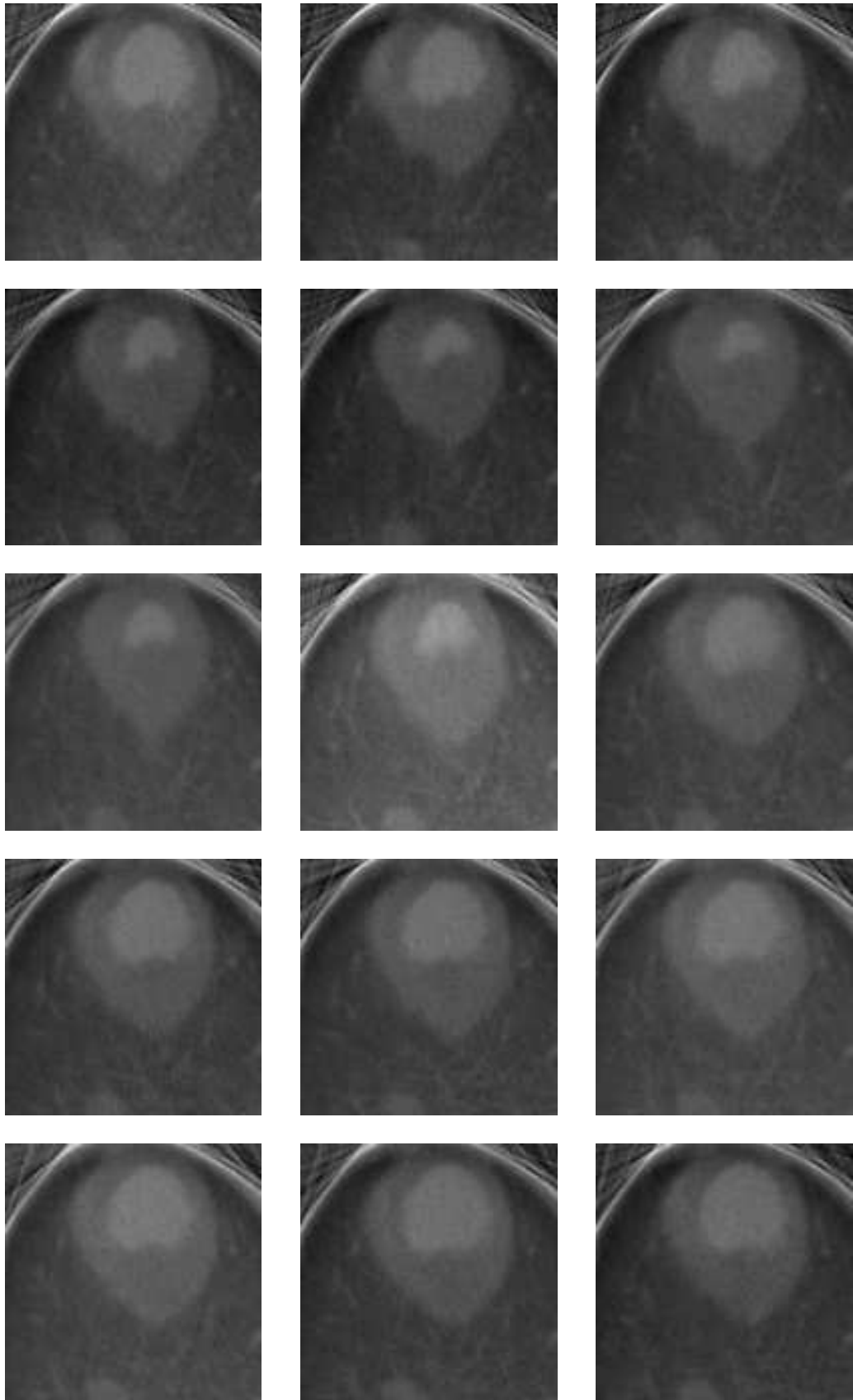


Figure 5.3: Samples of CT cardiac image over a cardiac cycle [1].



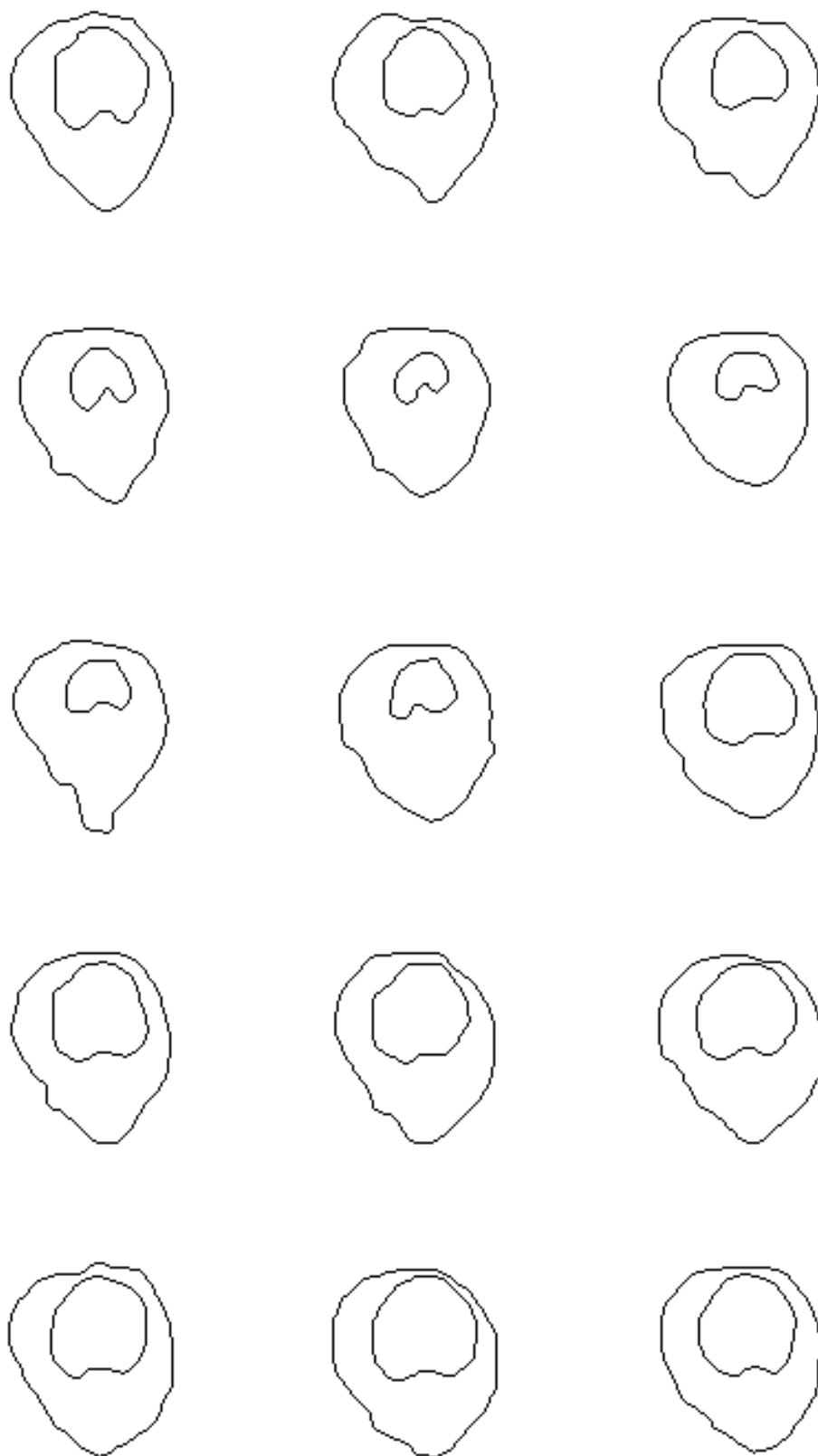


Figure 5.4: Manual segmentation results of images shown in Fig. 5.3.

of function  $\Phi$  on the curves. To overcome this problem, a curve  $\mathcal{C}$  is represented by its signed Euclidean distance transform map  $\Psi : [0, a] \times [0, b] \rightarrow \mathbb{R}$ , where  $\mathcal{C} = \{(x, y) \mid \Psi(x, y) = 0\}$  and  $\Psi$  has negative values inside  $\mathcal{C}$  and positive values outside  $\mathcal{C}$ . Following this, our shape representation is very similar to the level set representation in [78] and has numerous advantages [64].

When building the prior shape model, each curve  $\mathcal{C}_i$  in the training set is represented implicitly by the zero level set of its signed distance map  $\Psi_i$ . The mean and variance of the training data can be computed using *principal component analysis* (PCA) [78]. Subtracting the mean shape,  $\bar{\Psi} = \frac{1}{n} \sum_{i=1}^n \Psi_i$ , from each  $\Psi_i$  and reshaping the differences result in column vectors in a 2D matrix  $\mathbf{P}$  with size of  $(a \cdot b) \times n$ . Using the *singular value decomposition* (SVD), the matrix  $\mathbf{P}$  is decomposed as  $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ .  $\mathbf{U}$  is a matrix with orthogonal column vectors which consist of the modes of shape variation and  $\mathbf{\Sigma}$  is a diagonal matrix composed of corresponding singular values, *i.e.*, mode amplitudes. An estimate of the object shape can be represented by  $k$  principal components and a  $k$  dimensional vector of shape parameters  $\mathbf{b}$  (where  $k < n$ ) [78] as

$$\hat{\Psi} = \mathbf{U}_k \mathbf{b} + \bar{\Psi}. \quad (5.1)$$

Under the assumption of a Gaussian distribution of shape represented by  $\mathbf{b}$ , we can compute the probability of a certain curve as

$$p(\mathbf{b}) = \frac{1}{\sqrt{(2\pi)^k |\mathbf{\Sigma}_k|}} \exp \left( -\frac{1}{2} \mathbf{b}^T \mathbf{\Sigma}_k^{-1} \mathbf{b} \right) \quad (5.2)$$

where  $\mathbf{\Sigma}_k$  contains the first  $k$  rows and  $k$  columns of  $\mathbf{\Sigma}$ .

Fig. 5.3 shows a sequence of *computed tomography* (CT) cardiac images over a cardiac cycle [1]. These fifteen images are used to construct the training set. The corresponding manual segmentation results are shown in Fig. 5.4, based on which the prior shape model is built. The generated shapes with varying modes

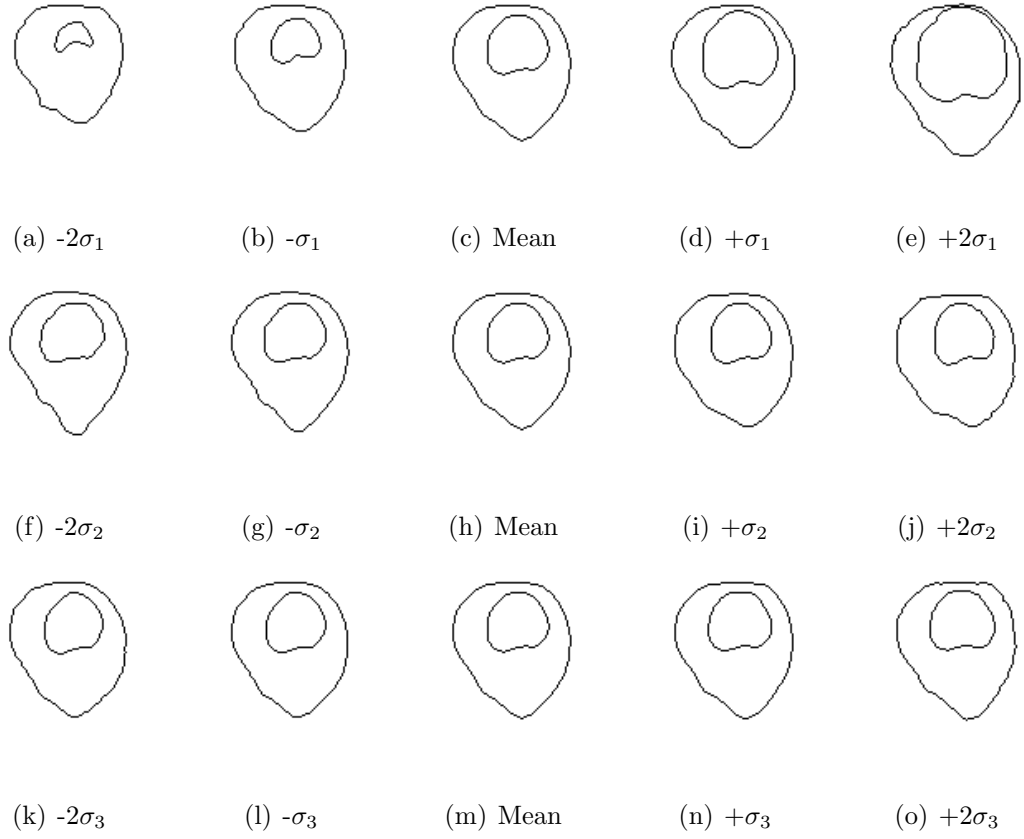


Figure 5.5: Extracted zero level set of the largest three modes of variation.

are shown in Fig. 5.5 by extracting the zero level sets from the generated maps. The mean shape and primary modes appear to be reasonable representations of the samples being learned. In Fig. 5.5, varying the first mode significantly changes the size of generated shape, while the second and third modes have a rather small influence on the shapes.

### 5.2.2 Worm Algorithm

Given that the potential energy map of the image has lower values near the edges or features, segmentation can be done by looking for a minimal path on it, as stated in Section 4.4. Our algorithm finds the minimal path from a single initial point using an extended version of our worm algorithm [16].

The worm for minimal path detection is composed of a sequence of points and

has a length  $L_w$  and energy  $E_w$ . The edges obtained along an object contour may not be continuous. So, the worm needs to be long enough to skip disjointed parts. As the worm needs to avoid local minima, its energy consists of both the *external energy*, which is based on the influence of image features and the prior shape model, and *internal energy* generated by its own topology. In this dissertation, only edges are used as the image features for simplicity although more complicated features can be used to further improve the performance. At each step, the worm compares all the possible ways ahead and moves itself along the one with minimal energy.

Let  $F_{shape}$ ,  $F_{image}$  and  $F_{int}$  denote the prior shape attraction force, the image feature attraction force and the internal force acting on the worm, respectively. The force  $F_{shape}$  represents the influence from the organ shape estimate produced by the maximum *a posteriori* (MAP) prior shape estimator, which is described in Section 5.2.3.  $F_{image}$  describes the force characterized by the weighted graph of the image, while  $F_{int}$  tries to stretch the worm body and keeps it from twisting. The overall energy  $E_w$  of the worm is then defined as:

$$E_w = \int_{\Omega} (\alpha F_{shape}(\omega) + \beta F_{image}(\omega) + F_{int}(\omega)) d\omega, \quad (5.3)$$

where  $d\omega$  is a small part of the worm body,  $\Omega$  denotes the extent of the worm body, and  $\alpha$  and  $\beta$  are positive real numbers that balance the forces.

In our proposed algorithm,  $F_{image}$  is the force that attracts the worm to the edges of the image. It is stronger when the worm is further away from the edges. The edges are detected by Canny edge detector [47] and the distance transform [84] is applied on the edge map, which is used to represent  $F_{image}$  as shown in Fig. 5.2.

$F_{shape}$  is defined in a similar way as  $F_{image}$ . It is strong when the worm is far away from the prior shape contour and becomes weak when near, so that this force attracts the worm to the prior shape estimate of the organ according to (5.3). Since the estimated shape is embedded as the zero level set in its signed distance map  $\Psi$ ,

$F_{shape}$  is represented using  $|\Psi|$ , *i.e.*, the Euclidean distance transform of the shape estimate.

The internal force  $F_{int}$  is used to constrain the bending of the worm. This makes the worm robust to noise and prevents it from twisting. It is defined as

$$F_{int}(\omega) = \gamma(\omega) \left| \frac{\partial^2 \mathcal{C}(\omega)}{\partial s^2} \right|^2, \quad (5.4)$$

where  $\mathcal{C}(\omega)$  is the body of the worm. In addition,

$$\gamma(\omega) = \begin{cases} \gamma_1 & F_{image}(\omega) \leq 0.5, \\ \gamma_2 & F_{image}(\omega) > 0.5, \end{cases} \quad \text{where } \gamma_1 \ll \gamma_2. \quad (5.5)$$

In (5.5),  $F_{image} \leq 0.5$  when the worm is at or very near an edge. To help the worm fit itself into a complex object shape,  $F_{int}$  should be insignificant at edges corresponding to the object contour. In almost all our numerical calculations, we generally select  $\gamma_1 = 0.1$  and  $\gamma_2 = 1.0$ .

Since raster images can be deemed as graphs with rectangular grids, the problem is transformed to graph searching and *dynamic programming* [83] is employed to select the minimal path. Dynamic programming was introduced to iteratively optimize deformable models by Amini *et al.* [85]. It requires a large amount of memory and has high computational complexity. The complexity of the algorithm is  $O(n \cdot m^2)$ , where  $n$  is the number of tasks and  $m$  is the number of stages. In Amini's model,  $n = 1$  and  $m = L$ , where  $L$  is the length of the longest possible contour and so its complexity is  $O(L^2)$ . In our algorithm,  $n = L'$ , where  $L'$  is the actual contour length, and  $m = L_w$ . The complexity of our algorithm is  $O(L' \cdot L_w^2)$ . Since  $L' \leq L$  and  $L_w$  is a fixed number about 10 in most cases, we have  $O(L' \cdot L_w^2) \ll O(L^2)$ . Thus, our algorithm has a much lower computational complexity. The reduction is more significant while dealing with larger objects.

### 5.2.3 MAP Shape Estimation

In order to incorporate the influence of prior shape model to the contour detection process, the shape of each organ must be correctly estimated. Let  $\Psi$  denote the estimated curve of the object. At each step of the curve evolution, it is estimated by

$$\Psi_{\text{MAP}} = \underset{\Psi}{\operatorname{argmax}} p(\Psi | \mathcal{C}, G(I)) \quad (5.6)$$

where  $\mathcal{C}$  is the current curve and  $G(I)$  denotes the weighted graph of the image  $I$ . To compute the MAP shape representation, we expand (5.6) using Bayes' Rule.

$$\begin{aligned} p(\Psi | \mathcal{C}, G(I)) &= \frac{p(\mathcal{C}, G(I) | \Psi) p(\Psi)}{p(\mathcal{C}, G(I))} \\ &= \frac{p(\mathcal{C} | \Psi) p(G(I) | \mathcal{C}, \Psi) p(\Psi)}{p(\mathcal{C}, E_p(I))} \end{aligned} \quad (5.7)$$

The normalization term in the denominator of (5.7) can be discarded since it does not depend on the estimated shape of the object. Other terms are defined separately as follows.

The first term  $p(\mathcal{C} | \Psi)$  in (5.7) represents the probability of existence of a curve  $\mathcal{C}$  given the estimated shape representation  $\Psi$ . Note that this term does not include any image information whatsoever. This term is modeled as a Laplacian density function over the energy of the evolving curve with  $|\Psi|$  as its weight map.

$$p(\mathcal{C} | \Psi) = \exp \left( -\lambda_1 \int_{\mathcal{C}} |\Psi(\mathcal{C}(s))| ds \right) \quad (5.8)$$

The second term in the numerator of (5.7) computes the probability of seeing certain weighted graph  $G(I)$ , given a curve  $\mathcal{C}$  and an estimated curve representation  $\Psi$ . Since object contours usually exist in detected edges, we would expect that the estimated shapes also lie on those edges. Thus, this term can be modeled as a Laplacian density function of the degree of matching shape estimate and edges in

the image.

$$p(G(I)|\mathcal{C}, \Psi) = \exp \left( -\lambda_2 \int_{\Psi=0} G(x, y) dx dy \right) \quad (5.9)$$

The last term in the numerator of (5.7) represents the probability of the estimated shape  $\Psi$ , which is described earlier in Section 5.2.1. The prior shape estimator is a Gaussian model over the shape parameters,  $\alpha$  with shape variance  $\Sigma_k$

$$p(\Psi) = p(\mathbf{b}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_k|}} \exp \left( -\frac{1}{2} \mathbf{b}^T \Sigma_k^{-1} \mathbf{b} \right). \quad (5.10)$$

Putting these into (5.7) and taking the negative log, we have

$$\begin{aligned} \Psi_{\text{MAP}} = \operatorname{argmin}_{\Psi} \left\{ \lambda_1 \int_{\mathcal{C}} |\Psi(\mathcal{C}(s))| ds \right. \\ \left. + \lambda_2 \int_{\Psi=0} E_p(x, y) dx dy + \frac{1}{2} \mathbf{b}^T \Sigma_k^{-1} \mathbf{b} \right\}. \end{aligned} \quad (5.11)$$

Using the MAP shape estimator in (5.11), the prior shape of the object can be estimated. The MAP shape is re-estimated every several evolution steps using simple gradient descent method.

## 5.3 Results and Discussions

Our worm algorithm was applied on the synthetic image shown in Fig. 5.6(a). Clearly, the proposed algorithm is able to delineate sharp corners and to connect disjointed parts to obtain a closed contour. The worm searches for all the possible paths within a range defined by its body length and chooses the shortest one. In addition, the worm has much lower internal energy when it is on or near the edges. These characteristics help the worm fit into sharp corners. The worm stops when the curve is closed (*i.e.*, self-intersection is detected) as shown in Fig. 5.6(b). However, this path is not naturally the desired object contour as the starting point

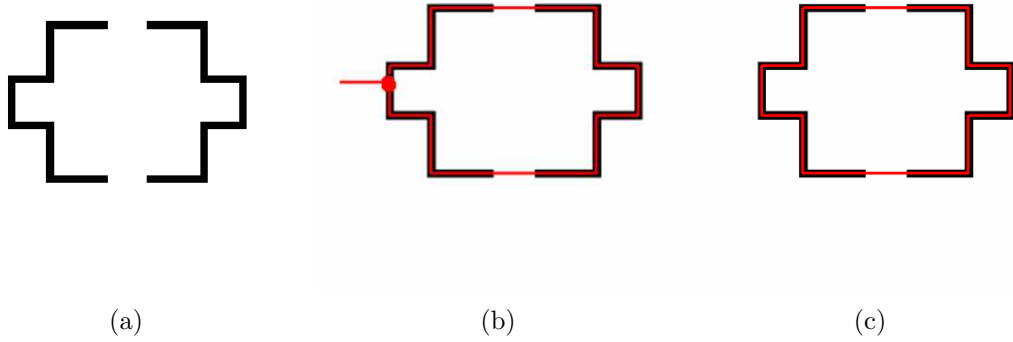


Figure 5.6: Illustrating the use of the worm algorithm for the synthetic image in (a), which consists of sharp corners and two breaks. In (b), the worm stops at the intersection point. (c) The final contour detection result.

may not be initialized on the contour. We need to detect the point from which the path begins to the actual object contour. The object contour is then obtained by giving a closed contour as shown in Fig. 5.6(c).

In the proposed object contour detection scheme, a statistical shape model is first built and represented implicitly as the example shown in the example of Fig. 5.5. Shape estimates are generated according to image information and the current detected contour using the MAP framework as demonstrated in (5.11). Then a worm starts from an initial point, moves along the minimal path under the influence of image force, its internal force and shape prior force, and finally stops when the curve is closed or image boundaries are reached. The final segmentation result is obtained by finding out the intersection point and removing the path before this point.

To reduce the complexity of our algorithm, of the four parameters  $\alpha$ ,  $\beta$ ,  $\lambda_1$ , and  $\lambda_2$ , we fix  $\beta = 1.0$  in (5.3), which implies that the image force has equal influence as the internal force on the evolution of the worm. We further set  $\lambda_1 = 0.1$  and  $\lambda_2 = 0.5$  in (5.11). With this setting, the shape priors have a major impact on the shape estimate while the evolving worm trail contributes the least since it is only part of the object contour. Then in our experiments, we just need to adjust



parameter  $\alpha$  in (5.3) to control the influence of shape prior.

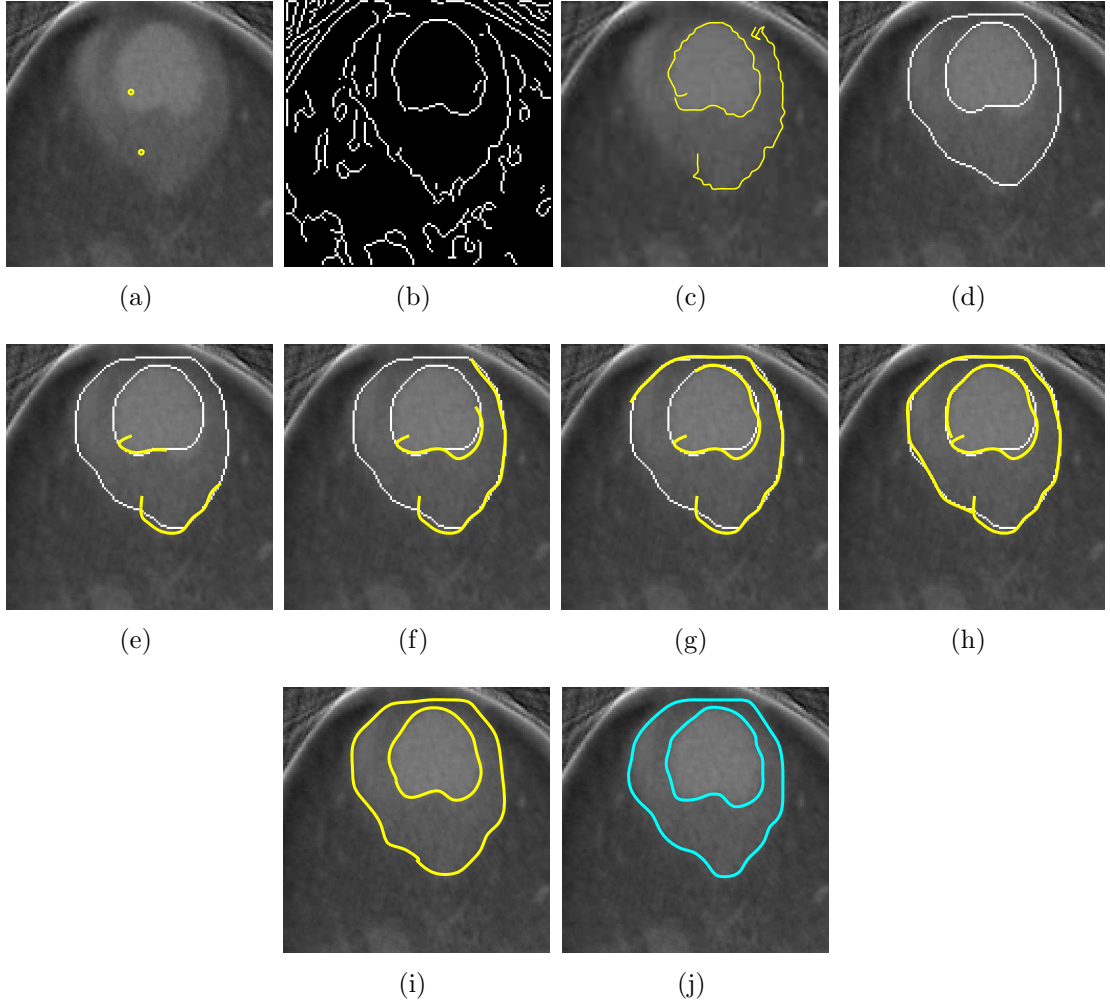


Figure 5.7: Illustration of segmentation process for CT cardiac image in (a) with two initial points, where  $\alpha = 0.6$  for the epicardium wall and  $\alpha = 0.2$  for the endocardium wall. (b) Detected edges. (c) Segmentation results without prior shape influence. (d) Initial shape estimates. (e)–(h) Intermediate segmentation results. (i) The final segmentation results. (j) Manual segmentation results.

Experiments on various medical images are carried out to demonstrate the performance of the proposed segmentation method. Fig. 5.7 shows the segmentation process of a CT cardiac image [1]. Segmentation results without using shape priors are first presented in Fig. 5.7(c), where the endocardium wall is extracted successfully while only part of the epicardium wall is obtained because no obvious edge information can be found. The situation is improved when shape estimates are considered. Fig. 5.7(d)–(i) show the segmentation process of our segmentation method in six steps. In our experimental results, thin curves indicate shape esti-

mates and thick curves represent segmentation results. When the edge information is weak, the shape estimate will guide the worm. In Fig. 5.7(h), we can see that the shape estimate of the endocardium wall is not very accurate since this image is quite different from the images in our training set. But the worm can still get the correct contour because the edge information is strong there. Thus, the worm does not totally rely on either the image information or the shape information. It balances the influence of image features and prior shape, and manages to obtain a better solution. The manual segmentation result (see Fig. 5.7(j)) is included for comparison.

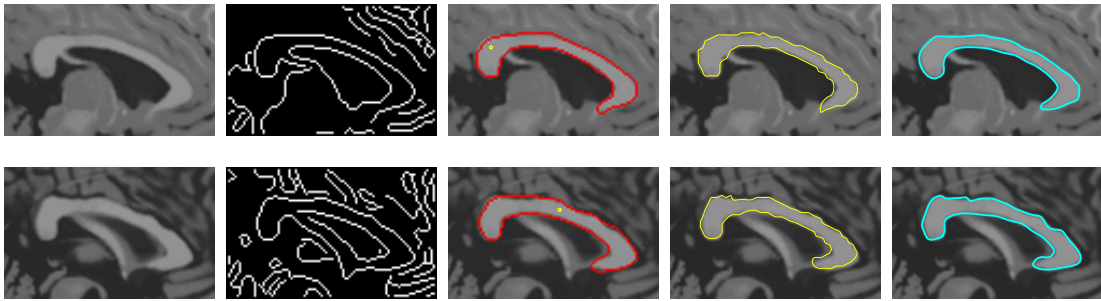


Figure 5.8: Segmentation results on MR brain images. In both experiments, we set  $\alpha = 0.8$ . The first row and the second row show the segmentation processes for data set one and data set two, respectively. First column: original images. Second column: edge maps. Third column: starting points and initial shape estimates. Fourth column: final segmentation results. Fifth column: Manual segmentation results.

In another experiment, the *magnetic resonance* (MR) T1 brain images [86] are segmented. Two sets of segmentation results are shown in Fig. 5.8. From the second column of Fig. 5.8, we can see that the edges of the object of interest are disconnected and are sometimes connected with other edges. Although the worm has the ability to skip breaks and join the disconnected parts, it is a very challenging task in that the worm can be easily “misled” by other salient edges. But with the aid of prior shape knowledge, the contours are successfully obtained as shown in the fourth column of Fig. 5.8. Manual segmentation results are provided in the fifth column of Fig. 5.8 for comparison.

To validate the results, the mean distance error is adopted as a metric to

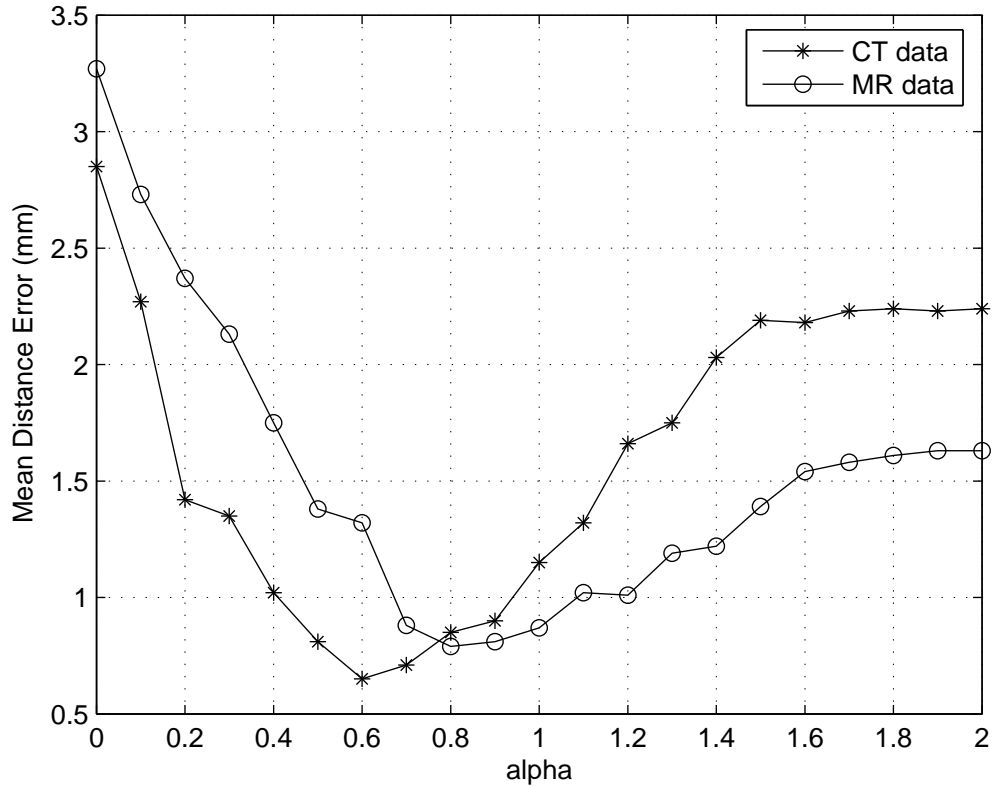


Figure 5.9: The effect of varying parameter  $\alpha$  on the segmentation errors.

measure the difference between the our segmentation results and the ground truth, *i.e.*, manual segmentation results obtained by radiologists. By using this metric, the effect of varying parameter  $\alpha$  is shown in Fig. 5.9. It can be seen that for both data sets the largest error occurs when  $\alpha = 0$ , which means no shape prior is used. By increasing the value of  $\alpha$ , which means by involving the shape prior more and more in the segmentation, the mean distance error gradually decreases and reaches a minimum point. Thus, the best balance between shape prior, image force and internal force is achieved. It is noted that when we continue to increase the weight of shape priors, however, the segmentation performance begins to degrade. The reason is that the shape of the structure is not able to be accurately estimated, which causes errors when the shape prior becomes dominant with large  $\alpha$ .

## 5.4 Summary

In this chapter, we propose a segmentation scheme which extracts object boundaries by finding a path with minimal energy. Segmenting organ from an image is then solved using graph searching method. An intelligent worm is used to evaluate the minimal path from a single initial point. A weighted graph of the image is designed for evaluating the paths. The worm can track complex topologies and link discontinued parts of object contour. To make it less sensitive to detected edges, an implicit prior shape model is incorporated and a more robust segmentation scheme is achieved. The corresponding MAP prior shape estimator is proposed and developed as well. Our approach requires a simple initialization task and has a low computational complexity. Promising segmentation results are obtained.

# Chapter 6

## Capillary Geodesic Active Contour

In this chapter, extracting cerebral vasculature from 3D *Magnetic resonance angiography* (MRA) images is studied. Brief review of existing MRA segmentation methods and some background information are provided in Section 6.1. The proposed *capillary geodesic active contour* (CGAC) is presented in Section 6.2. Section 6.3 provides the implementation details of the CGAC. Finally, experimental results are presented in Section 6.4 and Section 6.5 gives out a short summary.

### 6.1 Introduction

MRA is a noninvasive medical imaging modality that produces 3D images of vessels and cavities. Accurate extraction of 3D vascular structures from MRA images has become increasingly important for diagnosis and quantification of vascular diseases. A group of specific methods have been proposed for this particular kind of medical images. In this section, we first provide a brief review of current techniques and

then the CGAC is proposed. The background knowledge of capillary action is provided in Section 6.1.2. Finally, the state-of-the-art MRA segmentation algorithm CURVES is introduced in Section 6.1.3 for comparison purpose.

### 6.1.1 MRA Image Segmentation

Vessel segmentation algorithms are key components of automated radiological systems for vasculature diseases diagnosing. Existing segmentation methods vary greatly depending on the imaging modality, application domain, method being automatic or semi-automatic, and other specific factors. There is no single method that can extract vasculature from every medical image modality. While some methods employ pure intensity-based pattern recognition techniques such as thresholding followed by connected component analysis [87,88], some other methods apply explicit vessel models to extract the vessel contours [89,90]. Depending on the image quality and the general image artifacts such as noise, some segmentation methods may require image preprocessing prior to the segmentation algorithm [91,92]. On the other hand, some methods apply post-processing to overcome the problems arising from over segmentation.

In this section, we present a brief survey of MRA image segmentation techniques. A commonly used technique to evaluate MRA data sets, currently in hospitals, is *maximum intensity projection* (MIP) [74] (see Fig. 6.1). The MIP is generated by selecting the maximum value along an optical ray that corresponds to each pixel of the 2D MIP image. It is useful because the overall shapes and paths of the vessels become visible. However, it is known that the MIP may lead to underestimation of vessel width and a decreased *signal-to-noise ratio* (SNR) resulting in poor visualization of blood vessels [93]. Hence, more sophisticated solutions are needed to extract vasculature from 3D MRA images for better visualization results and a number of methods have been developed.

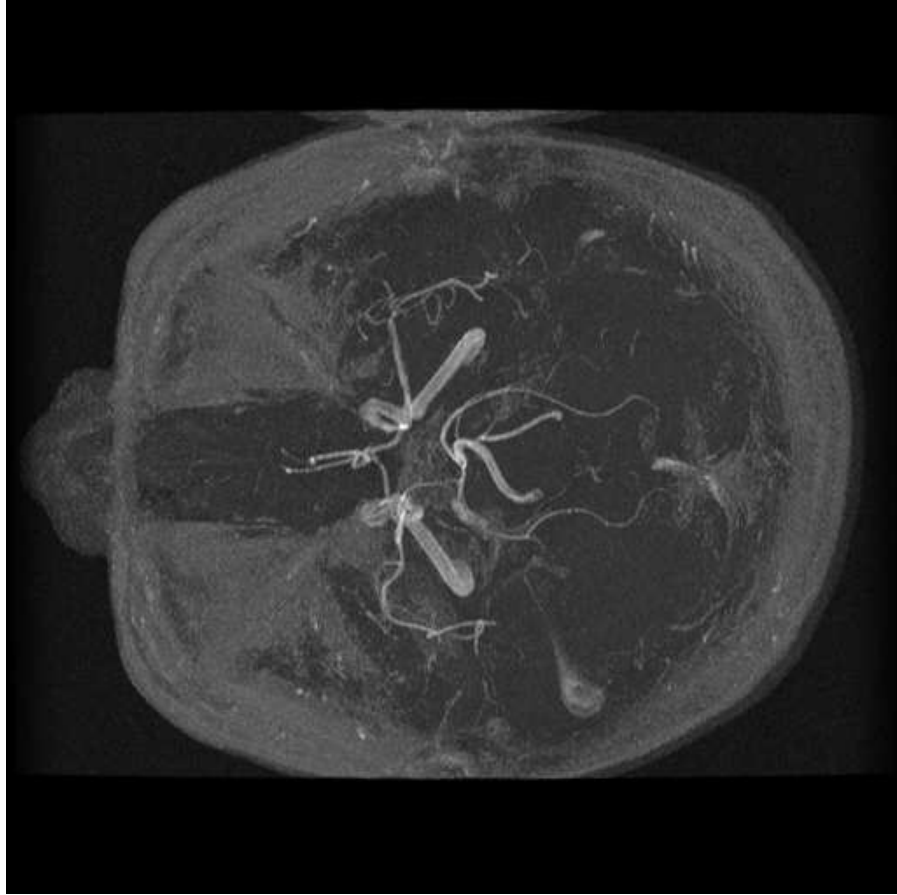


Figure 6.1: Maximum intensity projection of a cerebral MRA data set.

Existing MRA segmentation techniques can be broadly divided into two categories: skeleton-based and nonskeleton-based. Skeleton-based techniques are those indirect methods which segment and reconstruct the vessels by first detecting the centerlines of the vessels. Several methods have been developed based on this principle and multiscale schemes could be incorporated to allow for the diversity of vessel sizes [94–99]. In these approaches, the centerline can be generated explicitly, implicitly or via postprocessing by vessel modeling techniques. These methods involve applications of thresholding with object connectivity, thresholding followed by thinning procedure, and extracting based on graph description. The resulting centerline structure is used for 3D reconstruction of the vasculature. Contrary to above methods, nonskeleton-based techniques are those that compute the vessels in 3D directly. Here the vessel reconstruction is done without estimating the vessel cross sections. Many different methods have been proposed within this framework

like thresholding methods [74], fitting methods [94, 100], mathematical morphology based methods [101], fuzzy connectedness methods [2], deformable models based methods [90, 102–105], wave propagation based methods [106, 107], and so on. In this category, deformable model based methods have received considerable attention and success. They are discussed in more detail as follows.

Inspired by the success of deformable models in other image segmentation problems, Klein *et al.* [90] proposed to reconstruct 2D vessel boundaries or 3D vessel walls using deformable surface models represented by B-spline surfaces. However, it is not possible to employ parameterized deformable models to effectively deal with whole vessel trees, as the models would be required to change topology during evolution [108]. It is well known that the parameterized deformable models have difficulty in adapting topology in the evolving process unless some sophisticated mechanism is enabled [43, 109]. Yim *et al.* [103] proposed a deformable surface model based on triangulated meshes for vessel construction in 3D. Nevertheless, it may be problematic to apply these methods [90, 103] for segmentation of vessels from low contrast MRA images.

Segmentation methods using geometric deformable models, which are based on level set theory and can freely adapt into complex topologies of objects, were then proposed and applied on MRA images. Chen and Amini [102] employed a hybrid model using both parametric and geometric deformable models for segmentation of an entire vascular tree. In their work, the geometric deformable model is used to extract the vasculature and the parametric deformable model is employed to smooth the results. Descoteaux *et al.* [110] proposed to extract blood vessels from MRI data using the flux maximizing geometric flow algorithm, which is derived by Vasilevskiy and Siddiqi in [111] and implemented using level set method. However, these methods may have difficulty in extracting tiny vessels from 3D images. Small vessels and their branches, which exhibit much variability, are very important in planning and performing neurosurgical procedures. Greater details can



provide more precise navigation and localization information for computer guided procedures.

Recently, Lorigo *et al.* [104,112] proposed the “CURVES” (Curve Evolution for Vessel Segmentation) algorithm to extract thin vessels, which uses a GAC model based on a co-dimension two level set method grounded in [113]. However, to make the evolving curve stop at the object boundaries, the evolution speed needs to be multiplied by a heuristic factor. In addition, the vessels extracted by the CURVES algorithm are observed to be much thinner than those displayed in the images produced by MIP [112]. More detailed literature reviews on vessel extraction techniques can be found in [64] and [114].

Our work aims to develop an image segmentation methodology for automatically extracting the whole vasculature from 3D angiography. The proposed method aims to have the capability to recover thin vessels and obtain segmentation results that are as accurate as possible. Inspired by the common physical phenomenon of capillary action associated with capillary tubes as shown in Fig. 6.2(a), in which liquid climbs up to some height without external pulling force, an algorithm for vasculature extraction is proposed. In the capillary action, the thinner the tube, the higher the liquid level in the tube. The situation is very similar to the segmentation of blood vessels if we imagine thin blood vessels as capillary tubes. Hence, it may be useful to employ this mechanism for segmenting thin vessels in low contrast situations. By modeling this phenomenon using mathematics and fitting it into image segmentation problem, the capillary geodesic active contour is obtained. To apply to 3D segmentation, we introduce a 3D version, called *capillary geodesic minimal surface* (CGMS). However, for simplicity, we use CGAC to indicate the proposed algorithm in both 2D and 3D situations.

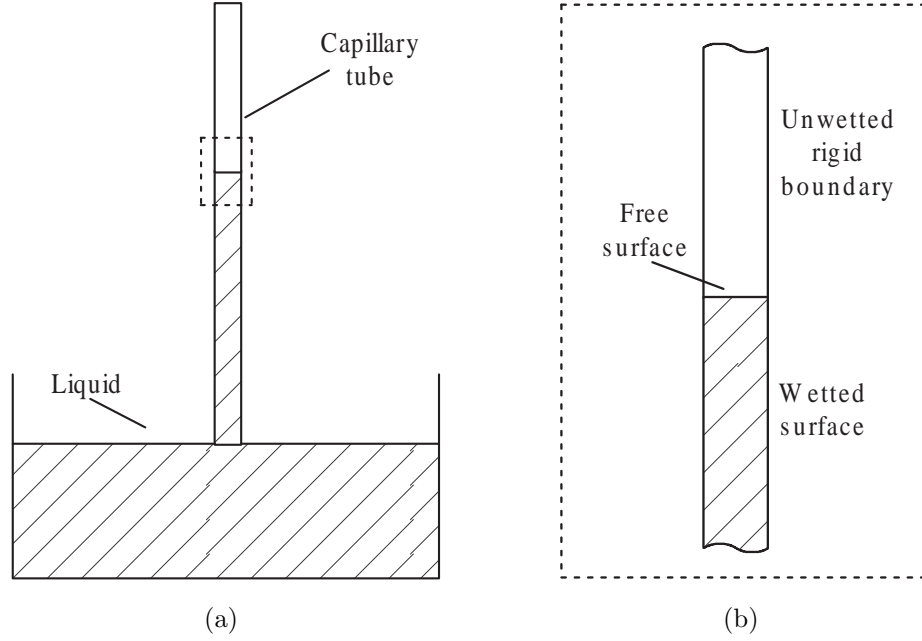


Figure 6.2: Illustration of capillary action. (a) Capillary tube, (b) Surfaces of a three-phase system.

### 6.1.2 Capillary Action

Capillarity is a broad subject in physics research [115]. Capillary action is one of its branches, which occurs in capillary tubes and can be easily observed. Capillary effects are caused by surface tensions, which include cohesion and adhesion. Cohesion is the attraction between molecules of the fluid and adhesion is the attractive force between molecules of fluid and solid tube boundary. For a general (three-phase) system consisting of fluid, gas and rigid bounding walls in a natural environment, the surface of fluid can be divided into two parts as shown in in Fig. 6.2(b), *free surface* and *wetted surface*. The free surface is the part of fluid surface in contact with gas, where only cohesion is considered. The wetted surface is the part of fluid surface in contact with rigid boundary, where adhesion dominates. Consequently, the energy of equilibrium capillary surface can be expressed as

$$E = \sigma_c S_f + \sigma_a S_w + \int \gamma \rho dx + cV. \quad (6.1)$$

Each term at the right side of Eqn. (6.1) describes an energy associated with the fluid. The terms are explained separately as follows.

- 1) *Free Surface Energy*: The free surface separates two media, fluid and gas. Its associated energy is the cohesion energy between molecules of fluid on the free surface. The energy for removal of fluid from the surface is proportional to the surface area, which is

$$E_f = \sigma_c S_f, \quad (6.2)$$

where  $\sigma_c$  is known as the cohesion coefficient. In other words, the surface tension coefficient  $\sigma_c$  in capillary action defines the needed energy to resize unit area of the fluid surface.

- 2) *Wetting Energy*: This term describes the adhesion energy between molecules of the fluid and the bounding wall on the wetted surface. It is analogous to free surface energy and represented by adhesion coefficient multiplying the area of the wetted surface as

$$E_w = \sigma_a S_w, \quad (6.3)$$

where  $\sigma_a$  is the adhesion coefficient.

- 3) *Gravitational Energy*: This potential energy is caused by the gravity in most cases. Assuming a more general potential energy  $\gamma$  per unit mass, depending on position within the media, the resultant energy is

$$E_\gamma = \int \gamma \rho dx, \quad (6.4)$$

where  $\rho$  is local density.

- 4) *Volume Constraints*: In our work, we assume that the fluid is incompressible. Then the volume of the fluid will be constant. The last term in Eqn. (6.1)

presents this constraint. A natural way to apply the volume constraint is to introduce the volume  $V$  multiplied by a Lagrange parameter  $c$  as a new energy term. This allows arbitrary displacements consistent with the constraint imposed by the rigid boundary. Thus, this term is expressed as

$$E_v = cV. \quad (6.5)$$

### 6.1.3 CURVES

To recover low contrast thin vessels from angiographies, Lorigo *et al.* [104, 112] proposed the *curve evolution for vessel segmentation* (CURVES) scheme. The main idea is to regularize a geometric flow in 3D using the curvature of a 1D curve, rather than the classical mean curvature of a 3D surface which tend to annihilate thin structures. However, the previous level set method no longer holds when evolving a 1D curve in 3D space, which is known as co-dimension two problem [113]. The problem can be described using level set as: let  $\Psi : \mathbb{R}^3 \rightarrow \mathbb{R}^+$  be an auxiliary function whose zero level set is exactly  $\mathcal{C}(p) : [0, 1] \rightarrow \mathbb{R}^3$ . Since it is impossible to define the inside and the outside of a 1D curve, the traditional level set evolution equation cannot be applied for such co-dimension two problems. Ambrosio and Soner [113] proved that evolving some 1D curve  $\mathcal{C}$  in 3D space according to

$$\vec{\mathcal{C}}_t = F\vec{\mathcal{N}} \quad (6.6)$$

is equivalent to evolving  $\Psi$  according to

$$\Psi_t = F(\nabla\Psi, \nabla^2\Psi), \quad (6.7)$$

where function  $F$  is defined as the smaller nonzero eigenvalue of matrix

$$P_{\nabla\Psi} \nabla^2\Psi P_{\nabla\Psi}. \quad (6.8)$$

Matrix  $P_q$  is defined as

$$P_q = I - \frac{q \otimes q}{|q|^2}, \quad q \neq 0, \quad (6.9)$$

where  $I$  is the identity matrix.

Based on this work, Lorigo *et al.* [104, 112] derived the geodesic active contour speed function for curve evolution through computing the Euler-Lagrange equation and got

$$\vec{C}_t = \kappa \vec{N} - \frac{\nabla g}{g} \Pi \left( \mathbf{H} \frac{\nabla I}{|\nabla I|} \right) \quad (6.10)$$

where  $\mathbf{H}$  is the Hessian of the intensity function. According to the co-dimension two level set theory [113], the level set update equation of Eqn. (6.10) is represented as

$$\Psi_t = F(\nabla \Psi, \nabla^2 \Psi) + \rho(\nabla \Psi \cdot \nabla I) \frac{g'}{g} \nabla \Psi \cdot \mathbf{H} \frac{\nabla I}{|\nabla I|}. \quad (6.11)$$

The multiplier  $\rho(\nabla \Psi \cdot \nabla I)$  in Eqn. (6.11) is added heuristically in order to make the evolution curve stop at object boundaries.

## 6.2 Modeling the CGAC

One of the difficulties in the segmentation of MRA images is the accurate extraction of small vessels. To deal with this problem, we propose a new capillary action based geodesic active contour method, which is modeled based on the physical phenomenon of capillary action. According to the theory of capillarity [115], the fluid surface can be divided into two parts, free surface and wetted surface. The free surface is the part of the liquid surface that does not come into contact with solid boundaries while the wetted surface is the part that is in contact with the solid boundaries. Since the capillary action can be understood as an energy minimization process, which involves surface tensions, we first introduce the energy functional associated with the free surface and wetted surface. Following that, the volume constraint of the fluid is considered. Thus, the phenomenon is modeled as

a minimization problem with constraint. To minimize the functional, the Euler-Lagrange equations are computed for each part using *variational calculus* [116]. Finally, these derived parts are integrated together to obtain the overall evolution equation of the fluid surface. Since the capillary force is incorporated, the derived method is called capillary geodesic active contour.

### 6.2.1 Free Surface Energy

*Free surface* is the part of the liquid surface that is not in contact with the solid boundaries. Capillary action can only be observed when the adhesion tension between molecules of fluid and molecules of solid is larger than the cohesion tension between fluid molecules. Therefore, molecules on the free surface tend to be attracted toward the solid boundaries because the adhesion tension is larger than the cohesion tension. Thus, fluid molecules will have lower potential energy when they are nearer to solid boundaries and this energy will be minimized when they are on the solid boundaries.

When applying to MRA image segmentation, solid boundaries refer to blood vessel walls. Fluid surface can be considered as the evolving surface to approach the vasculature. Thus, we calculate the gradient magnitude  $|\nabla I|$  to detect possible boundaries, where  $I : [0, a] \times [0, b] \times [0, c] \rightarrow \mathbb{R}^+$  denotes an image. A uniform decreasing function  $g : [0, +\infty[ \rightarrow \mathbb{R}^+$  is defined, where  $g(r) \rightarrow 0$  as  $r \rightarrow +\infty$ . Function  $g(|\nabla I(S(q))|)$  is used to describe the energy coefficient associated with the surface  $S(q) : [0, 1] \rightarrow \mathbb{R}^3$ , which is the surface tension coefficient in physics. To simplify the notation, we write  $g$  or  $g(S(q))$  for  $g(|\nabla I(S(q))|)$ . Let  $S_f(t)$  denote the free surface of the liquid at time  $t$ . Recall the definition of free surface energy in Section 6.1.2, then the capillary energy of free surface can be expressed as

$$E(S_f(t)) = \int_q g(S(q)) \left| \frac{\partial S(t, q)}{\partial q} \right| dq \quad (6.12)$$

by integrating over  $q$ , where  $S(t, q) \in S_f(t)$ .

Since the equilibrium of surface tensions is achieved when the potential energy is minimized, our objective here is to minimize the energy in Eqn. (6.12). The steepest-descent method is used to find out the solution of minimal energy. By computing the Euler-Lagrange of (6.12), the evolution equation of the free surface is obtained as

$$\vec{S}_t = g\kappa\vec{N} - (\nabla g \cdot \vec{N})\vec{N}, \quad (6.13)$$

where  $\kappa$  is the Euclidean curvature, and  $\vec{N}$  is the unit normal vector of the surface. The above equation is exactly the same as the evolution equation of the geodesic active contour in 2D [53] or that of the minimal surface in 3D [70].

### 6.2.2 Wetting Surface Energy

The fluid surface in contact with solid boundary is known as the *wetted surface* [115]. Since the adhesion force is larger than the cohesion force, the molecules on the free surface will be attracted to the solid boundaries and the unwetted surface is then converted into wetted surface. The movement is in the tangential directions of the solid boundaries since it is the shortest way to cover the unwetted surface by fluid. Once the fluid comes into contact with the solid boundaries, the area of contact will be considered as wetted surface. The total energy involved in this dynamic process can be modeled as

$$E(S_w(t)) = \beta S_w + \beta^* S_w^*, \quad (6.14)$$

where  $S_w$  is the area wetted by the fluid and  $S_w^*$  is the area in contact with the outer medium (unwetted surface). Variational calculus is used to analyze this dynamic energy. When making the variation, only those values of  $S_w$  and  $S_w^*$  in an immediate neighborhood of the contact line need to be taken into account. Neglecting terms

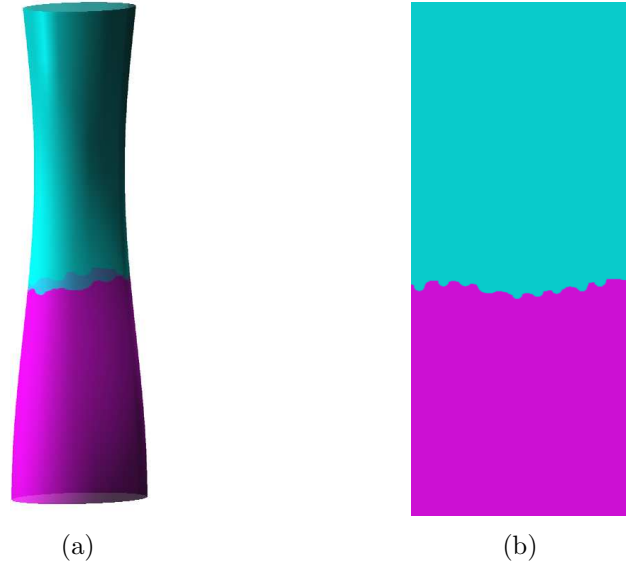


Figure 6.3: 3D tubular surface in (a) is stretched to get 2D surface in (b). Minimizing lower area of the 2D surface through evolving the contact line.

that are constant and noting that any variation of  $S_w$  is the negative of that of  $S_w^*$ , we may extend  $S_w$  in an arbitrarily continuous way into  $S_w^*$ , and write

$$E(S_w(t)) = \hat{\beta} S_w^* \quad (6.15)$$

where  $\hat{\beta} = \beta^* - \beta$ . Here  $S_w^*$  includes a neighborhood of the contact line and can be changed only by evolving the contact line. Then according to Eqn. (6.15), minimizing the wetted surface energy is equivalent to minimizing the surface  $S_w^*$  through evolving the contact line. However, solving this problem directly in 3D is problematic, because the surface to be minimized is not enclosed by the contact line. Since we are only interested in an immediate neighborhood of the contact line, it is always possible to find some view point that all the surface area is under the contact line (see Fig. 6.3). In addition, the 3D surface can be “cut” along a line and then stretched to becoming a 2D plane as shown in Fig. 6.3. This is inspired by the atlas of the world, where the surface of the earth is flattened to get a 2D map. Although it is not always possible to get a perfect 3D to 2D conversion, it can be a good approximation.



Through the above conversion, the problem of minimizing a 3D surface can be solved as a 2D one. Let  $\mathcal{C}(t, x) : [0, 1] \rightarrow \mathbb{R}^2$  denote the contact line between the fluid surface and the unwetted surface. Now we need to minimize the surface area under curve  $\mathcal{C}$  by evolving itself. In addition, the length of the curve  $\int \left| \frac{\partial \mathcal{C}(t, x)}{\partial x} \right| dx$  is considered as a regularization term to keep the curve smooth. Thus, the energy associated with the wetted surface in Eqn. (6.15) can be rewritten as

$$E(S_w(t)) = \int \mathcal{C}(t, x) dx + \lambda \int \left| \frac{\partial \mathcal{C}(t, x)}{\partial x} \right| dx. \quad (6.16)$$

where  $\lambda$  is a real positive constant parameter. To minimize the energy functional, the Euler-Lagrange of Eqn. (6.16) is computed. By using the steepest descent methods, the 2D evolving equation of contact line  $\mathcal{C}$  is obtained as

$$\vec{\mathcal{C}}_t = (1 + \lambda \hat{\kappa}) \vec{\mathcal{N}}_\Gamma, \quad (6.17)$$

where  $\hat{\kappa}$  and  $\vec{\mathcal{N}}_\Gamma$  are the Euclidean curvature and the unit normal vector of the contact line in 2D, respectively. As we have shown that the evolving equations of the contact line are the same for both 2D and 3D cases, Eqn. (6.17) can be applied for evolving the curve  $\mathcal{C}$  in 3D. However, since the derivation is done in 2D, an extension operation is needed before it can be applied to minimize 3D surfaces. Firstly, when working in 3D, the curvature  $\hat{\kappa}$  of the 2D contact line becomes a co-dimension two curvature  $\hat{\kappa}_2$  [104, 112, 113]. Furthermore, since we are considering the evolution of liquid along vessel walls, the direction of evolution should be parallel to the wall at each point, *i.e.*, the tangential direction of the wall. Hence, the normal vector  $\vec{\mathcal{N}}_\Gamma$  of the contact line in (6.17) is actually the tangential part of the liquid surface's normal vector at each point, *i.e.*,  $\vec{\mathcal{T}}_{sb}$  as shown in Fig. 6.4. Then Eqn. (6.17) is changed into

$$\vec{\mathcal{C}}_t = (1 + \lambda \hat{\kappa}_2) \vec{\mathcal{T}}_{sb}. \quad (6.18)$$

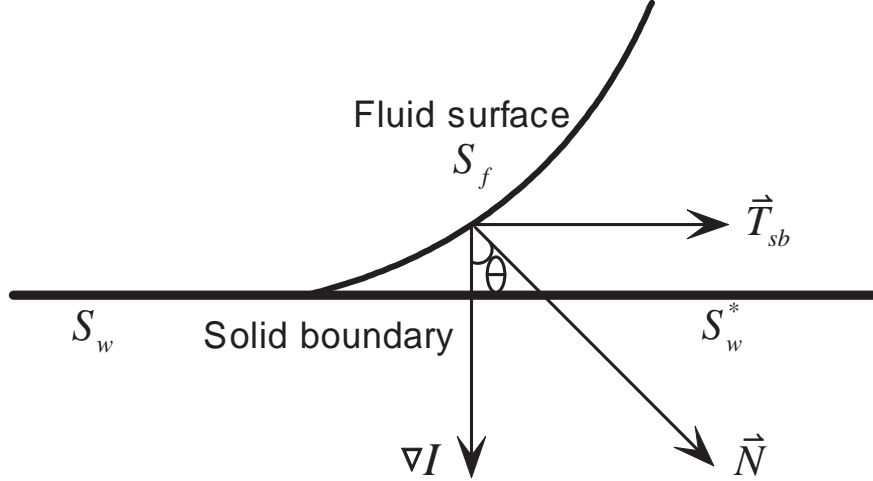


Figure 6.4: The evolving direction of the contact line is the tangential sub of the surface normal direction.

Since blood vessels in MRA images appear brighter than the background, the directions of gradients in the image could be known. Thus, the tangential subvector can be calculated by using the cosine of the angle between the normal to the evolution surface and the gradient in the image as shown in Fig. 6.4. It is expressed in mathematics as

$$\vec{T}_{sb} = \vec{N} - \frac{\nabla g}{|\nabla g|} \cos \theta, \quad (6.19)$$

where

$$\cos \theta = \vec{N} \cdot \frac{\nabla g}{|\nabla g|}. \quad (6.20)$$

Thus, the overall evolution equation describing the energy of wetting surfaces can be written as

$$\vec{S}_t = (1 + \lambda \hat{\kappa}_2) \left( \vec{N} - \frac{\nabla g}{|\nabla g|} \cos \theta \right). \quad (6.21)$$

### 6.2.3 Volume Constraint

In our process of modeling capillary action, the fluid is assumed to be incompressible. Hence, the volume of the given fluid is constant and this constraint needs to

be considered in the fluid surface evolution equation. As in Section 6.1.2, volume constraint is considered as an energy term  $E(V) = cV$ , which can be expanded as

$$E(V) = c \int S(t, q) dq, \quad (6.22)$$

where  $S$  is the whole surface area of the fluid. By applying the gradient descent method for minimization, we have evolution equation

$$\vec{S}_t = c\vec{\mathcal{N}}, \quad (6.23)$$

which is a constant velocity for minimizing the volume enclosed by the surface. Since liquid is bounded by solid walls, the fluid surface will snap to the boundaries finally. Thus, the constant velocity is multiplied by function  $g$  and then the evolution equation is modified into

$$\vec{S}_t = g(S(q))c\vec{\mathcal{N}}. \quad (6.24)$$

### 6.2.4 Evolution Equation

Based on the results obtained separately in the above sections, the final evolution equation is obtained by integrating these terms

$$\vec{S}_t = g(\kappa + c)\vec{\mathcal{N}} - (\nabla g \cdot \vec{\mathcal{N}})\vec{\mathcal{N}} + \alpha(1 + \lambda\hat{\kappa}_2) \left( \vec{\mathcal{N}} - \frac{\nabla g}{|\nabla g|} \cos \theta \right), \quad (6.25)$$

where parameter  $\alpha$  is a real positive constant. The constant term  $c$  in Eqn. (6.25) acts like balloon force in [55], which facilitates the evolving surface snapping to solid boundaries. Comparing the new speed function in Eqn. (6.25) with those of the geodesic active contour [53] and its 3D correspondence minimal surface [70], the third term is new which comes from the capillary action. According to its definition, the new term only affects the part of evolving surface, which is around

the contact line. It makes the fluid surface move along the solid boundaries. The capillary action term is expected to facilitate the evolving surface adapting into thin parts of objects, *e.g.* thin vessels. In the CURVES algorithm [104, 112], the second term also affects the evolution when the curve is near object boundaries. However, the curve evolves still in the normal direction and may not be able to freely adapt into thin objects. Furthermore, without heuristically incorporating a weighting term, the evolution surface can not stop at the vessel boundaries as expected.

## 6.3 Implementation

In this section, the implementation of capillary geodesic active contour is discussed. The CGAC is mainly used for vessel segmentation from MRA images. Since the vasculature usually has complex topology, the implementation of CGAC should provide great topology adaptability. The level set method is chosen for the implementation. The level set methods are able to handle sharp corners and cusps in the propagating solution, as well as topological changes. In addition, the computational labor is no higher than other methods, with the advantages of increased accuracy and robust modeling [61].

### 6.3.1 Level Set Evolution Equation

According to the level set theory and noting the fact that

$$\vec{\mathcal{N}} = -\frac{\nabla\Psi}{|\nabla\Psi|}, \quad (6.26)$$

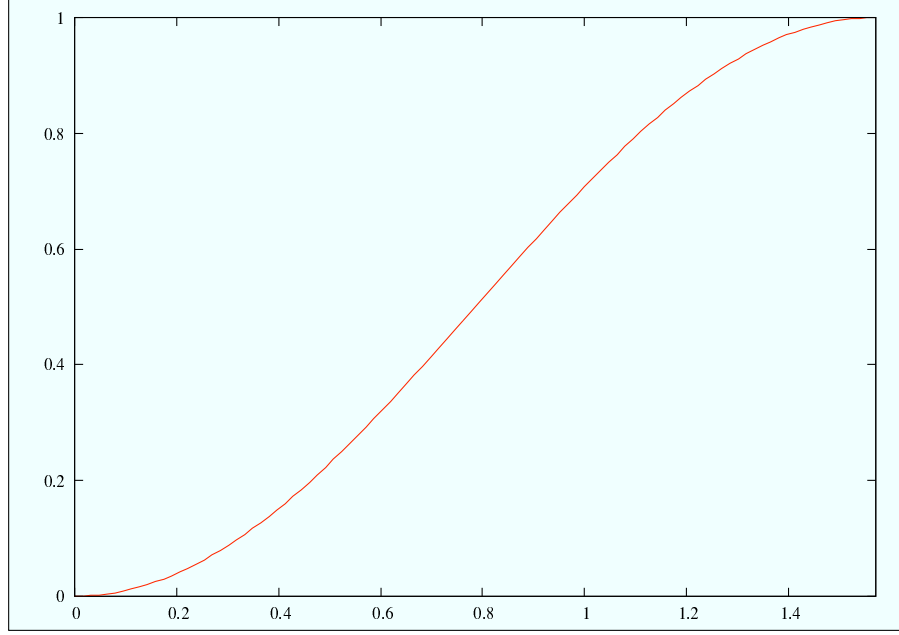


Figure 6.5: Illustration of the magnitude variation of  $1 - \cos^2 \theta$  with respect to the value of angle  $\theta$ .

evolving a surface  $S$  under the speed function (6.25) is equivalent to updating a volumetric map  $\Psi$  with

$$\Psi_t = g(\kappa + c)|\nabla\Psi| + \nabla g \cdot \nabla\Psi + \alpha(1 + \lambda\hat{\kappa}_2)|\nabla\Psi|(1 - \cos^2 \theta) \quad (6.27)$$

where  $S$  is the zero level set embedded in  $\Psi$  and

$$\cos \theta = \frac{\nabla\Psi \cdot \nabla g}{|\nabla\Psi||\nabla g|}. \quad (6.28)$$

Details of deriving Eqns. (6.27) and (6.28) according to the level set theory are presented in Appendix A. Eqn. (6.27) is the level set speed function of CGAC, which is incorporated with the capillary force. The last term at the right side of Eqn. (6.27) describes the capillary force in CGAC. Note that in Eqn. (6.27) the strength of the capillary force mainly depends on the magnitude of  $1 - \cos^2 \theta$ , whose variation is shown in Fig. 6.5. It has the minimal value zero when  $\theta = 0$ . When  $\theta$  is small, the contribution of capillary force is little as well. The contribution is slowly increasing and the maximum is achieved when  $\theta = \pi/2$ . However, we expect

that the capillary force contributes to the surface evolution even when the angle  $\theta$  is small. Therefore, a function  $f(x)$  is added into the updating equation (6.27) to make the contour evolve faster and in a more robust way. Then the last term at the right side of Eqn. (6.27) is changed into

$$\alpha(1 + \lambda\hat{\kappa}_2)|\nabla\Psi|f(1 - \cos^2\theta). \quad (6.29)$$

Theoretically, function  $f(x)$  can be any uniquely increasing function with minimum value zero and maximum value one. The point is that it should facilitate the surface evolution in the tangential direction of the boundary where the angle between surface normal and boundary normal  $\nabla g$  is small. In our implementation, function  $f$  is chosen to be a sigmoid function

$$f(x) = \frac{1}{\left(1 + e^{-\left(\frac{x-b}{a}\right)}\right)}, \quad (6.30)$$

where  $a$  decides the slope of the output curve and  $b$  defines the point around which the output window is centered. Effects of varying parameters  $a$  and  $b$  in function (6.30) are illustrated in Fig. 6.6.

The mean curvature  $\kappa$  in Eqn. (6.27) is computed as

$$\kappa = \operatorname{div} \left( \frac{\nabla\Psi}{|\nabla\Psi|} \right) \quad (6.31)$$

and  $\hat{\kappa}_2$  under codimension-two curve evolution is calculated according to Eqn. (6.7) which is introduced by Ambrosio and Soner [113]. As a summary, the final level set update equation is represented as

$$\Psi_t = g(\kappa + c)|\nabla\Psi| + \nabla g \cdot \nabla\Psi + \alpha(1 + \lambda\hat{\kappa}_2)|\nabla\Psi|f(1 - \cos^2\theta). \quad (6.32)$$

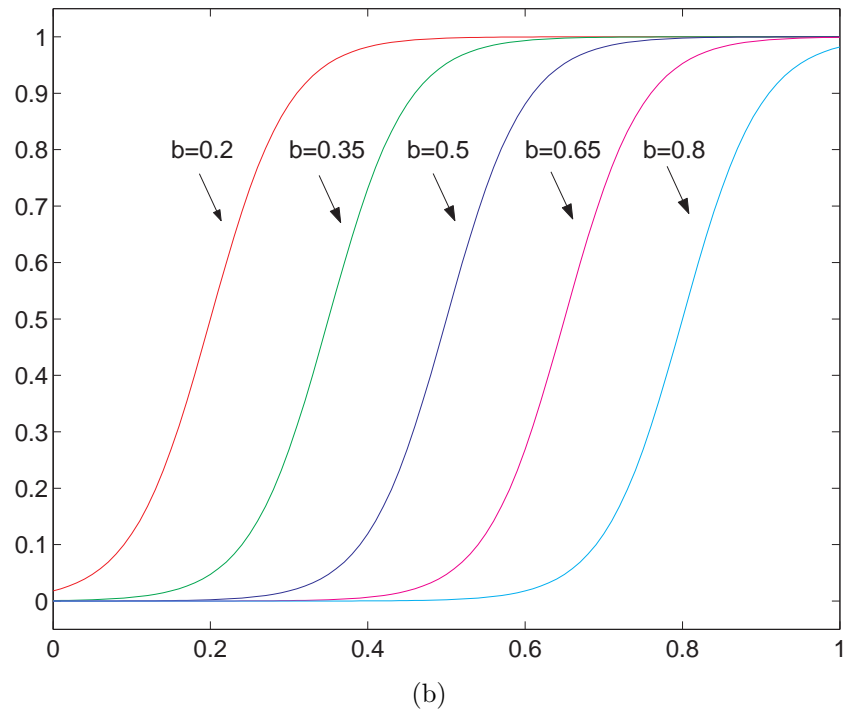
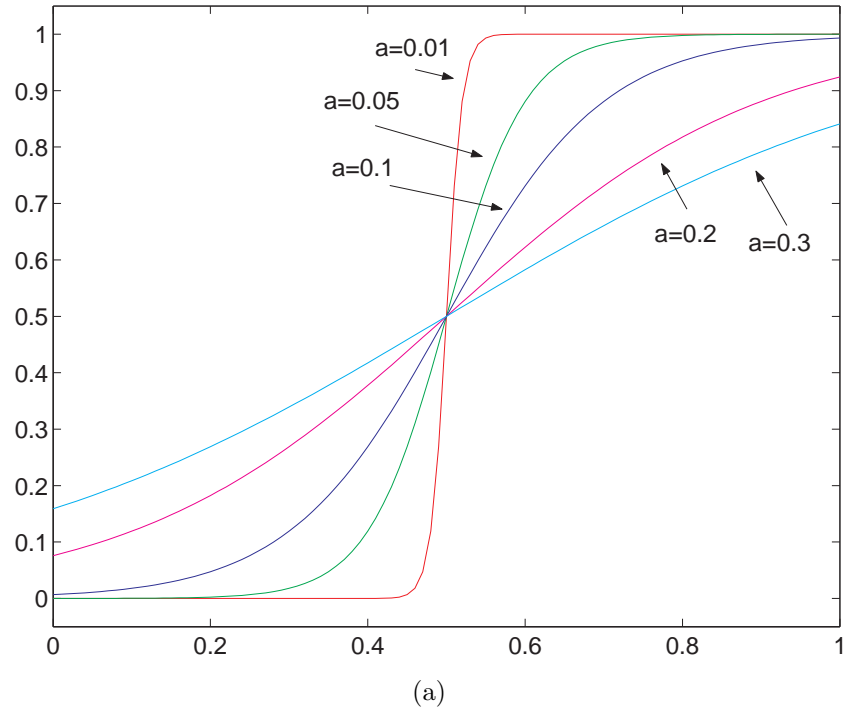


Figure 6.6: Illustration of various parameter settings for the sigmoid function  $f$ . (a) Effects of varying  $a$  under  $b = 0.5$ ; (b) Effects of varying  $b$  under  $a = 0.05$ .

### 6.3.2 Numerical Implementation

Our system takes in a 3D image that contains thin tubular structures, such as an MRA image. An initial segmentation estimate is generated by thresholding the image. That thresholding result is used to generate an initial signed distance function  $\Psi_0$ , which has negative values inside objects and positive values at outside. Then level set function  $\Psi$  is iteratively updated according to

$$\Psi^{n+1} = \Psi^n + \Delta\Psi^n \Delta t, \quad (6.33)$$

where  $\Delta\Psi$  is calculated using Eqn. (6.32). Convergence is achieved when volumetric change is very small over some iterations.

Before applying the algorithm, the image is smoothed by using a small isotropic Gaussian since the level set algorithm inherently requires some smoothness of gradients. Further, the level set map  $\Psi$  is periodically reinitialized to be a signed distance function: the zero level set  $S$  is extracted from  $\Psi$ , and then the value at each point is set to be its distance to  $S$ . This is needed because  $\Psi$  is defined as the signed distance map of  $S$ , which, however, cannot be ensured during the evolving process.

It is noted that when computing  $\cos \theta$  according to Eqn. (6.28), singular values may be produced if the value of  $|\nabla\Psi|$  or  $|\nabla g|$  is too small, even “division by zero” errors can happen. This could be the case when the evolving surface is far away from vessel walls. Under this situation, capillary force should not be applied because capillary action occurs only near or at the solid boundaries. Thus, in our implementation, when either  $|\nabla\Psi|$  or  $|\nabla g|$  is lower than a predefined threshold, the value of  $\cos \theta$  is set to be one. Then the contribution of capillary force becomes zero according to (6.27).



### 6.3.3 Toolkits

The algorithm described in this chapter has been coded for 3D segmentation based on the *insight segmentation and registration toolKit* (ITK) [117], an open source software developed as an initiative of the United States National Library of Medicine and freely available at [www.itk.org](http://www.itk.org). A large number of leading edge segmentation and registration algorithms have been implemented inside this toolkit.

In our research work, segmentation results are visualized using surface rendering. Hence, isosurface is required to be extracted for visualization. An isosurface is a surface that passes through all locations in space where a continuous data volume is equal to a constant value. The construction of isosurfaces is a well-studied problem [73]. The result of a typical isosurface algorithm usually consists of a set of triangle meshes that are discrete representations of the corresponding continuous isosurfaces.

Our isosurface extracting and surface rendering programs are implemented based on the *visualization toolKit* (VTK) [73], which is also an open source software for 3D computer graphics, image processing, and visualization. VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as implicit modelling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. In addition, dozens of imaging algorithms have been directly integrated to allow the user to mix 2D imaging/3D graphics algorithms and data. It is freely available at [www.vtk.org](http://www.vtk.org).

## 6.4 Results and Discussions

The proposed capillary geodesic active contour is applied on both synthetic image and 3D MRA data sets for testing and comparison.

### 6.4.1 Illustration of Capillary force

We first study the effectiveness of the capillary force in tubular objects segmentation. Fig. 6.7 shows the application of the CGAC in a computer generated cylinder, which illustrates the effects of the capillary force. In this experiment, initialization is done by using the fast marching method [61], which can easily create a signed distance map for some simple geometric shape. In this example, a sphere is generated from a given seed point and a specified diameter. The zero level set of the initial signed distance map is visualized in Fig. 6.7(b).

Starting from the initialization, evolution is done according to the level set speed function in Eqn. (6.27). In our experiments, all the parameters are fixed except that  $\alpha$  is varying. The parameter  $\alpha$  controls the influence of capillary force. The evolution results with different parameter settings after 50 iterations are visualized in Fig. 6.7.

When  $\alpha$  is set to zero, the CGAC in fact evolves exactly like a GAC. The evolution of the free surface in the axial direction of the tubular object is driven by the mean curvature  $\kappa$  and constant speed  $c$  in Eqn. (6.27). Since this term is multiplied by the uniquely decreasing function  $g$ , its contribution will be small when the surface is near the object boundary, and becomes zero when the surface is on the edges. Hence, we can see the free surface is convex as shown in Fig. 6.7(c). When the tubular object becomes much thinner, the evolution may stop because of the small value of  $g$  everywhere. After incorporating the capillary force, the free surface is attracted by the unwetted surface when near the solid boundaries and *meniscus* [115] can be observed as shown in Fig. 6.7(d)(e)(f). The capillary action is caused by adhesion between elements of solid boundary and fluid molecules. Adhesion of liquid to the vessel walls causes an upward force on the liquid at the edges in this example and results in a meniscus which turns upward. The surface tension acts to hold the surface intact, so instead of just the edges moving upward, the whole liquid surface is dragged upward. With larger value of  $\alpha$ , the surface

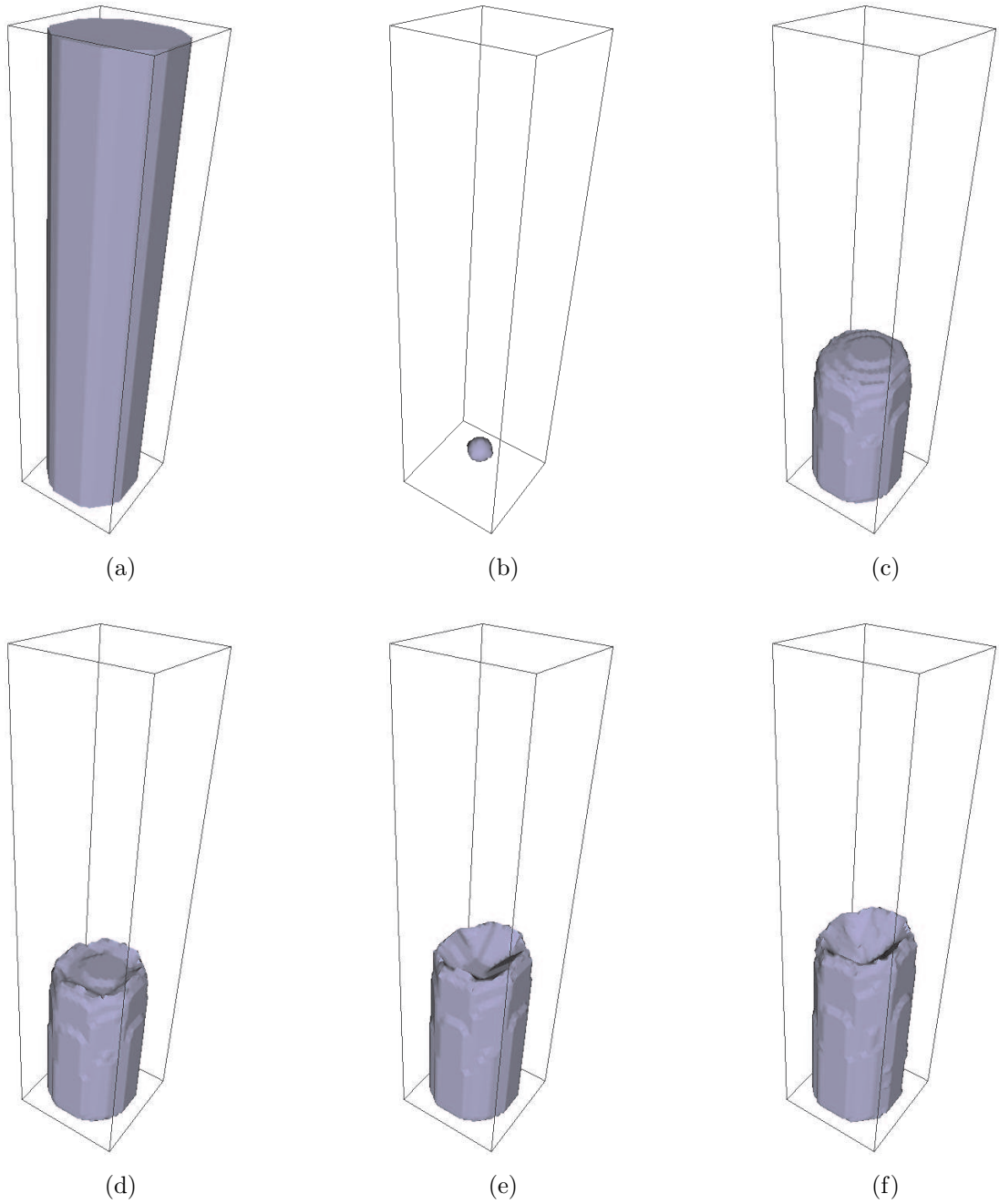


Figure 6.7: Illustration of the effects on varying capillary force coefficient  $\alpha$ . (a) Generated cylinder. (b) Initialization of the algorithm. (c)  $\alpha = 0$ . (d)  $\alpha = 0.25$ . (e)  $\alpha = 0.5$ . (f)  $\alpha = 0.75$ .

evolution along the object boundary will be stronger and faster. Therefore, the capillary force can facilitate the evolution of the free surface even when the vessels are very thin.

### 6.4.2 Segmentation Results of 3D MRA Images

The CGAC method is applied on 3D MRA images for segmenting blood vessels. Some of the results are presented in this section. The size of image A in our experiments is  $512 \times 512 \times 120$  with spacing  $0.43 \times 0.43 \times 1.2$ . Some sample 2D slices of the MRA data are shown in Fig. 6.8. The initial surface is obtained by thresholding the raw dataset. The initialization should be inside the expected segmentation results, because the concept of “*Once a particle is burnt, it stays burnt*” [61] is used in our implementation for robustness and computational simplicity. The MIP of MRA data set A is shown in Fig. 6.1. To make the results clearer, a region of interest (ROI) is extracted (Fig. 6.9) and only vessels inside the ROI are segmented and visualized.

Our proposed CGAC method is compared with the state-of-the-art CURVES in [104, 112]. Identical parameter settings in the evolution equations are used for both methods except for the capillary force coefficient, which is specific to the CGAC. The segmentation results of CURVES and CGAC are visualized in Fig. 6.10 and Fig. 6.11 with different points of view, respectively. From these results we can see that although the CURVES can extract most parts of the vasculature, compared with our CGAC algorithm, it fails to identify some thin parts of blood vessels. This is because when the vessel is very thin, the evolution terms of CURVES become zero and the evolving surface will stop. Thus, some thin vessels can not be extracted. The CGAC behaves in a different way that the evolving surface will move along the boundaries under the capillary force when the vessels are very thin. Thus, more details of the vasculature can be obtained by using our CGAC method.

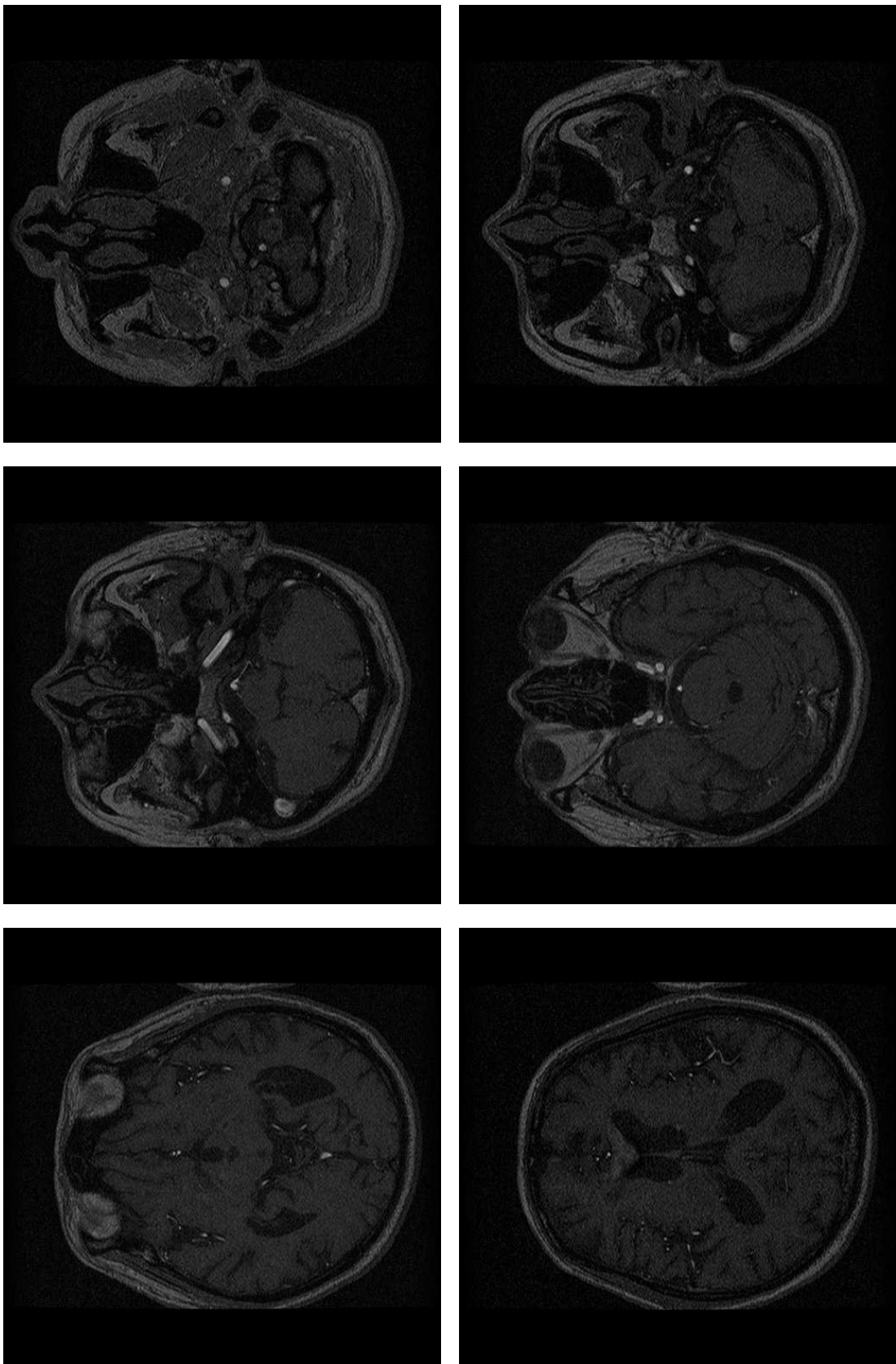


Figure 6.8: Samples of MRA data set A. Bright regions and points are blood vessels.

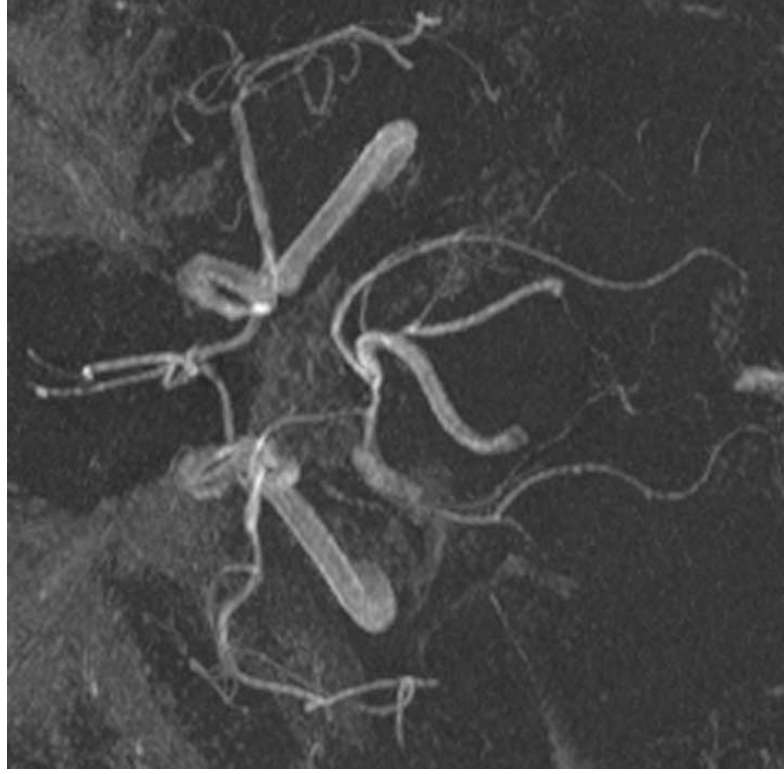


Figure 6.9: MIP of the region of interest of cerebral MRA data set A.

The size of the 3D image B is  $384 \times 320 \times 80$ . From the MIP shown in Fig. 6.12, it is noted that the signal to noise ratio is low and the intensities vary greatly from left to right. In addition, the voxel intensities of other tissues are quite similar to those of the blood vessels. For comparison, the *fuzzy connectedness* segmentation method [2] is also applied on the image. Segmentation results of fuzzy connectedness method and our CGAC method are visualized and shown in Fig. 6.13 and Fig. 6.14, respectively. From Fig. 6.13, we can see that the main connected vessels are extracted. However, when the image intensity varies, the rightmost part of the vessel in Fig. 6.12 is not extracted. In addition, the surface of the extracted vessels is not smooth, due to the presence of strong noise. When our CGAC method is used, the segmentation result is much improved as shown in Fig. 6.14. The whole vessel is extracted successfully and the surface of the vessel is quite smooth. The reason is that the evolving surface is attracted toward the boundaries and moves along the boundaries under the capillary force. Thus, it is not sensitive to the variance of intensities. Furthermore, the curvature term  $\kappa$  in

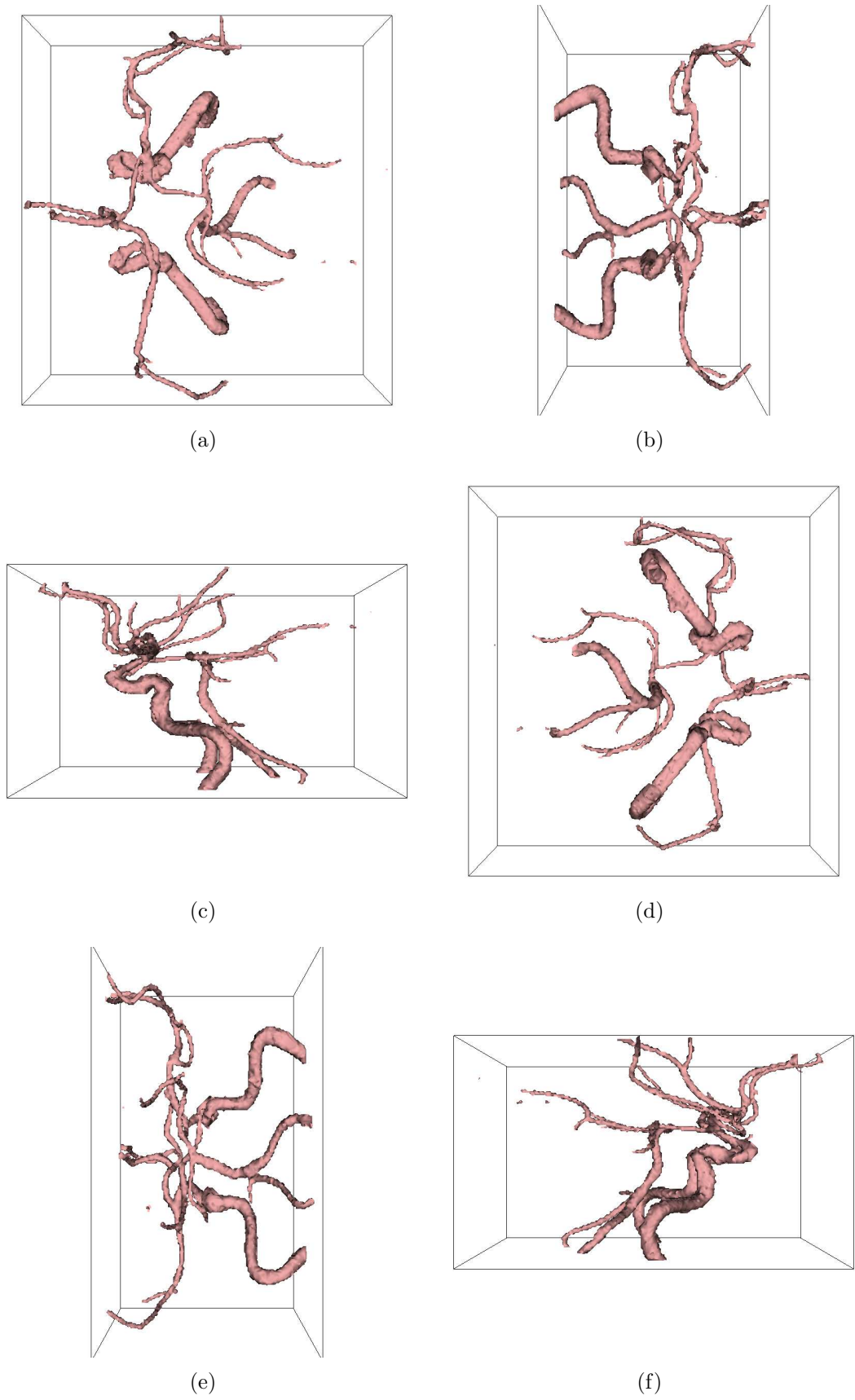


Figure 6.10: MRA segmentation results of the CURVES algorithm with different view points.

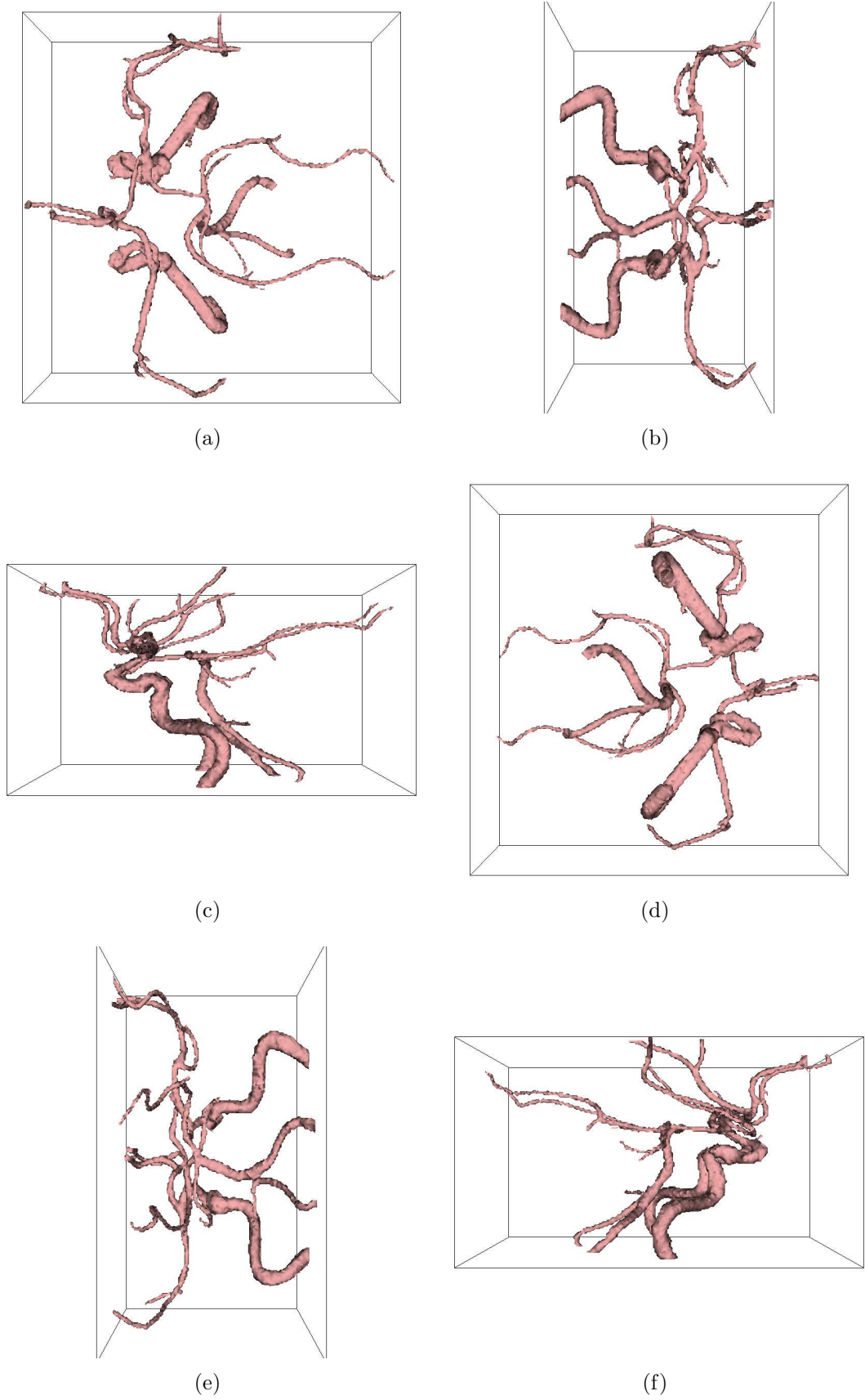


Figure 6.11: MRA segmentation results of the proposed CGAC algorithm with different view points.





Figure 6.12: MIP of the 3D MRA image B.

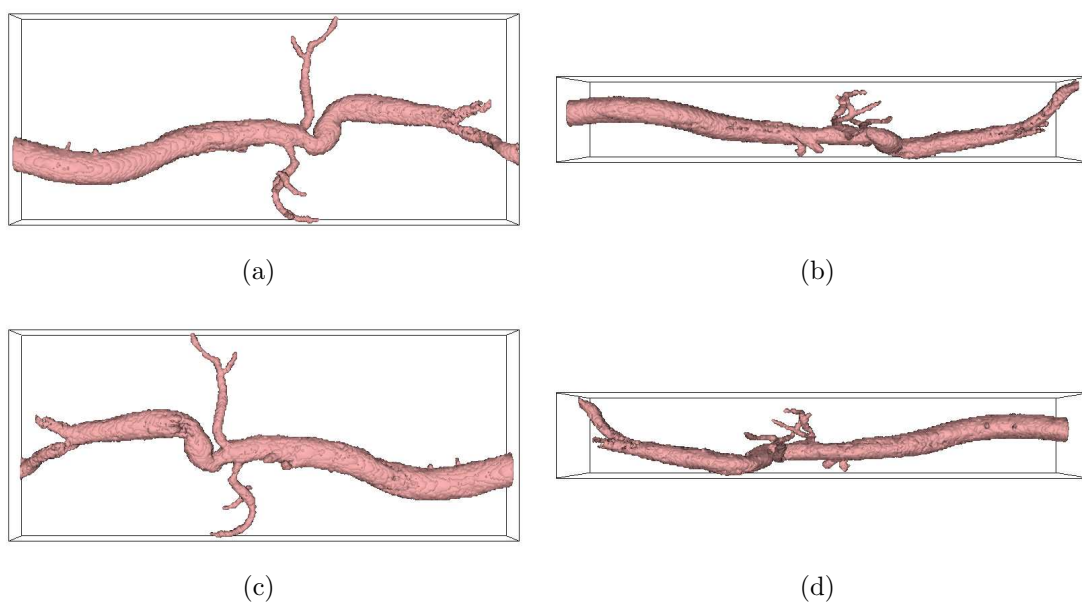


Figure 6.13: Segmentation results of MRA image B using the fuzzy connectedness method [2].

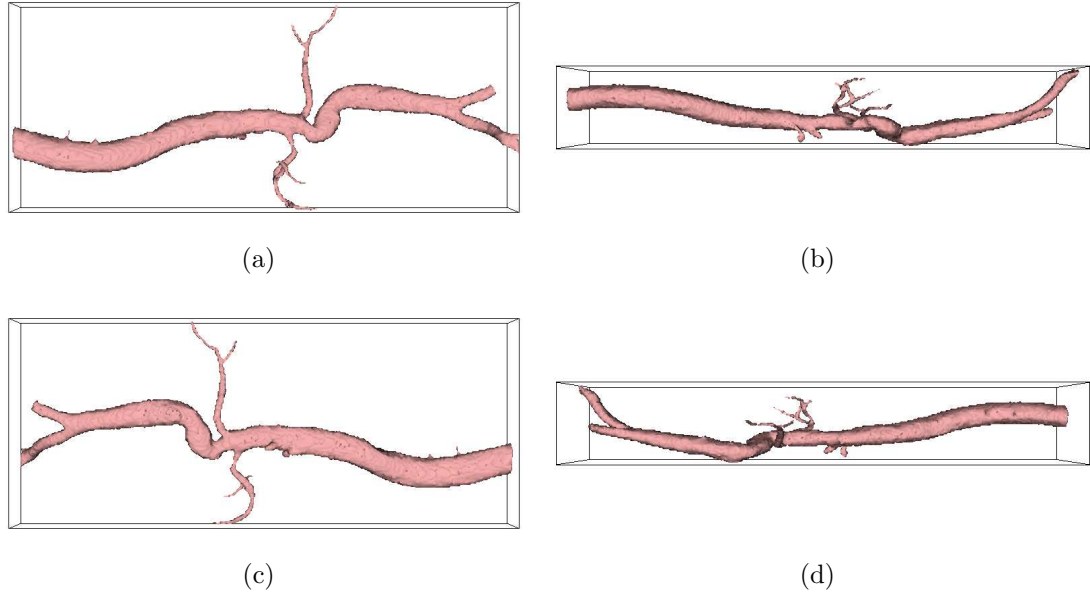


Figure 6.14: Segmentation results of MRA image B using the CGAC method.

the evolution equation makes the evolving surface keeping smooth, while there is no such a regularizer in fuzzy connectedness method.

The size of the MRA image C used in our experiments is  $384 \times 320 \times 64$ . The MIP is shown in Fig. 6.15. The segmentation results using our CGAC method are shown in Fig. 6.16. We can see that the whole vessel is extracted successfully.

The image D in our experiments is a 3D cerebral MRA image with the size of  $512 \times 512 \times 120$  and spacing  $0.43 \times 0.43 \times 1.2$ . The MIP of the image is shown in Fig. 6.17. The segmentation results using our CGAC method are shown in Fig. 6.18. It shows that the whole complex cerebral vasculature is successfully extracted.

## 6.5 Summary

In this chapter, we have proposed and applied the capillary geodesic active contour to both 3D synthetic and cerebral MRA images. The incorporated capillary force adapts the evolution surface into very thin branches of blood vessels and obtains more accurate segmentation as demonstrated in our experiments. Compared with

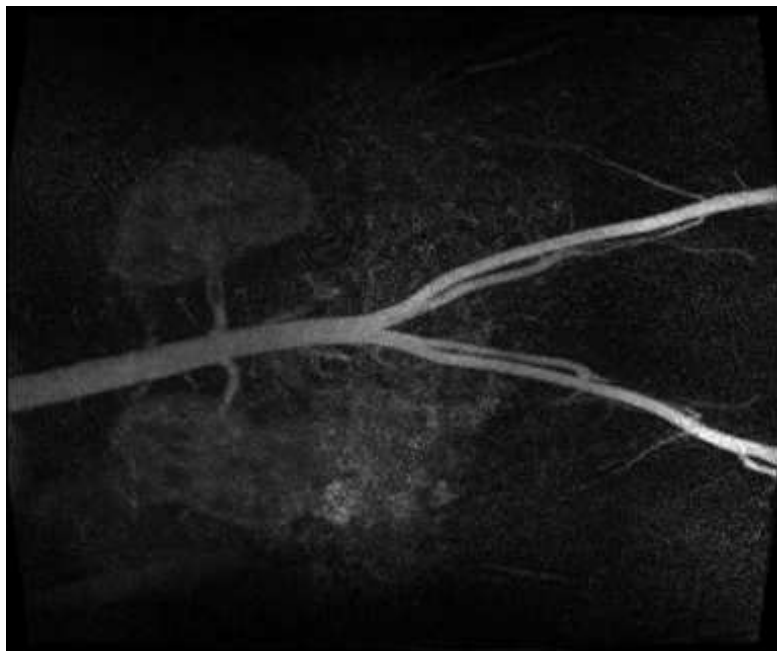


Figure 6.15: MIP of the MRA image C.

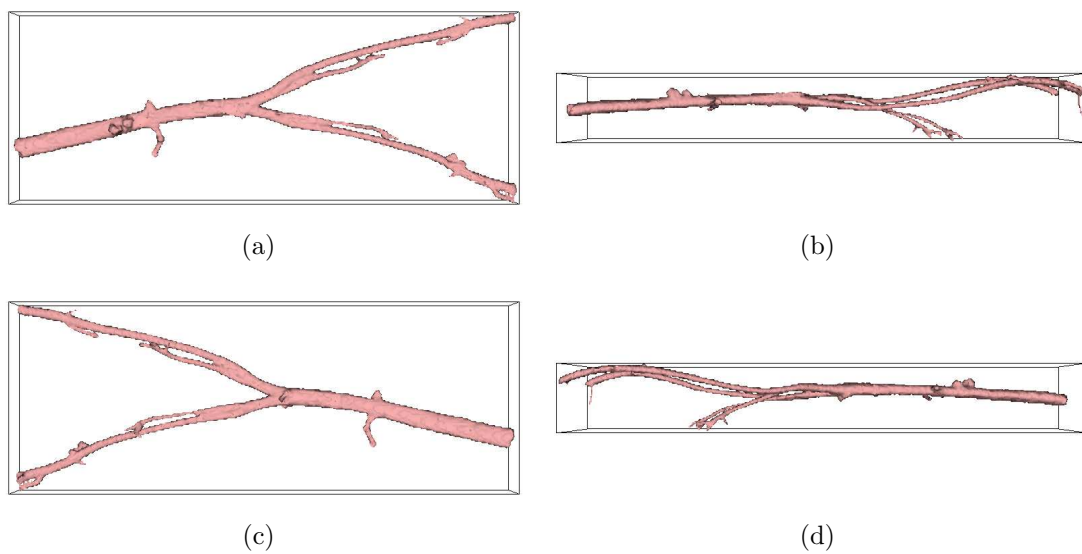


Figure 6.16: Segmentation results of the 3D MRA image C using our CGAC method.

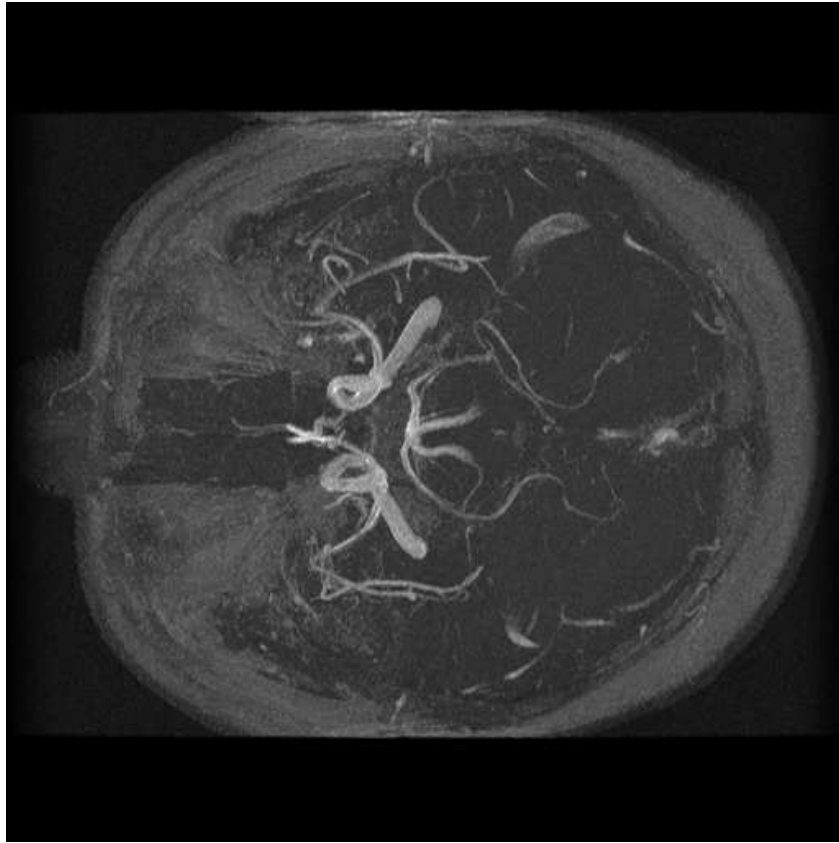
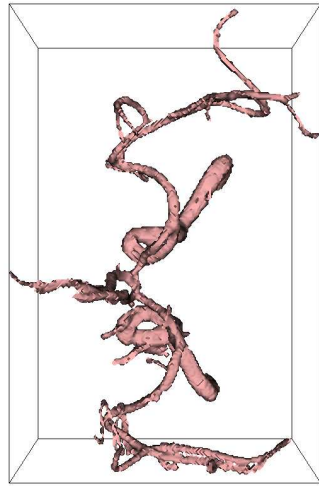
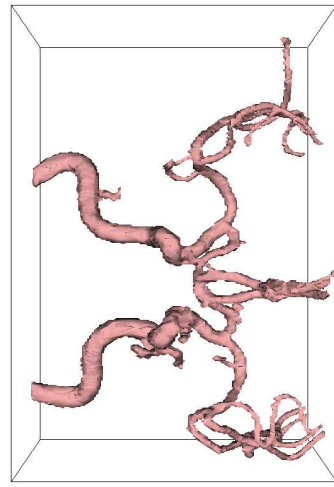


Figure 6.17: MIP of the MRA image D.

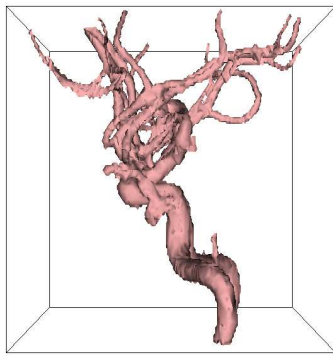
other techniques, the CGAC can achieve more details of vasculature. Our approach is geometric in nature and topology free due to that implicit representation of the evolving surfaces is used.



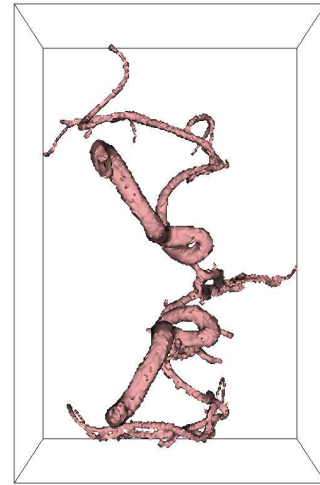
(a)



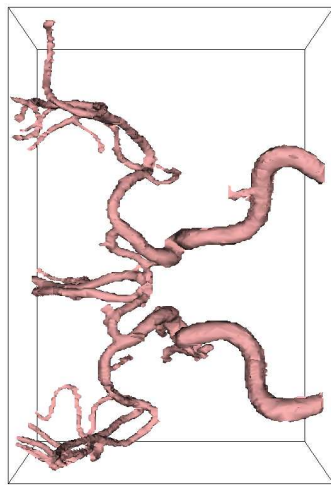
(b)



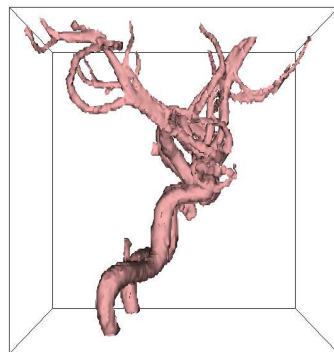
(c)



(d)



(e)



(f)

Figure 6.18: Segmentation results of cerebral MRA data set D using the CGAC.

# Chapter 7

## Conclusions

In this chapter, we conclude this dissertation with a review of our main research contributions, and propose directions for future research.

### 7.1 4D Medical Image Compression

In Chapter 3, we present a complete scheme for compressing 4D medical images. Although 2D and 3D medical image compression has been well studied, our scheme is the first one, to the best of our knowledge, to provide lossy-to-lossless 4D medical image compression. Since 4D image data is composed of a sequence of 3D volumetric images taken continuously over time, both spatial and temporal redundancies exist. Furthermore, lossless compression is usually required for medical image compression. Thus, a motion compensated lossy-to-lossless compression scheme is proposed to effectively and efficiently compress the 4D medical images. The key features of this work are summarized below.

- 3D motion estimation is employed in our algorithm to reduce the temporal redundancy among continuous 3D frames. A fast 3D cube matching algo-

rithm is proposed to reduce the heavy computational complexity caused by 3D motion estimation.

- Lossless compression is normally required for medical image compression, while lossy compression sometimes could be very useful for its high compression ratio. With lossy compression, fewer bits are needed to reconstruct an image with perceptual acceptable quality. Therefore, the progressive lossy-to-lossless compression is enabled in our scheme, which is very useful in applications like browsing database or previewing in telemedicine.
- Random access and progressive reconstruction of each 3D frame is enabled by developing a modified 3D-SPIHT algorithm. Therefore, there is no need to reconstruct the whole sequence for just viewing one of the frames.
- In the lossless compression mode, the proposed scheme can reduce the coding bit rates by more than 25% when compared with other state-of-the-art medical image compression algorithms. The PSNR values of each reconstructed slice of the proposed 4D compression scheme are normally 3–5dB higher than those of other algorithms in the lossy mode.

## 7.2 Medical Image Segmentation

Extracting organs from medical images is a key issue in many medical applications ranging from collecting clinical diagnosing information to surgery planning and navigating. In this dissertation, we present two novel medical image segmentation methods using deformable models. The main research contributions are summarized in this section.

### 7.2.1 Minimal Path Deformable Model

In Chapter 5, the minimal path deformable model incorporated with implicit prior shape information is used to extract object contour. Some key features of our algorithm are summarized as follows:

- Our minimal path deformable model greatly simplifies the initialization process of deformable models by selecting a single starting point.
- The algorithm employs a new energy minimization strategy, which obtains object contours by finding a minimal path. The paths are evaluated on a weighted graph map which consists of the weighted graph of the image and the weighted graph of the prior shape estimate. The shape estimate of the object of interest is incorporated into the segmentation process by representing it implicitly as zero level sets of the signed distance transform map of the estimated contour.
- An intelligent “worm” algorithm is proposed to find the minimal path. The worm is attracted to object contours by external force. Internal energy is defined for the worm to make it robust to noise and prevent it from twisting. The worm moves under the influence of both internal and external energy.
- Compared with the traditional parametric and geometric deformable models, the proposed model has lower computational complexity by avoiding evolving the whole curve at each step. With the implicit prior shape model, the segmentation algorithm is robust and accurate.
- The method has been validated using both simulated images and real medical images.



### 7.2.2 Capillary Geodesic Active Contour

In Chapter 6, the capillary geodesic active contour is proposed and developed to segment MRA images. Capillary action is incorporated into the segmentation process to extract thin vessels of vasculature. Our main research contributions are summarized as follows.

- In this algorithm, a new energy functional is formulated by modeling the capillary action, which is described in mathematics as a energy minimization process. Minimizing this energy leads to a new deformable model, capillary geodesic active contour.
- The CGAC is implemented using level set theory for its free topology adaptability. Compared with the speed functions of other level set based methods, there is a new force term which comes from the capillary force. This capillary force can help the curve/surface evolve into very thin blood vessels.
- Our model is derived directly from the capillary action using physics and mathematics. Contrary to existing MRA segmentation techniques, which have difficulty on extracting tiny vessels, the CGAC is able to segment the whole structure successfully from 3D MRA images with the help of capillary force.
- The effect of capillary force is clearly demonstrated by the experiments on synthetic images. The method is applied on real MRA images and better results have been obtained compared with other state-of-the-art MRA segmentation methods.

## 7.3 Future Work

This dissertation opens up some interesting directions for further investigation. We describe some of them in the following sections.

### 7.3.1 Object Based Coding

In Chapter 3, we have presented a motion compensated lossy-to-lossless 4D medical image compression scheme. In our current scheme, the whole data set, including some less important background noise, is encoded losslessly. It is possible to further reduce the overall coding bit rates, if only the objects of interest are encoded losslessly while other background information is encoded with lossy coding techniques. With our research work done on medical image segmentation, extracting objects of interest before coding becomes possible. Thus, lossless object based 4D medical image coding will be part of our future research work. However, the object based coding arises a new problem on how to efficiently encode an object with irregular shape. This is a relatively new research topic, although extensive research has been done on rectangular image compression.

### 7.3.2 Vasculature Measurement

Our algorithm for MRA image segmentation is presented in Chapter 6, where the whole vasculature can be successfully extracted from 3D MRA images. However, this is only the first step to provide diagnostic information to achieve *computer aided diagnosis* (CAD). MRA imaging is mainly used in diagnosing the possible diseases of blood vessels. For example, cerebral MRA imaging is mainly used to find the possibility of stroke or aneurysm. These possibilities are measured through checking the degree of stenosis or dilation along blood vessels. Since the whole

vasculature has been extracted out, this information could be ready for measuring. The degree of stenosis or dilation can be checked by computing the diameters of the blood vessels at each point. Nevertheless, due to the complex topologies of vascular structures, efforts need to be made on working out a robust and precise numerical measurement.

### 7.3.3 Medical Image Segmentation with Prior Knowledge

Prior shape information plays a key role in many computer vision and image processing applications, especially in medical image analysis where anatomic structures can be identified and classified in terms of their unique shapes. In order to further improve the precision and robustness of the proposed segmentation algorithms, anatomic knowledge of the objects to be extracted can be incorporated into the segmentation process. The prior knowledge may act like regularizers in the segmentation algorithms. It would be interesting to compare two possible choices to add these regularizers: using statistical models and detecting anatomic landmarks.

The statistical models can influence segmentation by estimating shape of the target object according to image information and/or current segmentation status. To construct a statistical model, large number of training data sets are required. These training data sets should be able to cover all the possible variety of the target object. This method is effective for segmenting objects with simple topologies. Some work based on this idea has been done in Chapter 5, where a prior shape model is constructed for each object and incorporated into the segmentation process using implicit representation. It is possible to further improve the performance of the algorithm by considering multiple models simultaneously if there is more than one object to be segmented. For example, a joint model can be constructed for the endocardium wall and the epicardium wall when extracting them from the CT cardiac images. The associated problem is how to effectively derive the probability density function and the corresponding MAP shape estimation.

In future, we would also like to investigate the landmark-based shape deformation methods for medical image segmentation. If the topology of the object is complex, like the cerebral vasculature, both training and estimating statistical models become very difficult. Then landmarks could be used to provide critical navigating information during segmentation. Currently, landmarks are mainly used in medical image registration. Some works have been done on using landmarks in medical image segmentation, however, there are still two key issues to be addressed: (a) identification of landmark points from given input images, and (b) regularization on the evolution of deformable models using landmarks.

As we have discussed, both these methods to incorporate prior shape knowledge into segmentation process have their own pros and cons. More research works are needed to gain further insight into these approaches. More importantly, the proposed algorithms need to be verified and validated with real medical images.

# Appendix A

## Deriving Level Set Evolution Equation of CGAC

The derivation process of level set evolution equation (6.27) of CGAC from the speed function (6.25) using level set theory is presented in this section.

Consider a surface  $S$  evolving according to

$$S_t = F\vec{N}, \quad (\text{A.1})$$

for a given speed function  $F$ . By using the level set theory, surface  $S$  is represented as the zero level set of a map function  $\Psi : \mathbb{R}^3 \rightarrow \mathbb{R}$ . Thus, we have a definition

$$\{S \in \mathbb{R}^3 : \Psi(S, t) = 0\}. \quad (\text{A.2})$$

It is reasonable to assume that  $\Psi$  is negative in the interior of the zero level set and positive in its exterior (usually, the signed distance function is used). Our problem is to find out how to update the function  $\Psi$ . By differentiating the definition above with respect to  $t$ , we obtain

$$\nabla \Psi \cdot S_t + \Psi_t = 0. \quad (\text{A.3})$$

Eqn. (A.3) can also be represented as

$$\Psi_t = -\nabla\Psi \cdot S_t. \quad (\text{A.4})$$

The we replace  $S_t$  in above equation by using Eqn. (6.25) to obtain the level set evolution equation. For convenience, Eqn. (6.25) is divided into three parts as

$$S_t = g(\kappa + c)\vec{\mathcal{N}} \quad (\text{A.5})$$

$$-(\nabla g \cdot \vec{\mathcal{N}})\vec{\mathcal{N}} \quad (\text{A.6})$$

$$+\alpha(1 + \lambda\hat{\kappa}_2) \left( \vec{\mathcal{N}} - \frac{\nabla g}{|\nabla g|} \cos \theta \right). \quad (\text{A.7})$$

Note that for any level set, the following relation holds:

$$\vec{\mathcal{N}} = -\frac{\nabla\Psi}{|\nabla\Psi|}. \quad (\text{A.8})$$

Thus, from Eqn. (A.5) we have

$$\begin{aligned} \Psi_{tI} &= g(\kappa + c) \left( -\frac{\nabla\Psi}{|\nabla\Psi|} \right) \cdot (-\nabla\Psi) \\ &= g(\kappa + c) |\nabla\Psi|. \end{aligned} \quad (\text{A.9})$$

Similarly, we obtain following expression from Eqn. (A.6):

$$\Psi_{tII} = \nabla g \cdot \nabla\Psi. \quad (\text{A.10})$$

Recall that  $\cos \theta$  in Eqn. (A.7) is defined as

$$\cos \theta = \vec{\mathcal{N}} \cdot \frac{\nabla g}{|\nabla g|}. \quad (\text{A.11})$$

We have

$$\begin{aligned}\Psi_{t\text{III}} &= \alpha(1 + \lambda\hat{\kappa}_2) \left( \left( -\frac{\nabla\Psi}{|\nabla\Psi|} \right) \cdot (-\nabla\Psi) - \frac{\nabla g}{|\nabla g|} \frac{\nabla g}{|\nabla g|} \left( -\frac{\nabla\Psi}{|\nabla\Psi|} \right) \cdot (-\nabla\Psi) \right) \\ &= \alpha(1 + \lambda\hat{\kappa}_2) |\nabla\Psi| (1 - \cos^2 \theta),\end{aligned}\tag{A.12}$$

where  $\cos \theta$  is redefined as

$$\cos \theta = \frac{\nabla\Psi \cdot \nabla g}{|\nabla\Psi| |\nabla g|}.\tag{A.13}$$

Finally, we combine the three parts  $\Psi_{t\text{I}}$ ,  $\Psi_{t\text{II}}$  and  $\Psi_{t\text{III}}$  together to get the evolution equation of CGAC

$$\begin{aligned}\Psi_t &= \Psi_{t\text{I}} + \Psi_{t\text{II}} + \Psi_{t\text{III}} \\ &= g(\kappa + c) |\nabla\Psi| + \nabla g \cdot \nabla\Psi + \alpha(1 + \lambda\hat{\kappa}_2) |\nabla\Psi| (1 - \cos^2 \theta),\end{aligned}\tag{A.14}$$

which is exactly the Eqn. (6.27).

# Bibliography

- [1] E. L. Ritman, R. A. Robb, and L. D. Harris, *Imaging Physiologic Functions: Experience with the Dynamic Spatial Reconstructor*. New York: Praeger, 1985.
- [2] J. K. Udupa, P. K. Saha, and R. A. Lotufo, “Relative fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1485–1500, Nov. 2002.
- [3] Z. Xiong, X. Wu, S. Cheng, and J. Hua, “Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms,” *IEEE Trans. Medical Imaging*, vol. 22, pp. 459–470, Mar. 2003.
- [4] N. D. Memon and X. Wu, “Recent developments in context-based predictive techniques for lossless image compression,” *The Computer Journal*, vol. 40, no. 2/3, pp. 127–136, 1997.
- [5] Y. Kim and W. A. Pearlman, “Lossless volumetric medical image compression,” in *Proc. SPIE*, vol. 3808, 1999, pp. 305–312.
- [6] L. Zeng, C. P. Jansen, S. Marsch, M. Unser, and P. R. Hunziker, “Four-dimensional wavelet compression of arbitrarily sized echocardiographic data,” *IEEE Trans. Med. Imag.*, vol. 21, pp. 1179–1187, Sept. 2002.



- [7] I. Peter and W. Straber, "The wavelet stream - progressive transmission of compressed light field data," in *IEEE Visualization*, San Francisco, CA, USA, Oct. 1999, pp. 69–72.
- [8] S. M. Lawrie and S. S. Abukmeil, "Brain abnormality in schizophrenia: A systematic and quantitative review of volumetric magnetic resonance imaging studies," *British Journal of Psychiatry*, vol. 172, pp. 110–120, Feb. 1998.
- [9] P. Taylor, "Computer aids for decision-making in diagnostic radiology—a literature review," *British Journal of Radiology*, vol. 68, no. 813, pp. 945–957, 1995.
- [10] A. P. Zijdenbos and B. M. Dawant, "Brain segmentation and white matter lesion detection in MR images," *Critical Reviews in Biomedical Engineering*, vol. 22, no. 6, pp. 401–465, 1994.
- [11] A. Worth, N. Makris, V. Caviness, and D. Kennedy, "Neuroanatomical segmentation in MRI: technological objectives," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, pp. 1161–1187, 1997.
- [12] V. Khoo, D. P. Dearnaley, D. J. Finnigan, A. Padhani, S. F. Tanner, and M. O. Leach, "Magnetic resonance imaging (MRI): considerations and applications in radiotherapy treatment planning," *Radiother Oncol*, vol. 42, pp. 1–15, 1997.
- [13] H. Muller-Gartner, J. Links, J. Prince, R. Bryan, E. McVeigh, J.P.Leal, C. Davatzikos, and J. Frost, "Measurement of tracer concentration in brain gray matter using positron emission tomography: MRI-based correction for partial volume effects," *J. Cerebral Blood Flow and Metabolism*, vol. 12, no. 4, pp. 571–583, 1992.
- [14] N. Ayache, P. Cinquin, I. Cohen, L. Cohen, F. Leitner, and O. Monga, "Segmentation of complex three-dimensional medical objects: A challenge and a requirement for computer-assisted surgery planning and performance," in

- Computer-Integrated Surgery*, R. Taylor, S. Lavallee, G. Burdea, and R. Mosges, Eds. The MIT Press, 1996, ch. 4, pp. 59–74.
- [15] W. E. L. Grimson, G. J. Ettinger, T. Kapur, M. E. Leventon, W. M. Wells, and R. Kikinis, “Utilizing segmented MRI data in image-guided surgery,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 8, pp. 1367–1397, 1997.
- [16] P. Yan and A. A. Kassim, “Medical image segmentation with minimal path deformable models,” in *IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 2733–2736.
- [17] A. A. Kassim, P. Yan, W. S. Lee, and K. Sengupta, “Motion compensated lossy-to-lossless compression of 4D medical images using integer wavelet transforms,” *IEEE Trans. Inform. Technol. Biomed.*, vol. 9, no. 1, pp. 132–138, Mar. 2005.
- [18] G. Menegaz and J.-P. Thiran, “Three-dimensional encoding/two-dimensional decoding of medical data,” *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 424–440, March 2003.
- [19] J. Wang and K. Huang, “Medical image compression by using three-dimensional wavelet transformation,” *IEEE Trans. Med. Imag.*, vol. 15, pp. 547–554, Aug. 1996.
- [20] W.-J. Hwang, C.-F. Chine, and K.-J. Li, “Scalable medical data compression and transmission using wavelet transform for telemedicine applications,” *IEEE Trans. Inform. Technol. Biomed.*, vol. 7, pp. 54–63, Mar. 2003.
- [21] A. Bilgin, G. Zweig, and M. W. Marcellin, “Three-dimensional image compression with integer wavelet transforms,” *Appl. Opt.: Inform. Proc.*, vol. 39, pp. 1799–1814, Apr. 2000.

- [22] P. Yan and A. A. Kassim, “Lossless and near-lossless motion-compensated 4D medical image compression,” in *IEEE Int. Workshop on BioMedical Circuits and Systems*, Singapore, Dec. 2004.
- [23] M. J. Weinberger, G. Seroussi, and G. Shapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Trans. Image Processing*, vol. 9, pp. 1309–1324, 2000.
- [24] X. Wu and N. D. Memon, “Context-based adaptive lossless image coding,” *IEEE Trans. Commun.*, vol. 45, pp. 437–444, 1997.
- [25] “WG1N1523 JPEG 2000 Part I Committee Draft Version 1.0,” ISO/IEC JTC1/SC29/WG1, 1999.
- [26] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.
- [27] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, “Wavelet transforms that map integers to integers,” *J. Appl. Comput. Harmon. Anal.*, vol. 5, pp. 332–369, 1998.
- [28] J. M. Shapiro, “Embedded image coding using zerotrees of wavelets,” *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
- [29] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, 1996.
- [30] E. Chiu, J. Vaisey, and M. S. Atkins, “Wavelet-based space-frequency compression of ultrasound images,” *IEEE Trans. Inform. Technol. Biomed.*, vol. 5, pp. 300–310, Dec. 2001.

- [31] G. Menegaz and J.-P. Thiran, "Lossy to lossless object-based coding of 3D MRI data," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 1053–1061, Sept. 2002.
- [32] I. Daubechies and W. Sweldens, "Factoring wavelet and subband transforms into lifting steps," *J. Fourier Anal. Application*, vol. 4, no. 3, pp. 245–267, 1998.
- [33] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Proc. Magz.*, vol. 14, pp. 82–100, Sept. 1997.
- [34] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, pp. 438–442, 1994.
- [35] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search for block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, pp. 369–377, Aug. 1998.
- [36] A. A. Kassim and W. S. Lee, "Embedded color image coding using spiht with modified spatial orientation trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 203–206, Feb. 2003.
- [37] B. Kim, Z. Xiong, and W. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374–1387, 2000.
- [38] J. Reichel, G. Menegaz, M. Nadenau, and M. Kunt, "Integer wavelet transform for embedded lossy to lossless image compression," *IEEE Trans. Image Processing*, vol. 10, no. 3, pp. 383–392, March 2001.
- [39] M. D. Adams and F. Kossentini, "Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis," *IEEE Trans. Image Processing*, vol. 9, pp. 1010–1024, June 2000.

- [40] A. Signoroni and R. Leonardi, “Modeling and reduction of PSNR fluctuations in 3D wavelet coding,” *ICIP’01*, vol. 3, pp. 812–815, Oct. 2001.
- [41] N. Uzun and R. A. Haddad, “Cyclostationary modeling, analysis, and optimal compensation of quantization errors in subband codecs,” *IEEE Trans. Image Processing*, vol. 43, pp. 2109–2119, Sept. 1995.
- [42] M. Iwahashi, Y. Tonomura, S. Chokchaitam, and N. Kambayashi, “Pre-post quantization and integer wavelet for image compression,” *Electronics Letters*, vol. 39, no. 24, pp. 1725–1726, Nov. 2003.
- [43] T. McInerney and D. Terzopoulos, “Deformable models in medical image analysis: a survey,” *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, 1996.
- [44] J. S. Duncan and N. Ayache, “Medical image analysis: Progress over two decades and the challenges ahead,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 85–106, 2000.
- [45] J. Suri, K. Liu, S. Singh, S. Laxminarayana, and L. Reden, “Shape recovery algorithms using level sets in 2-D/3-D medical imagery: A state-of-the-art review,” *IEEE Trans. Information Technology in Biomedicine*, vol. 6, pp. 8–28, Mar. 2002.
- [46] C. Xu and J. L. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Trans. Image Processing*, vol. 7, pp. 359–369, Mar. 1998.
- [47] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.
- [48] S. W. Zucker, “Survey: Region growing: Childhood and adolescence,” *Computer Vision, Graphics and Image Processing*, vol. 5, pp. 382–399, 1976.
- [49] S. Beucher, “Watersheds of functions and picture segmentation,” in *ICASSP82*, 1982, pp. 1928–1931.

- [50] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Upper Saddle River, New Jersey: Prentice Hall, 2001.
- [51] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.
- [52] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.
- [53] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [54] S. Kichenassamy, A. Kumar, P. J. Olver, A. Tannenbaum, and A. J. Yezzi, “Gradient flows and geometric active contour models,” in *IEEE Int. Conf. Computer Vision*, 1995, pp. 810–815.
- [55] L. D. Cohen, “On active contour models and balloons,” *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.
- [56] L. H. Staib and J. S. Duncan, “Parametrically deformable contour models,” in *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 1989, pp. 98–103.
- [57] J. W. Snell, M. B. Merickel, J. M. Ortega, J. C. Goble, J. R. Brookeman, and N. F. Kassell, “Model-based boundary estimation of complex objects using hierarchical active surface templates,” *Pattern Recognition*, vol. 28, no. 10, pp. 1599–1609, 1995.
- [58] I. Cohen, L. Cohen, and N. Ayache, “Using deformable surfaces to segment 3D images and infer differential structures,” *CVGIP: Image Understanding*, vol. 56, no. 2, pp. 242–263, 1992.
- [59] L. H. Staib and J. S. Duncan, “Model-based deformable surface finding for medical images,” *IEEE Trans. Medical Imaging*, vol. 15, no. 5, pp. 720–731, Oct. 1996.

- [60] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *J. Computational Physics*, vol. 79, pp. 12–49, 1988.
- [61] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed. New York: Cambridge University Press, 1999.
- [62] V. Caselles, F. Catte, T. Coll, and F. Dibos, “A geometric model for active contours,” *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.
- [63] R. Malladi, J. A. Sethian, and B. C. Vermuri, “Shape modeling with front propagation: A level set approach,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–174, 1995.
- [64] J. S. Suri, K. Liu, L. Reden, and S. Laxminarayan, “A review on MR vascular image processing: skeleton versus nonskeleton approaches: part II,” *IEEE Trans. Information Technology in Biomedicine*, vol. 6, no. 4, pp. 338–350, Dec. 2002.
- [65] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
- [66] L. D. Cohen and R. Kimmel, “Global minimum for active contour models: a minimal path approach,” *International Journal of Computer Vision*, vol. 24, pp. 57–78, 1997.
- [67] L. D. Cohen and T. Deschamps, “Grouping connected components using minimal path techniques,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 102–109.
- [68] C. Han, T. S. Hatsukami, J.-N. Hwang, and C. Yuan, “A fast minimal path active contour model,” *IEEE Trans. Image Processing*, vol. 10, pp. 865–873, June 2001.

- [69] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” *Proc. National Academy of Sciences*, vol. 93, pp. 1591–1595, 1996.
- [70] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, “Minimal surfaces based object segmentation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 394–398, Apr. 1997.
- [71] K. Siddiqi, Y. B. Lauziere, A. Tannenbaum, and S. W. Zucker, “Area and length minimizing flows for shape segmentation,” *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 433–443, Mar. 1998.
- [72] N. Paragios and R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, Mar. 2000.
- [73] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*, 3rd ed. Kitware Inc., 2002.
- [74] H. E. Cline, C. L. Dumoulin, W. E. Lorensen, S. P. Souza, and W. J. Adams, “Volume rendering and connectivity algorithms for MR angiography,” *Magnetic Resonance in Medicine*, vol. 18, pp. 384–394, 1991.
- [75] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models – their training and application,” *Comput. Vision Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.
- [76] T. F. Cootes, C. Beeston, G. J. Edwards, and C. J. Taylor, “A unified framework for atlas matching using active appearance models,” in *Proc. Int. Conf. on Image Processing in Medical Imaging*, 1999, pp. 322–333.
- [77] L. H. Staib and J. S. Duncan, “Boundary finding with parametrically deformable models,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, 1992.



- [78] M. Leventon, W. Grimson, and O. Faugeras, “Statistical shape influence in geodesic active contours,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2000, pp. 316–323.
- [79] Y. Chen, S. Thiruvankadam, H. D. Tagare, F. Huang, D. Wilson, and E. A. Geiser, “On the incorporation of shape priors into geometric active contours,” in *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001, pp. 145–152.
- [80] M. Rousson, N. Paragios, and R. Deriche, “Implicit active shape models for 3D segmentation in MR imaging,” in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, C. Barillot, D. R. Haynor, and P. Hellier, Eds., vol. 3216, 2004, pp. 209–216.
- [81] J. Xie, Y. Jiang, and H. T. Tsui, “Segmentation of kidney from ultrasound images based on texture and shape priors,” *IEEE Trans. Medical Imaging*, vol. 24, no. 1, pp. 45–57, 2005.
- [82] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [83] J. K. Udupa, P. K. Saha, and R. A. Lotufo, “Boundary detection via dynamic programming,” *SPIE Proc. Medical Imaging 1992*, vol. 1808, pp. 33–39, 1992.
- [84] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, “Linear time euclidean distance transform algorithms,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 529–533, May 1995.
- [85] A. A. Amini, T. E. Weymouth, and R. C. Jain, “Using dynamic programming for solving variational problems in vision,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 855–866, 1990.
- [86] C. Cocosco, V. Kollokian, R.-S. Kwan, and A. Evans, “BrainWeb: Online interface to a 3D MRI simulated brain database,” in *Proc. 3rd Int. Conf.*

- Functional Mapping of the Human Brain*, vol. 5, Copenhagen, May 1997, p. 425.
- [87] W. E. Higgins, W. J. T. Spyra, E. L. Ritman, Y. Kim, and F. A. Spelman, "Automatic extraction of the arterial tree from 3-D angiograms," in *IEEE Conf. Eng. in Medicine and Bio.*, vol. 2, Nov. 1989, pp. 563–564.
- [88] N. Niki, Y. Kawata, H. Satoh, and T. Kumazaki, "3D imaging of blood vessels using X-ray rotational angiographic system," in *Nuclear Science Symposium and Medical Imaging Conference*, vol. 3, San Francisco, CA, USA, 1993, pp. 1873–1877.
- [89] C. Molina, G. P. Prause, P. Radeva, and M. Sonka, "3-D catheter path reconstruction from biplane angiography using 3D snakes," in *SPIE - Medical Imaging*, San Diego, California, 1998, pp. 441–444.
- [90] A. K. Klein, F. Lee, and A. A. Amini, "Quantitative coronary angiography with deformable spline models," *IEEE Trans. Medical Imaging*, vol. 16, pp. 468–482, Oct. 1997.
- [91] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, "3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical Image Analysis*, vol. 2, no. 2, pp. 143–168, 1998.
- [92] D. Guo and P. Richardson, "Automatic vessel extraction from angiogram images," in *Computers in Cardiology*, Cleveland, OH, USA, Sept. 1998, pp. 441–444.
- [93] M. M. Orkisz, C. Bresson, I. E. Magnin, O. Champin, and P. C. Douek, "Improved vessel visualization in MR angiography by nonlinear anisotropic filtering," *Magnetic Resonance in Medicine*, vol. 37, pp. 914–919, 1997.
- [94] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. van Walsum, and M. A. Viergever, "Model-based quantitation of 3-D magnetic resonance angio-

- graphic images,” *IEEE Trans. Medical Imaging*, vol. 18, pp. 946–956, Oct. 1999.
- [95] O. Wink, W. J. Niessen, and M. A. Viergever, “Fast delineation and visualization of vessels in 3-D angiographic images,” *IEEE Trans. Medical Imaging*, vol. 19, no. 4, pp. 337–346, Apr. 2000.
- [96] P. J. Yim, P. L. Choyke, and R. M. Summers, “Gray-scale skeletonization of small vessels in magnetic resonance angiography,” *IEEE Trans. Medical Imaging*, vol. 19, no. 6, pp. 568–576, June 2000.
- [97] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Troussel, “Model based detection of tubular structures in 3D images,” *Computer Vision and Image Understanding*, vol. 80, no. 2, pp. 130–171, Nov. 2000.
- [98] S. R. Aylward and E. Bullitt, “Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction,” *IEEE Trans. Medical Imaging*, vol. 21, no. 2, pp. 61–75, Feb. 2002.
- [99] E. Sorantin, C. Halmi, B. Erdohelyi, K. Palagyi, L. Nyul, K. Olle, B. Geiger, F. Lindbichler, G. Friedrich, and K. Kiesler, “Spiral-CT-based assessment of tracheal stenoses using 3D skeletonization,” *IEEE Trans. Medical Imaging*, vol. 21, no. 3, pp. 263–273, Mar. 2002.
- [100] Q. Huang and G. C. Stockman, “Generalized tube model: recognizing 3D elongated objects from 2D intensity images,” in *IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY, USA: IEEE, jun 1993, pp. 104–109.
- [101] Y. Masutani, T. Kurihara, and M. Suzuki, “Quantitative vascular shape analysis for 3D MR-angiography using mathematical morphology,” in *Proc. Int. Conf. Computer Vision, Robotics and Virtual Reality in Medicine (IC-CVRMed)*, apr 1995, pp. 449–454.

- [102] J. Chen and A. A. Amini, “Quantifying 3-D vascular structures in MRA images using hybrid PDE and geometric deformable models,” *IEEE Trans. Medical Imaging*, vol. 23, no. 10, pp. 1251–1262, 2004.
- [103] P. J. Yim, G. Boudewijn, C. Vasbinder, V. B. Ho, and P. L. Choyke, “Iso-surfaces as deformable models for magnetic resonance angiography,” *IEEE Trans. Medical Imaging*, vol. 22, no. 7, pp. 875–881, July 2003.
- [104] L. M. Lorigo, O. Faugeras, W. E. L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin, “CURVES: Curve evolution for vessel segmentation,” *Medical Image Analysis*, vol. 5, pp. 195–206, 2001.
- [105] P. Yan, A. A. Kassim, S. C. Wang, and B. Shuter, “MRA image segmentation with capillary geodesic active contours,” submitted to, 2005.
- [106] F. K. H. Quek and C. Kirbas, “Vessel extraction in medical images by wave propagation and traceback,” *IEEE Trans. Medical Imaging*, vol. 20, no. 2, pp. 117–131, 2001.
- [107] P. J. H. de Koning, J. A. Schaap, J. P. Janssen, J. J. M. Westenberg, R. J. van der Geest, and J. H. C. Reiber, “Automated segmentation and analysis of vascular structures in magnetic resonance angiographic images,” *Magnetic Resonance in Medicine*, vol. 50, pp. 1189–1198, 2003.
- [108] J. Montagnat, H. Delingette, and N. Ayache, “A review of deformable surfaces: topology, geometry and deformation,” *Image and Vision Computing*, vol. 19, no. 14, pp. 1023–1040, Dec. 2001.
- [109] T. McInerney and D. Terzopoulos, “T-Snakes: Topology adaptive snakes,” *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, June 2000.
- [110] M. Descoteaux, L. Collins, and K. Siddiqi, “Geometric flows for segmenting vasculature in MRI: theory and validation,” in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Saint-Malo, France, 2004, pp. 500–507.

- [111] A. Vasilevskiy and K. Siddiqi, “Flux maximizing geometric flows,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1565–1578, Dec. 2002.
- [112] L. M. Lorigo, O. Faugeras, W. E. L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.-F. Westin, “Codimension-two geodesic active contours for the segmentation of tubular structures,” in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 444–451.
- [113] L. Ambrosio and H. M. Soner, “Level set approach to mean curvature flow in arbitrary codimension,” *J. Differential Geometry*, vol. 43, pp. 693–737, 1996.
- [114] C. Kirbas and F. Quek, “A review of vessel extraction techniques and algorithms,” *ACM Computing Surveys*, vol. 36, no. 2, pp. 81–121, 2004.
- [115] R. Finn, *Equilibrium Capillary Surfaces*. New York: Springer-Verlag, 1986.
- [116] M. J. Forray, *Variational calculus in science and engineering*. New York: McGraw-Hill, 1968.
- [117] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*. Kitware Inc., 2003. [Online]. Available: <http://www.itk.org>

# List of Publications

- 1) **P. Yan** and A. A. Kassim, “MRA image segmentation with capillary active contours,” *Medical Image Analysis*, vol. 10, no. 3, June, pp. 317–329, 2006.
- 2) **P. Yan** and A. A. Kassim, “Medical image segmentation using minimal path deformable models with implicit shape priors,” *IEEE Trans. Information Technology in Biomedicine*, to appear, 2006.
- 3) **P. Yan**, W. Shen, A. A. Kassim, and M. Shah, “Modeling Interaction for Segmentation of Neighboring Structures,” submitted to *IEEE Trans. Medical Imaging*, 2005.
- 4) **P. Yan**, W. Shen, A. A. Kassim, and M. Shah, “Segmentation of neighboring organs in medical image with model competition,” In Proceedings of *Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, vol. 1, pp. 270–277, 2005.
- 5) **P. Yan** and A. A. Kassim, “MRA image segmentation with capillary active contour,” In Proceedings of *Int. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, vol. 1, pp. 51–58, 2005. (**MICCAI 2005 NDI Student Award**)
- 6) A. A. Kassim, **P. Yan**, W. S. Lee, and K. Sengupta, “Motion compensated lossy-to-lossless compression of 4D medical images using integer wavelet transforms,” *IEEE Trans. Information Technology in Biomedicine*, vol. 9, no. 1, pp. 132–138, Mar. 2005.

- 7) **P. Yan** and A. A. Kassim, “Lossless and near-lossless motion-compensated 4D medical image compression,” in *IEEE Int. Workshop on BioMedical Circuits and Systems*, Singapore, Dec. 2004.
- 8) **P. Yan** and A. A. Kassim, “Medical image segmentation with minimal path deformable models,” in *IEEE Int. Conf. Image Processing (ICIP)*, Singapore, Oct. 2004, pp. 2733–2736.