MODEL-BASED IMAGE SEGMENTATION IN MEDICAL APPLICATIONS

BY ZHEN QIAN

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Computer Science

Written under the direction of

Dimitris N. Metaxas

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE THESIS

Model-Based Image Segmentation in Medical Applications

by Zhen Qian

Thesis Director: Dimitris N. Metaxas

Image segmentation is an essential and indispensable step in medical image analysis. It partitions the image into meaningful anatomic or pathological structures. Because medical image segmentation needs high level medical and anatomic knowledge, model-based segmentation methods are highly desirable. In this thesis, we will first give a short survey of current approaches of medical image segmentation. Then we specifically develop appearance and shape models for different segmentation tasks. These models are either obtained from visual observation and prior human expertise, or from certain automatic machine learning methods. In this thesis, two model-based image segmentation algorithms are developed for 3D MR colonography and 2D cardiac tagged MRI. For 3D MR colonography, we manually build the shape and intensity model. For 2D tagged MRI, we learn the shape and local appearance model from a training set. In each application, besides the models, we give complete details in solving the segmentation problems, such as how we correct the MR image intensity inhomogeneity and how we automatically initialize the segmentation. Both model-based methods perform well on real medical image data.

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Dimitris Metaxas for his continuous support and guidance in both academic and daily life ever since my first day at Rutgers. I would also like to thank my co-advisor Prof. Leon Axel in the BME program. His insights and suggestions contributed greatly to this thesis.

Many thanks to my thesis committee members, Profs. Vladimir Pavlovic and Ahmed Elgammal. The theories and skills that I have learned from their enlightening Computer Vision and Artificial Intelligence classes helped me greatly in conducting this research. Their valuable advice and suggestions help me to complete this work.

I have joined CBIM for 5 years. I enjoyed a lot the close team working environment and friendly atmosphere in this lovely group. I am grateful to my former CBIM labmates, Xiaolei Huang, Rong Zhang and Kyoungju Park. During my first two years at Rutgers, I learned a lot from them. I appreciate the enlightening discussions with my current CBIM labmates. They are Rui Huang, Atul Kanaujia, Zhiguo Li, Viorel Mihalef, Jinghao Zhou, Suejung Huh, Chansu Lee, Xiaoxu Wang, Peng Yang, Junzhou Huang, Yuchi Huang, and Jintao Feng. Thanks to Ting Chen, Tushar Manglik, Jöel Schaerer and other members in the Radiology Department at NYU Medical Center. Thanks also to Dr. Matthias Wolf and Dr. Luca Bogoni from Siemens Medical Solutions. I had a fruitful internship experience there in summer 2006.

Last but not least, I thank my parents and my sister for always being there when I needed them most, and for supporting me through all the years.

Table of Contents

List of Figures

Chapter 1

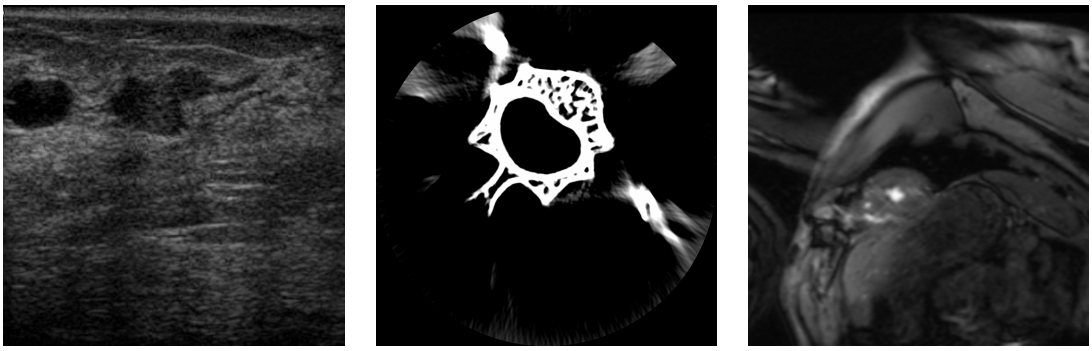
Introduction

1.1 Motivation

Image segmentation is a classic problem in computer vision. It is a process that partitions the image pixels into meaningful groups so that we can achieve a compact representation of the image [1]. Segmentation is usually performed based on many factors, such as intensity, color, or texture similarities, pixel continuity, and higher level knowledge about the objects model.

Image segmentation has many applications in the biomedical field. Medical imaging has become an important part in health care nowadays. X-rays, ultrasound, computer tomography (CT) and magnetic resonance (MR) images have been routine diagnostic methods that are performed in hospitals and clinics. See Figure 1.1 for some examples. A notable trend of medical imaging is its integrations with the state-of-art computer techniques. Computers are used to not only display, store, transfer, and manage medical image data, but also help doctors with image reading and analysis. Computer aided detection and diagnosis play important roles in lowering doctor's workload and increasing diagnostic accuracy. Therefore medical image analysis have become a keen research topic in image processing and computer vision.

In medical image analysis, image segmentation is an essential and indispensable step for many anatomy and pathology studies. Medical image segmentation can help clinicians 1, differentiate and visualize organs and tissues; 2, measure and compare the size of tissue or pathologies; and 3, plan surgery and other treatments. For example, in cardiac images, the accurate segmentation of the left ventricle endocardium is critical for estimating the ejection fraction, which is an important indicator of cardiac function. In colonography, segmentation is needed for 3D lumen reconstruction and polyp detection



(a) (b) (c)

Figure 1.1: (a) is an ultrasound breast image. The lesions are darker than the normal tissue. (b) is a CT image of a rat's vertebra. We can see calcium get highlighted in CT modality. (c) is a cardiac MR image. We find the heart sits in a complex background which consists of many other anatomical structures.

and size-measurement, which are critical for early colon cancer screening.

1.2 Problem Statement

Medical image segmentation distinguishes itself from conventional image segmentation tasks in the following aspects.

First, medical imaging techniques have limitations in fully revealing the clinical relevant information of the anatomy. In many cases, the tissue or organ of interest is difficult to be separated from its surroundings, when they share similar intensity levels with the other tissues, and their boundaries lack strong edge or ridge information. For example, CT technique is weak at imaging soft tissues. It cannot differentiae grey and white matters well in neural tissues.

Second, many imaging modalities, such as ultrasound and PET, generate very noisy and blurred images due to their intrinsic imaging mechanisms. At the same time, for patients' better acceptance, radiologists tend to lower radiation doses and shorten acquisition time on CT and MRI, which also results in low SNR images. In some special imaging techniques, such as tagged cardiac MRI, artificial markers or tags are added into the images, which may even make the segmentation tasks more difficult.

Third, besides of the image information, higher level knowledge of anatomy and

pathology is critical for medical image segmentation. Usually medical image has complex appearance due to the complicated anatomic structures. Medical expertise is required to understand and interpret the image, so that the segmentation algorithms could meet the clinicians' needs. Segmentation results should always be validated by clinicians as well.

This Master's thesis focuses on model-based image segmentation tasks in medical applications. We will first give a brief survey of current medical image segmentation methods. Then we will focus on two medical image segmentation tasks: 3D MR colonography segmentation and 2D cardiac tagged MR image segmentation. Since medical image segmentation has the previously stated characteristics, the segmentation methods need to be specific with respect to different problems. We will develop two model-based segmentation methods. One is a 3D shape modeling and tracking method for MR colonography, and the other is a 2D shape and local appearance modeling method for cardiac tagged MRI. These case-specific methods are more adaptive to certain medical applications.

1.3 Thesis Outline

The thesis is presented in four parts. The first part reviews the recent advances of image segmentation methods in medical applications (Chapter 2). In the second part we develop a 3D shape modeling and tracking approach for segmentation in tube-shaped structures. An implementation in colonography is given (Chapter 3). In the third part, we develop a shape and local appearance modeling method. An implementation in cardiac tagged MRI is given (Chapter 4). In the fourth part, we have a discussion and conclude this thesis (Chapter 5).

The thesis is organized as follows:

Chapter 1. Introduction: The background, motivation and structure of the thesis are provided.

Chapter 2. Related Work on Medical Image Segmentation: A survey of related medical image segmentation methods will be given, such as manual, threshold,

shape model based, edge based, region based and hybrid segmentation.

Chapter 3. 3D Colonography Segmentation: A model-based 3D segmentation method for MR colonography will be given.

Chapter 4. Segmentation in Cardiac Tagged MRI: A model-based 2D segmentation method for cardiac tagged MRI will be provided.

Chapter 5. Discussion and Conclusion: We will give a discussion and conclude this thesis.

Chapter 2

Related Work on Medical Image Segmentation

2.1 Manual Segmentation

Manual segmentation requests a user to manually trace the desired boundaries. Usually certain relevant hardware and software supporting user interactions are needed. It is a basic, but still very commonly used technique in medical image analysis. In many situations, manual segmentation is indispensable, especially when the current segmentation algorithms cannot reach the satisfied results. Also for machine learning purposes, the training data are often obtained from manual segmentation.

The pro of manual method is that the user can make full use of his or her medical knowledge. If we trust the user's expertise, then the manual results should be the most reliable. Actually in many cases, experts' results are treated as the ground truth. But on the other hand, manual segmentation is very time consuming. And the manual results tend to be inconsistent across different users. To solve this, averaging several users' results are encouraged.

To speed up the manual segmentation, a handy user interface is essential. We have developed a user interaction system that enables the user to freely move the points on the boundaries. A post-processing procedure to smooth the user defined boundary is optional. See figure 2.1 for an example of the manual user interface [2].

2.2 Threshold Methods

If the intensity level of the foreground objects are very distinctive with that of the background, threshold techniques could be a good choice for image segmentation. Threshold

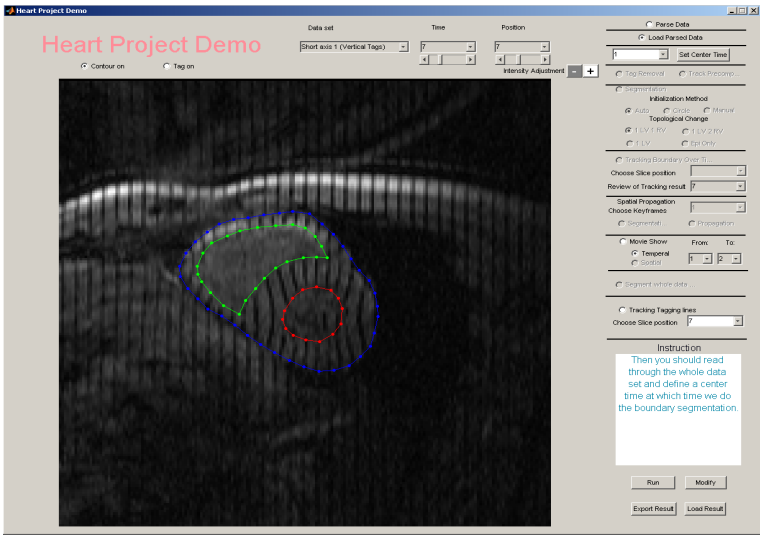


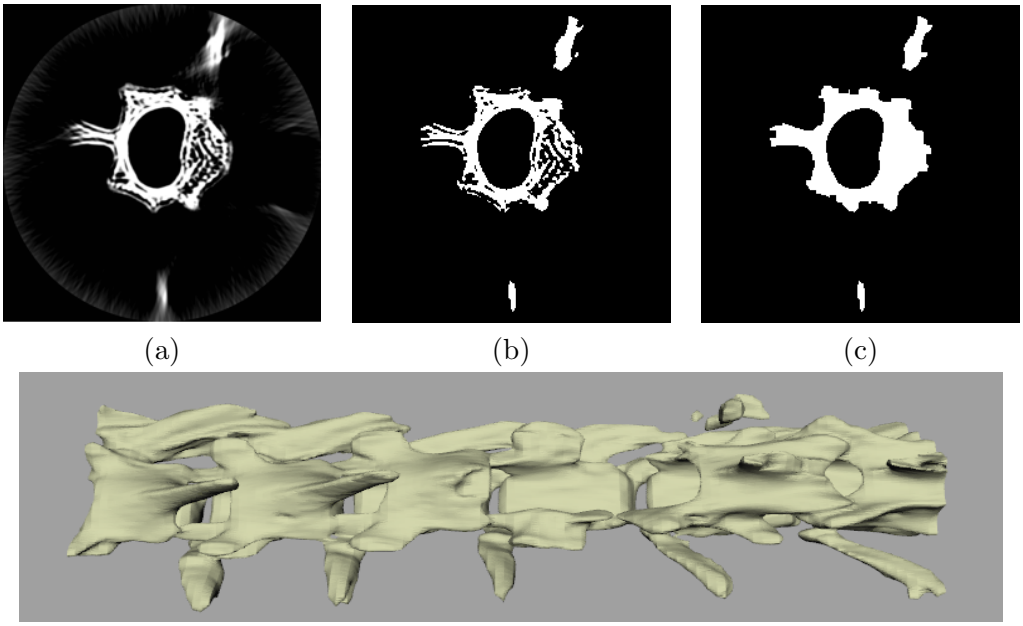
Figure 2.1: A manual interface is designed to let user interactively trace the cardiac boundaries.

methods make decisions by partitioning the image histogram into several parts. Pixels that fall in a certain intensity range are classified into the same group. The main drawback of this method is that only intensity histogram is used, but spatial information is ignored. Thus morphological operations are usually needed to post-process the segmentation results of threshold methods. See figure 2.2 for an example.

2.3 Shape Constraints

Shape constraint is of special importance in medical applications, where prior knowledge of the anatomy is needed to help constrain the evolution of the segmentation process. Such as the morphological operations described in the previous section, shape constraints keep the desired topological structures and trim the boundaries.

Deformable models [3] are physics-based models that are capable of controlling the geometry and smoothness of the segmented boundaries, and allowing significant variabilities of the biological structures. In deformable model formulations, two energy terms are considered to evolve the contour. The external energy term is obtained from the image information, such as intensity level and intensity gradient. The internal energy term is from the contour smoothness. Lower energy corresponds to smaller curvature and smaller change in curvature. Thus deformable model segmentation is



(a) (b) (c)

(d)

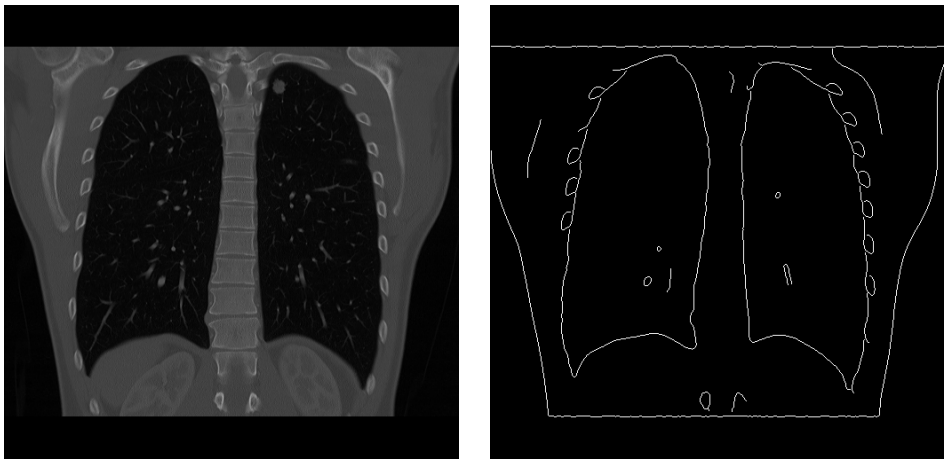
Figure 2.2: Threshold methods. (a) is the input CT image. (b) is the binary image after thresholds. (c) is the refined results after morphological operations. (d) is the 3D reconstructed results of a rat's vertebra.

formulated as an energy minimization problem.

Snake [4] is one of the most popular deformable models, which is usually used in 2D piecewise continuous and smooth contour segmentation. In Snake algorithm, the contour is presented as a parameterized curve with a fixed topology.

Levelset [5] is another important deformable model. It implicitly represents shape as the zero level of a levelset function: the boundaries are embedded in a higher dimensional space. Then the higher dimensional levelset function is evolved rather than the boundary itself. In this way, levelset achieves flexible topological changes.

In many shape analysis methods, high dimensional shape space is projected onto lower dimension. Active Shape Model (ASM) [6] uses principle component analysis to linearly model shape variations. A new shape can be represented as the mean shape plus a linear combination of the principle components.



(a) (b)

Figure 2.3: (a) is a CT image of the lungs. (b) is the obtained from a Canny edge detector.

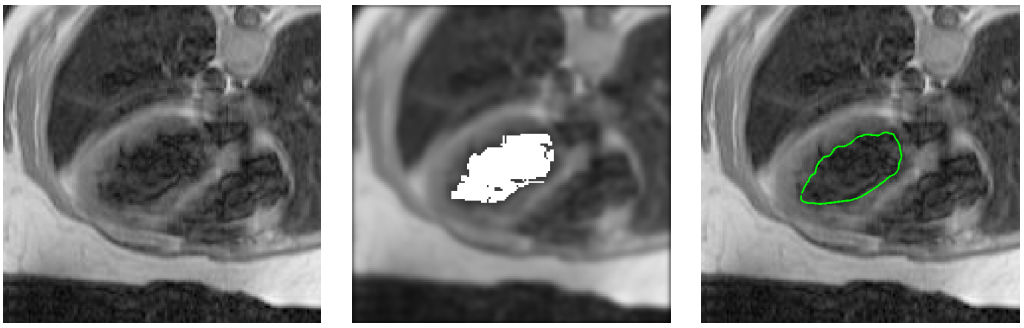
2.4 Edge Based Methods

Edge is usually formulated as intensity gradient. It is a good indicator of the boundary locations when edges are prominent in the image. For instance, Canny edge detector [7] is a multi-stage algorithm that detects the zero-crossings of the second directional derivative of the smoothed image. See figure. 2.3 for an example. For very noisy and blurred images, edge based methods are prone to error.

In deformable models, the main limitation of edge-based image force is that edge is a local image descriptor. When the curve is far away from the edge, there is no or little image force to attract or push it. To solve this problem, balloon force or gradient vector flow (GVF) [8] techniques can be applied. GVF essentially defuses the gradient field of the input image so that the edge forces can reach farther.

2.5 Region Based Methods

Region based methods partition the image into connected regions that contain pixels of similar intensity. It tries to minimize the intensity difference inside each segmented regions and maximize the intensity difference in between regions. For instance, MRF [9] and fuzzy affinity [10] are two region based methods. Both methods combine intensity similarity and pixel connectedness to form a more complex region segmentation



(a) (b) (c)

Figure 2.4: Fuzzy affinity segmentation. (a) is a chest MR image. The endocardium of the left ventricle needs segmentation. (b) is the binary results of the fuzzy affinity algorithm. (c) is the segmentation results refined by a 2D Snake.

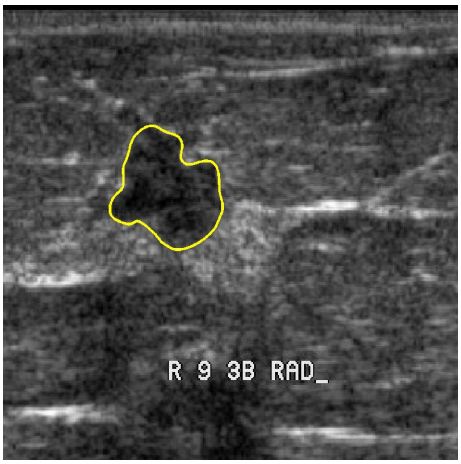


Figure 2.5: Metamorphs segmentation of breast lesion in an ultrasound image.

rule. See figure. 2.4 for a deformable model segmentation example using fuzzy affinity technique.

2.6 Hybrid Methods

Since both edge and region based methods have limitations, it is desirable to develop hybrid methods that combines them. Metamorphs [11] is a deformable model using both edge and region information. The edge term is embedded in a distance map function, and the region term is represented as a probability density function. The shape evolves to minimize both energy terms on a free form deformation grids. See figure 2.5 for an example.

Chapter 3

3D Colonography Segmentation

3.1 Background of MR Colonography

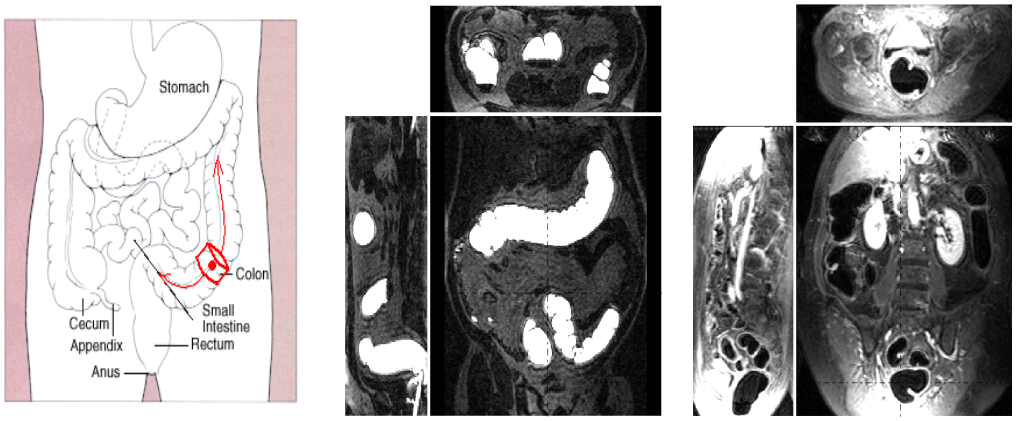
MR colonography is a new technology for the accurate detection of colonic polyps [12]. This technology is less painful for patients than conventional colonoscopy, which leads to better patient participation in screening programs of colorectal cancer. MR colonography is also considered safer than CT colonography, because MR imaging has not radiations. But MR images have higher noise levels and lower resolutions than CT images, which makes the tasks of image postprocessing and image analysis more difficult.

There are two main technologies for MR colonography [13]. One is bright lumen colonography, which needs bowel cleansing. The other one is dark lumen colonography, which doesn't need bowel cleansing and has better patient acceptance. However this dark lumen technique has higher noise level. See figure 3.1 (b, c) for two image samples.

We are trying to segment the colon in 3D MR scan of human abdomen. Our segmentation results should facilitate the following polyps detection and classification tasks. This segmentation method should work for both bright and dark colonography cases.

There are some previous work which was based on multi-resolution region growing. It worked well on bright lumen images. However for dark lumen images, because of the spurious edges and the inconsistent intensity levels in both the dark lumen area and the other abdomen organs, it has limitations.

In our method, we are trying to develop a model-based segmentation method. We designed a tube-shaped shape model to direct the segmentation and tracking. Voxel



(a) (b) (c)

Figure 3.1: (a) Illustration of the tracking method over the human colon. (b) Bright lumen colonography. (c) Dark lumen colonography.

intensities and intensity gradients in the local region are the main image forces considered. Before the segmentation begins, a seed point in the lumen area needs to be selected manually. Then we try to fit a initial cylinder (tube) model for this seed position in an initialization step. Then the initialized model should be able to grow in both directions and track over the colon, allowing possible tuning of the parameters of the shape model. See figure 3.1 (a) for an illustration of the scheme.

The shape model is designed as a bendable cylinder, which is controlled by 5 parameters: dx, dy for the translation, r for the radius, and ϕ_x, ϕ_z for the rotation angles. The bending angles are determined by the differences of the rotation angles $d\phi_x, d\phi_z$ from two consecutive steps.

3.2 Preprocessing: Inhomogeneity Correction

Intensity inhomogeneity is a common problem in MR imaging. Inhomogeneity correction increases the image readability and helps the segmentation. There are many literatures on this topic [14, 15]. Many of these methods are computationally demanding, e.g., in [14] their method requires about 1 minute for a 2D MR image. Since our data is fully 3D, which usually consists of more than 70 2D slices, these methods are computational infeasible.

Here we implemented a simpler and faster inhomogeneity correction method [16],

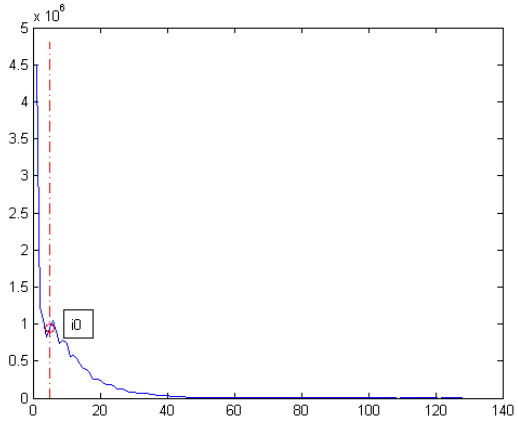


Figure 3.2: The histogram of the input 3D MR image. i_0 is the intensity threshold to determine whether a voxel belongs to the foreground or the background.

whose results are not perfect but can greatly help the following segmentation.

For simplicity, we assume the MR image only consists of two layers: the foreground and the background. First the intensity values $I(x, y, z)$ of the input 3D image are linearly scaled to the range of $[0 - 127]$. Then its histogram H is calculated as:

$$H(i) = \sum_{x,y,z} h(x, y, z), \quad for \quad 0 \leq i \leq 127 \tag{3.1}$$

where,

$$h(x, y, z) = \begin{cases} 1, & when \quad i - 0.5 \leq I(x, y, z) < i + 0.5 \\ 0, & otherwise \end{cases} \tag{3.2}$$

From H's first derivative H', we find the first index i_0 such that $H'(i_0) > 0$ as the threshold value. For a voxel (x, y, z) in the MR image, if $I(x, y, z) > i_0$ then we classify it as a foreground voxel, otherwise we classify it as a background voxel.

For all the foreground voxels, we find their median intensity value M_f. Then we can construct a foreground image I_f by replacing all the intensity values of the background voxels by M_f:

$$I_f(x, y, z) = \begin{cases} M_f, & if \quad I(x, y, z) \in [0, i_0] \\ I(x, y, z), & otherwise \end{cases} \tag{3.3}$$

The foreground image I_f is blurred to I_b by convolving with a 3D Gaussian kernel,

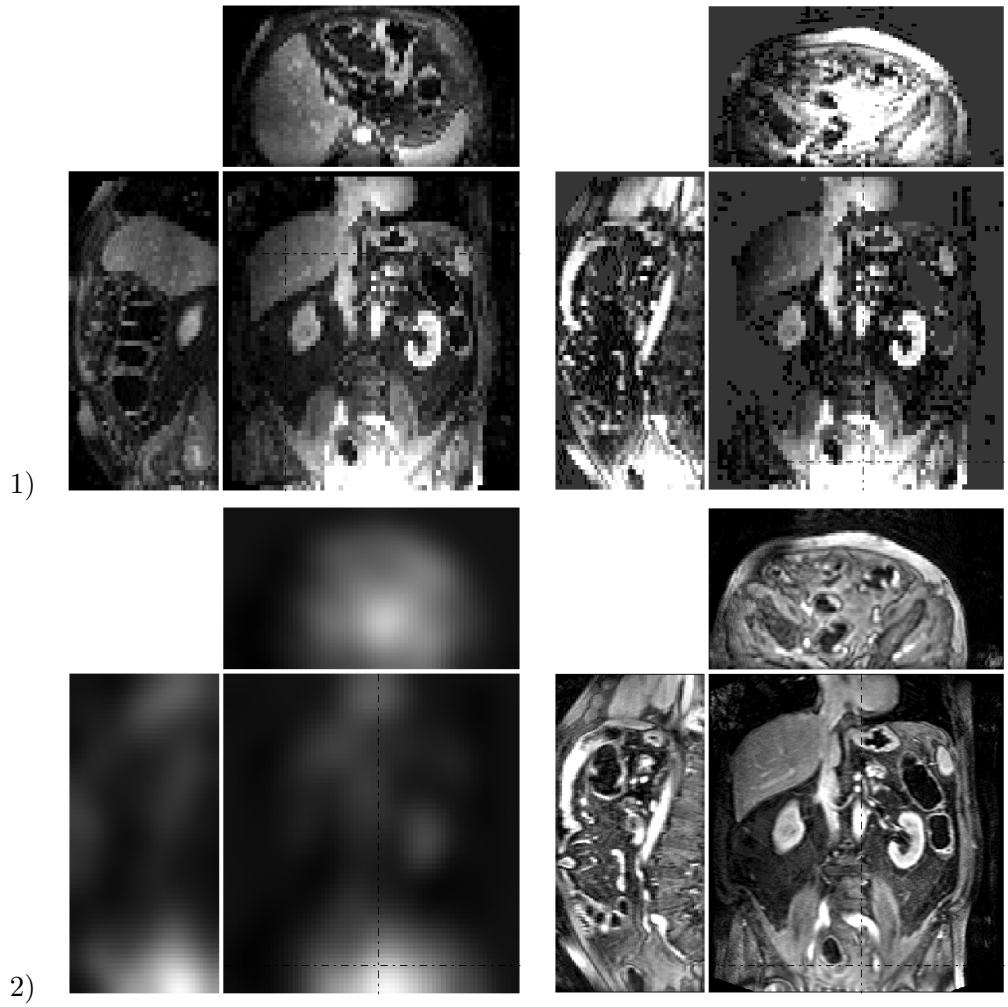


Figure 3.3: (1a) is the 3D view of the original input image. (1b) is the foreground image I_f. (2a) is the Gaussian blurred foreground image. (2b) is the inhomogeneity corrected image I_b. Image is down-sampled in (1a), (1b), and (2a) to achieve faster implementation.

whose standard deviation σ is experimentally set to one third of the image size in the $X-Y$ plane. For example, for a $512 \times 512 \times 72$-sized 3D image, we set $\sigma = 170$. Then the inhomogeneity corrected image I_c is derived by normalizing the input I with the blurred foreground I_b:

$$I_c(x, y, z) = I(x, y, z)/I_b(x, y, z) \tag{3.4}$$

The computation of this MR image inhomogeneity correction method is fast. We

implement this method on a P4 2.8G workstation using MatLab 6.5. For a 512×512×72-sized 3D image, it takes less than 10s.

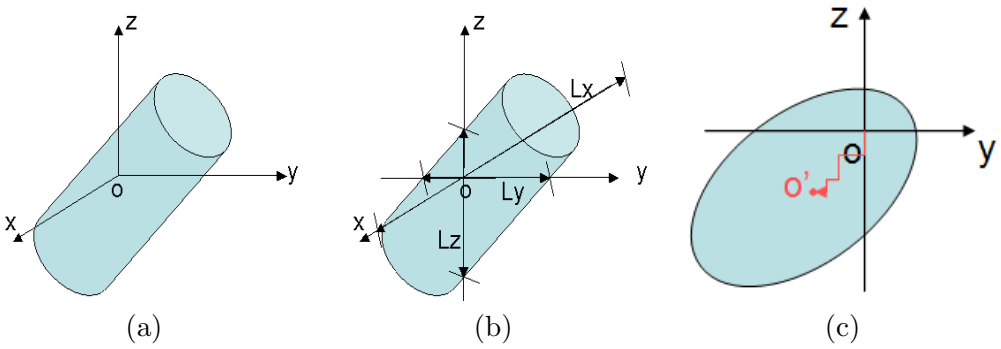
3.3 Segmentation Initialization

In the initialization step we will find the radius and orientation of the local colon segment at some seed point positions. For current implementation, these seed points are manually placed in the lumen area (in future work, we will try to find the seed points with an automatic method). And because the dark lumen case is more difficult, in this work, we are more focusing on the segmentation of dark lumen MR colonography. Since in the initialization step, we don't have much prior knowledge of the colon's size and orientation, we prefer to choose the seed points at where it has better image quality and simpler colon structure to make the initialization easier. For example, we will choose areas where the dark lumen has very low and relatively homogenous intensities, and where the local colon segment has a very small curvature.

Then we will fit a tube-shaped models with the local image. The fitting process is done in a multi-resolution way to make the computation faster.

As shown in Fig. 3.4, suppose we have a local colon segment illustrated as a cylinder, and o is the manually picked seed point. Then along the x, y and z coordinates, we find the lengths L_x, L_y and L_z of the segments whose intensities are below a certain threshold. The threshold is set experimentally in our algorithm. But for dark lumen MRI after intensity inhomogeneity correction, this threshold doesn't vary much. For example, when the image intensity is re-scaled to the range of $[0, 1.5]$, a threshold value of 0.1 works for most of the image data that we tested.

The manually selected seed point o usually doesn't locate at the geometrical center of the colon. We leave out the coordinate who has the biggest L, e.g., in Fig. 3.4(b), L_x, and iteratively look for the center along the other two coordinates, as shown in Fig. 3.4(c), to find the geometrical center o'. Note that since this method only uses the 1D profiles, the computation is very fast. However this method is sensitive to noise. This is why we mentioned that the seed point needs to be carefully chosen in



(a) (b) (c)

Figure 3.4: (a) A local colon segment illustrated as the cylinder with a seed point o inside. (b) The coordinates lengths inside the colon are illustrated as L_x, L_y and L_z. (c) Iteratively centering in y and z directions to find the new center o'.

the beginning of this section.

After we get the geometrical center o' and the new $L_x{}'$, $L_y{}'$ and $L_z{}'$, we can estimate the radius and orientation of the local colon segment. The radius r and rotation angles ϕ_x, ϕ_z are illustrated in Fig. 3.5(a). Hence r is derived by:

$$r = \sqrt{\frac{2}{\frac{1}{(\frac{L_x{}'}{2})^2} + \frac{1}{(\frac{L_y{}'}{2})^2} + \frac{1}{(\frac{L_z{}'}{2})^2}}} \tag{3.5}$$

And ϕ_x, ϕ_z are:

$$\phi_x = \pm \text{asin}\left(\frac{2r}{L_z{}'}\right) \tag{3.6}$$

$$\phi_z = \pm \text{asin}\left(\frac{2\sqrt{r^2 - (\frac{L_x{}'}{2})^2 \cdot cos^2(\phi_x)}}{L_x{}' \cdot sin(\phi_x)}\right) \tag{3.7}$$

Both ϕ_x and ϕ_z can be either positive and negative. Hence there are 4 possible configurations of the orientation angles.

We design the tube-shaped model based on the intensity profile of the colon image. As shown in Fig. 3.5(b), which is the slice view of a tube-shaped model, we set the intensities of the voxels inside the tube model positive, and the voxels outside negative. Thus when we fit the model onto the image, we only need to do template matching and search for the lowest dot production value of the model and the local image.

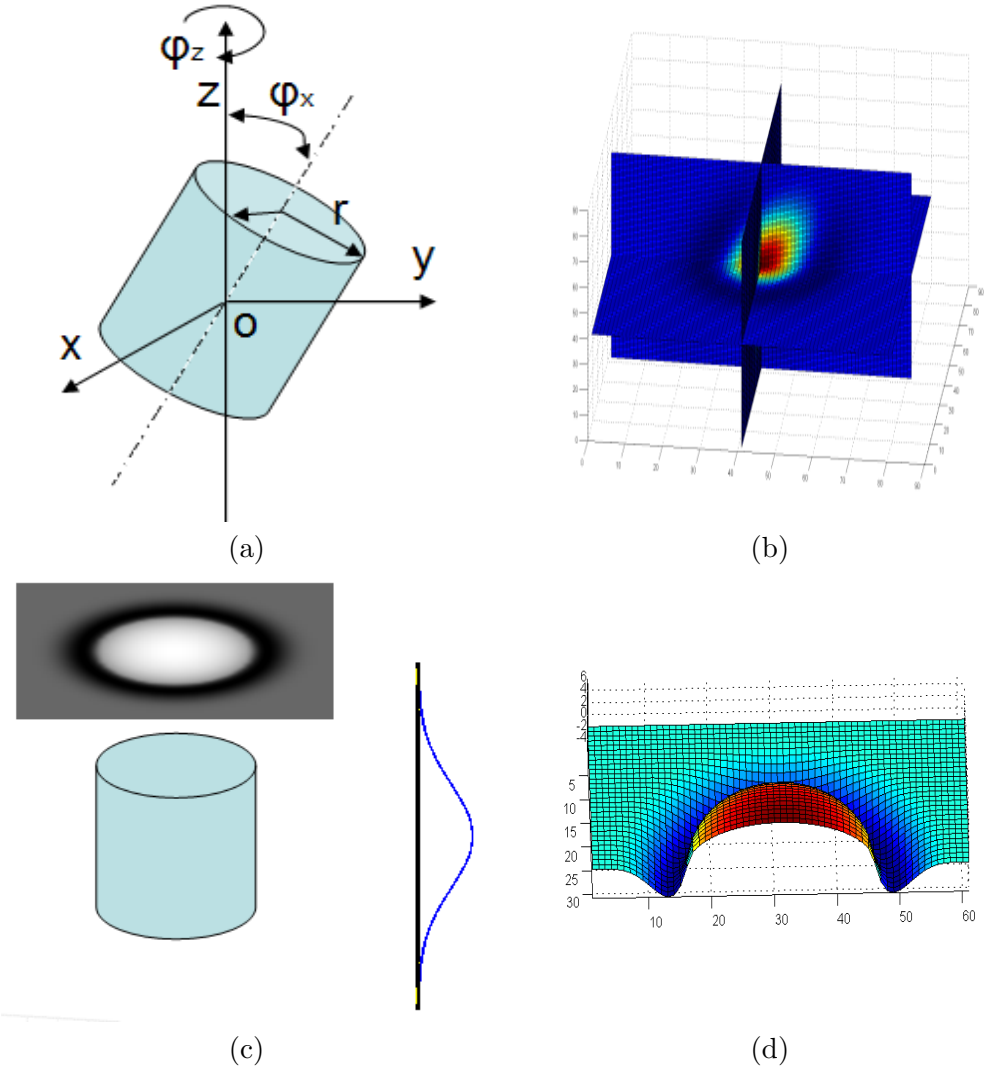


Figure 3.5: (a) The tube-shaped model is determined by 3 parameters, r, ϕ_x and ϕ_z. (b) The slice view of a model which is rotated by ϕ_x and ϕ_z. (c) The structure of the un-rotated tube model. (d) The intensity values of a cross section plane. Only half of the plane is drawn to show the intensity values along the center line.

For normalization purpose, the summation of the intensity value of each voxel in the tube-shaped model need to be zero. Therefore we design the model as following. For a un-rotated model, whose long axis is parallel to the z coordinate, as shown in Fig. 3.5(c), the intensity value along the z direction is modeled as a Gaussian with the standard deviation experimentally set to equal r, which means we put more importance on the tube's center area. If we keep the intensity summation in the cross section to be zero, then we can make the intensity summation of the whole volume be zero. Hence we model the voxel's intensity value i in the cross section plane as two Beta functions w.r.t. its Euclidian distance d to the center axis:

$$i(d) = \begin{cases} \beta_1 \cdot (\frac{d-r}{r_{wall}})^{a_1-1} \cdot (1 - \frac{d-r}{r_{wall}})^{b_1-1}, & \text{when} \quad d > r \quad \text{and} \quad d < r + r_{wall} \\ \beta_2 \cdot (\frac{r-d}{2r})^{a_2-1} \cdot (1 - \frac{r-d}{2r})^{b_2-1}, & \text{when} \quad d \leq r \end{cases} \tag{3.8}$$

where,

$$\begin{aligned} \beta_1 &= \frac{1}{r \cdot r_{wall} \cdot B(a_1,b_1) + r_{wall}^2 \cdot B(a_1+1,b_1)} \\ \beta_2 &= \frac{1}{r^2 \cdot B(a_2,b_2) - 4r^2 \cdot F_B(0.5;a_2+1,a_2) \cdot B(a_2+1,b_2)} \end{aligned} \tag{3.9}$$

Where $B(a, b)$ is a Beta function, $F_B(x; a, b)$ is a cumulative Beta function, and $\Gamma(x)$ is the Gamma function:

$$\begin{aligned} B(a,b) &= \tfrac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \\ F_B(x;a,b) &= \tfrac{1}{B(a,b)} x^{a-1}(1-x)^{b-1} \\ \Gamma(x) &= \textstyle\int_0^\infty t^{x-1} e^{-t} dt \end{aligned} \tag{3.10}$$

Here we use r_{wall}, (a_1, b_1) and (a_2, b_2) to control the curve shape of the intensity profile along the center line of the cross section. As shown in Fig. 3.5(d), r_{wall} is set to approximate the thickness of the colon wall, i.e., the width of the curve that is below zero. And (a_1, b_1) and (a_2, b_2) are experimentally set to $a_1 = 2$, $b_1 = 5$, $a_2 = 1.5$, and $b_2 = 1.5$.

From Equations 3.5,3.6,3.7, we test the four sets of rotation angles and find the best match. Then near this set of parameters r, ϕ_x and ϕ_z we exhaustively search for

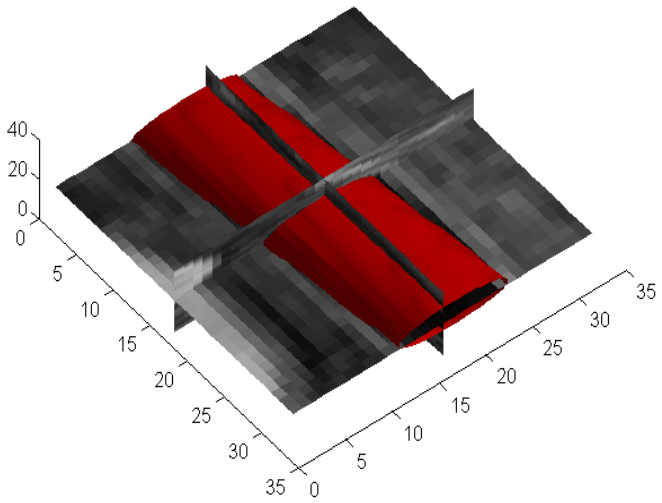


Figure 3.6: The red tube is the initial fitting of the local colon segment.

the optimal parameter set. This search is done in multi-resolutions to achieve faster implementation. See Fig. 3.6 for an initialization result.

3.4 Segmentation via Tracking

After the initialization, we start to track from the initial model from the both ends of the cylinder. There are 5 tunable parameters during the tracking: dr for the tube's radius change, $d\phi_x$ and $d\phi_z$ for the tube's orientation changes, and dx, dy for the center point's translations within the plane of the colon cross section.

Note that $d\phi_x$ and $d\phi_z$ are rotation angles in the local coordinates. In the global coordinates, the newer orientation angles ϕ'_x and ϕ'_z are derived by:

$$\begin{aligned}\phi'_x &= \text{acos}\left(-\sin(\phi_x)\cos(d\phi_z)\sin(d\phi_x) + \cos(\phi_x)\cos(d\phi_x)\right) \\ \phi'_z &= \text{atan}\left(\frac{cos(\phi_z)sin(d\phi_z)sin(d\phi_x)+sin(\phi_z)cos(\phi_x)cos(d\phi_z)sin(d\phi_x)+sin(\phi_z)sin(\phi_x)cos(d\phi_x)}{-sin(\phi_z)sin(d\phi_z)sin(d\phi_x)+cos(\phi_z)cos(\phi_x)cos(d\phi_z)sin(d\phi_x)+cos(\phi_z)sin(\phi_x)cos(d\phi_x)}\right) + k\end{aligned} \tag{3.11}$$

Where:

$$k = \begin{cases} \pi, & when \quad \frac{-sin(\phi_z)sin(d\phi_z)sin(d\phi_x)+cos(\phi_z)cos(\phi_x)cos(d\phi_z)sin(d\phi_x)+cos(\phi_z)sin(\phi_x)cos(d\phi_x)}{sin(\phi'_x)} < 0 \\ 0. & otherwise \end{cases} \tag{3.12}$$

Also note that (dx, dy) are not the translations in the global coordinates. The global translation $[dx', dy', dz']$ can be derived by:

$$\begin{pmatrix} dx' \\ dy' \\ dz' \end{pmatrix} = \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\phi'_x) & -sin(\phi'_x) \\ 0 & sin(\phi'_x) & cos(\phi'_x) \end{pmatrix} \cdot \begin{pmatrix} cos(\phi'_z) & -sin(\phi'_z) & 0 \\ sin(\phi'_z) & cos(\phi'_z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \right)^T \cdot \begin{pmatrix} dx \\ dy \\ 0 \end{pmatrix} \tag{3.13}$$

3.4.1 Tracking Templates Design

The tracking template is similar with the tube-shaped model we used in the initialization step. However, we add two more features to the model to make it more robust for noisy images and in those highly curved colon regions.

First, we make the template bendable. As shown in Fig. 3.7(a), in the local coordinate, the step size s and rotation angle $d\phi_x$ determine the curvature of the template. The curvature radius R is set as:

$$R = \frac{s}{sin(d\phi_x)} \tag{3.14}$$

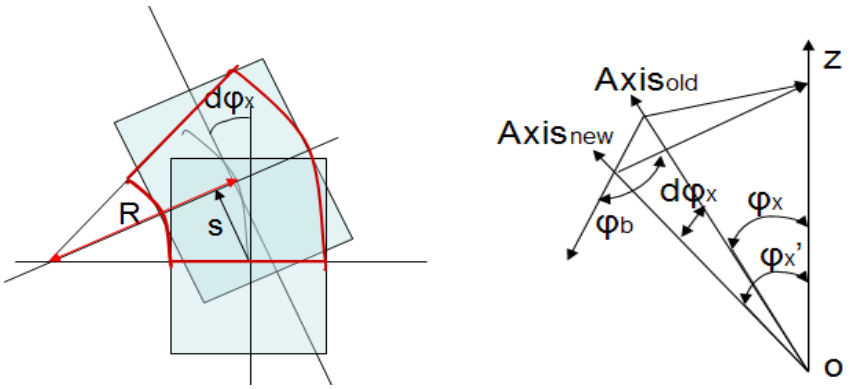
And as shown in Fig. 3.7(b), the bending orientation ϕ_b is determined as:

$$\phi_b = \left(\text{acos}\left(\frac{cos(\phi_x) - cos(d\phi_x)cos(\phi'_x)}{sin(d\phi_x)sin(\phi'_x)} \right) + k \right) \cdot p \tag{3.15}$$

where,

$$k = \begin{cases} \pi, & when \quad d\phi_x < 0, s > 0 \quad or \quad d\phi_x > 0, s < 0 \\ 0, & otherwise \end{cases} \tag{3.16}$$

and,



(a) (b)

Figure 3.7: (a) The curvature of the bending tube is determined by $d\phi_x$. (b) The orientation of the bending is determined by Equation 3.15.

$$p = \begin{cases} -1, & when \quad d\phi_z < 0 \\ 1, & otherwise \end{cases} \tag{3.17}$$

Second, we separate the tube-shaped model into two independent templates, the edge term and the intensity term. We also discretize the edge term into several sets of 1D profiles. Thus we don't need to consider the normalization problem as in the previous model.

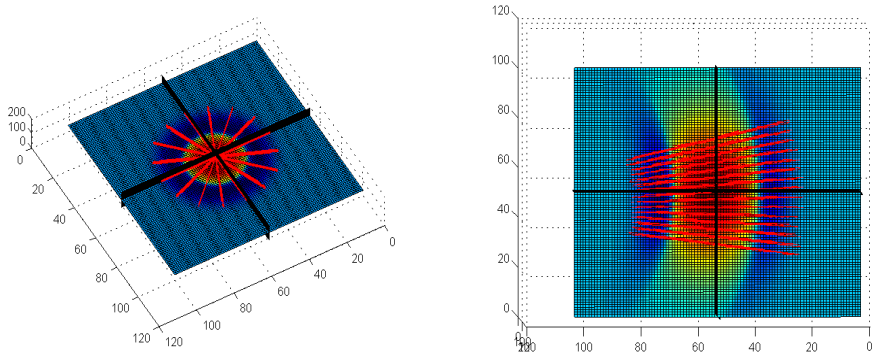
Edge Term

And as shown in Fig. 3.8(a), on the cross section of the tube model, the edge template is discretized into 8 1D profiles that are angularly evenly spaced. And as shown in Fig. 3.8(b), along the axis of the tube, it is also discretized into a set of cross section planes. The angular range of the plane position is experimentally determined by $[-1.3 \cdot r/R, 1.3 \cdot r/R]$, where R is the curvature radius from Equation 3.14.

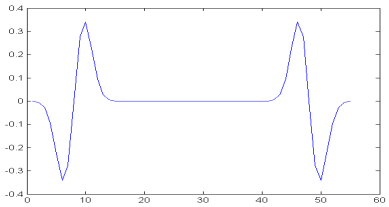
The 1D profile is set as the combination of two edge detectors, as shown in Fig. 3.8(c), which is the summation of two Gaussians' derivative.

Intensity Term

The intensity term is set similar as a non-negative form of the model in the initialization step. For a voxel at position (x, y, z) in an un-rotated template, we set:



(a) (b)



(c)

Figure 3.8: (a) A cross section view of the edge template. (b) A axis view of the edge template. (c) The 1D profile is matching with the edge detectors.

$$c = y \cdot cos(d\phi_z) + x \cdot sin(d\phi_z) \tag{3.18}$$

$$\alpha = atan(\frac{z}{R-c}) \tag{3.19}$$

Then the distance from this voxel to the nearby tube axis is:

$$d = \sqrt{((Rcos(\alpha) - R)sin(d\phi_z) + x)^2 + ((Rcos(\alpha) - R)cos(d\phi_z) + y)^2 + (z - Rsin(\alpha))^2} \tag{3.20}$$

And the intensity of this voxel in the template is defined as:

$$i(d) = \frac{\beta_2(\frac{r-d}{2r})^{a_2-1}(1-\frac{r-d}{2r})^{b_2-1}}{\sqrt{2\pi}r} e^{\frac{-\alpha^2 R^2}{2r^2}} \tag{3.21}$$

See Fig. 3.9 for an example of the intensity template.

3.4.2 Tracking via Sampling

The goal of tracking is to find the best set of parameters step by step while the model grows. We have tried using several conventional optimization algorithms, such as the

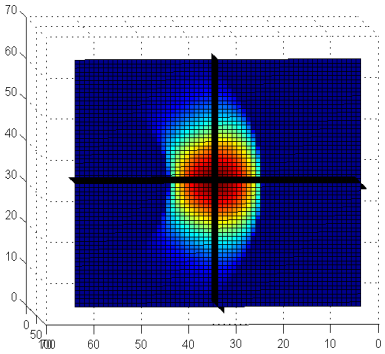


Figure 3.9: A slice view of the intensity template.

simplex method and the gradient descent method, to search for the best set of parameters. But these optimization methods are not very robust and tend to get stuck in local minima, which often leads to leakage. Another problem of these optimization methods is that although the parameters from the previous step in the tracking process gives a good initialization for the current step, it is still difficult to predefine and constrain the parameters' range.

Thus we tried a sampling approach. Based on the parameter values from the previous step, we predefine a set of distributions of the possible parameter changes, then randomly sample from these distributions.

The distributions correlate closely to the step size s. If s is small, we can make the distribution range narrower, because with smaller s, the colon's shape and orientation have less variations.

Among the 5 tunable parameters, dr, $d\phi_x$, dx, and dy can be modeled as Gaussian distributions. Their standard deviations will be smaller if we use smaller s.

The other parameter $d\phi_z$ is randomly selected in the range of $[0, 2\pi]$, which means it is independent with the value of s.

Since we have two separated terms, the edge term and the intensity term, we have to properly combine them. We tested the two terms' changing rates on a phantom image, and find the energy function of the edge term changes 50 to 100 times faster than the intensity term. Thus we approximately set the rate $k = 70$. Suppose E_i and E_e are energy values from the intensity and edge terms, and we want to minimize the

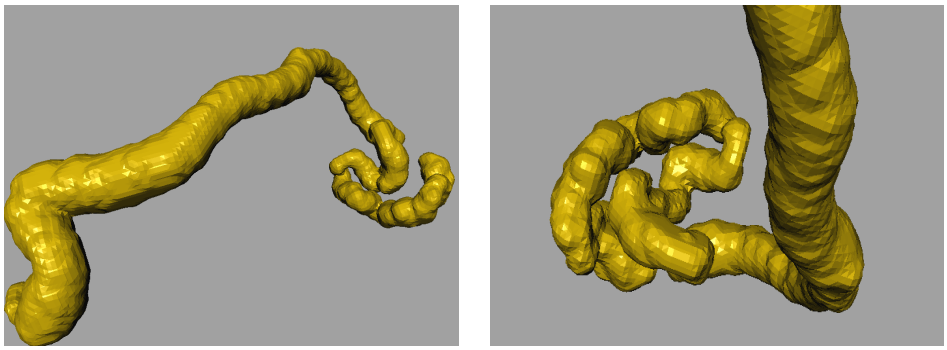


Figure 3.10: A result of the tracking method in two view angles. The colon portion we segmented is the right hand side part of a patient's colon.

both energies. Then the combined energy function is set as:

$$E = e^{k(E_i - E_{ip})} + e^{E_e - E_{ep}} \tag{3.22}$$

where E_{ip} and E_{ep} are the energy values from the previous step. The combined energy value E gives a performance measurement of the tracking. If E keeps less than or equal 2, we can say the tracking process is not getting worse.

Since random sampling is not efficient, we try to iteratively shift the sampling center and narrower the distribution range if E is less than a certain threshold. In this way our sampling method converges much faster.

3.5 Experimental Results

We have tried our method on several image data. With bigger step size, e.g., $s = r/3$, our method works well at smooth colon regions. When we choose smaller step size, e.g., $s = r/8$, our method can tracking well in the highly curved sigmoid colon region. However smaller step size increases the computational load.

See Fig. 3.10 for some example results. We can see that our tracking method successfully tracks the colon in those highly curved area, such as the sigmoid colon region, whose geometry is highly complicated, and where non-model based segmentation method tends to have error connections and error colon pathways. See Fig. 3.11 . We draw the center line of the sigmoid colon from our tracking method, which is correct.

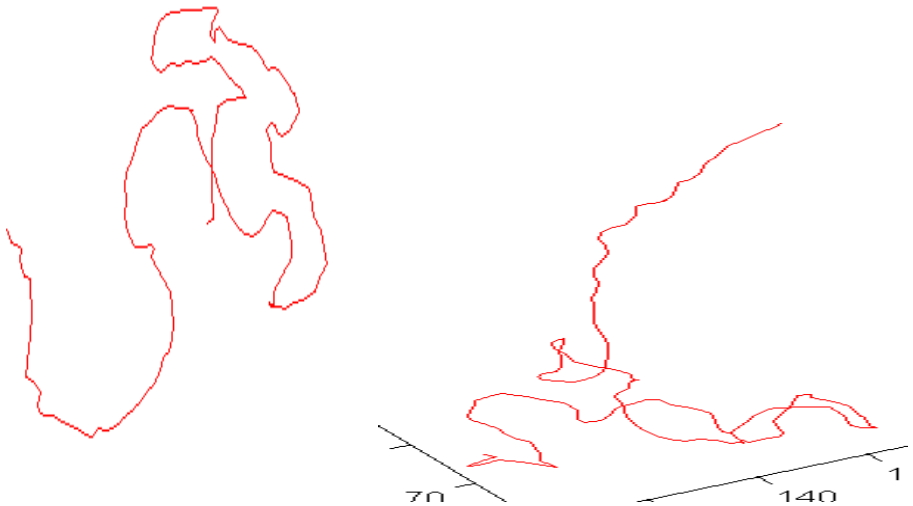


Figure 3.11: The center line extracted from our tracking results in the sigmoid colon region.

We can find the centerline has high curvatures and an irregular path way. Conventional method may leak through the colon wall and get a wrong pathway.

Chapter 4

Segmentation in Cardiac Tagged MRI

4.1 Background of Cardiac Tagged MRI

Cardiac tagged magnetic resonance imaging(MRI) is a well known technique for non-invasively visualizing the detailed motion of myocardium throughout the heart cycle. The tagged MRI technique generates a set of equally spaced parallel tagging lines within the myocardium as temporary markers at end-diastole by spatial modulation of the magnetization. Then the dark-colored tagging lines will persist for a short period of time in the myocardium and deform with the underlying tissue during the cardiac cycle in vivo, which provides the detailed myocardial motion information. See figure 4.1 for some examples.

This technique has the potential of early diagnosis and quantitative analysis of various kinds of heart diseases and malfunction. However, before it can be used in the routine clinical evaluations, an imperative but challenging task is to automatically find the boundaries of the epicardium and the endocardium.

Segmentation in tagged MRI is difficult for several reasons. First, the boundaries are often obscured or corrupted by the nearby tagging lines, which makes the conventional edge-based segmentation method infeasible. Second, tagged MRI tends to increase the intensity contrast between the tagged and un-tagged tissues at the price of lowering the contrast between the myocardium and the blood. At the same time, the intensity of the myocardium and blood vary during the cardiac cycle due to the tagging lines fading in the myocardium and being flushed away in the blood. Third, due to the short acquisition time, the tagged MR images have a relatively high level of noise. These factors make conventional region-based segmentation techniques impractical. The last and the most important reason is that, from the clinicians' point of view, or for the

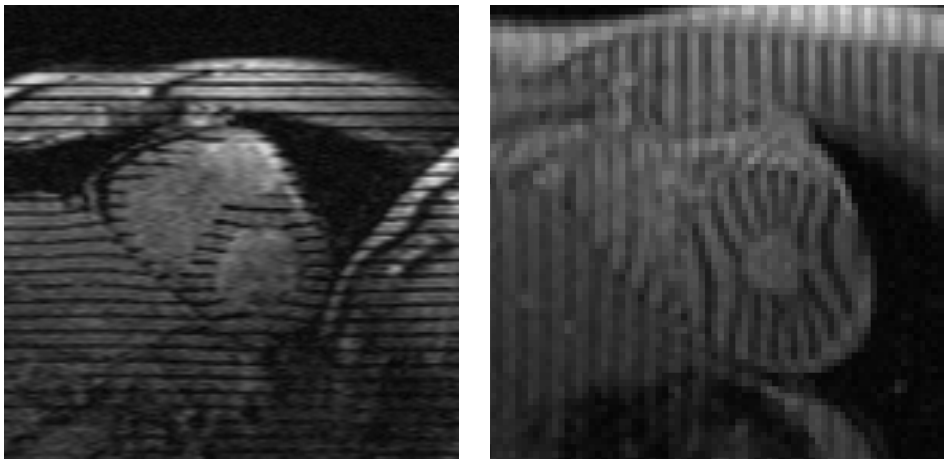


Figure 4.1: Two sample images of cardiac tagged MRI.

purpose of 3D modeling, the *accurate* segmentation based solely on the MR image is usually not possible. For instance, for conventional clinical practice, the endocardial boundary should exclude the papillary muscles for the purpose of easier analysis. However, in the MR images, the papillary muscles are often apparently connected with the endocardium and cannot be separated if only the image information is used. Thus the prior shape knowledge is needed to improve the results of automated segmentation.

There have been many efforts to achieve tagged MRI segmentation. In [17] gray scale morphological operations were used to find non-tagged blood filled regions. Then they used thresholding and active contour methods to find the boundaries. In [18] a learning method with a coupled shape and intensity statistical model was proposed. In [2] Gabor filtering was used to remove the tagging lines before the segmentation. These methods work in some cases. However they are still imperfect: morphological operations are sensitive to image noise, intensity statistical model cannot capture the complex local texture features, and filtering methods blur the boundaries and decrease the segmentation accuracy.

In this work, in order to address the difficulties stated above, we propose a novel and fully automatic segmentation method based on three learning frameworks: 1. An rotation invariant shape model is used as the prior heart shape model; 2. A set of local boundary criteria are learned by Adaboost at landmark points of the shape model using the appearance features in the nearby local regions. These criteria give the probability

of the local region's center point being on the boundary, and force their corresponding landmark points to move toward the direction of the highest probability regions. 3. An Adaboost detection method is used to initialize the segmentation's location, orientation and scale. The second component is the most essential contribution of our method. We abandon the usual edge or region-based methods because of the complicated boundary and region appearance in the tagged MRI. It is not feasible to designate one or a few edge or region rules to solve the complicated segmentation task. Instead we try to use all possible information, such as the edges, the ridges, and the breaking points of tagging lines, to form a *complex rule*. It is apparent that at different locations on the heart boundary, this *complex rule* must be different too. It is impractical to manually set up each of these *complex rules*. Therefore, we introduce Adaboost, a popular learning scheme, to learn a set of rules at each landmark point on the shape model. The first and the second frameworks are tightly coupled. The shape model deforms by the forces from the Framework 2 while controlled and smoothed by the Framework 1. To achieve fully automatic segmentation, in Framework 3 the detection method automatically provides an approximate position and size of the heart to initialize the segmentation step.

The remainder of this paper is organized as follows: in Section 4.2, we present the rotation invariant shape model of Framework 1. In Section 4.3, we give the local appearance modeling method of Framework 2. In In Section 4.4, we briefly introduce the heart detection technique of Framework 3. In Section 4.5 we give some details of our experiments and show some encouraging experimental results.

4.2 Rotation Invariant Shape Modeling

Many shape modeling methods use a linear combination of the component vectors to represent the possible variations given the input training shapes. For example, active shape model and active appearance model were first introduced by T. F. Cootes and C. J. In their ASM algorithm [6], a statistic shape model is set up based on the principle component analysis (PCA) of a set of training data. PCA method uses a small set of principle components to represent a large data set. And a shape can be reconstructed by a linear combination of the mean shape and a weighted sum of the principle components.

There are also some other shape modeling methods such as independent component analysis (ICA) and factor analysis (FA) which are basically similar to the PCA method. However, all these methods have limitations. In practice, the input training data may not align well and are usually subjected to random transformations, such as translation, rotation, and scaling. In these cases, if we directly apply the above shape modeling methods to the unaligned training data, the results will be severely blurred and the useful structure may be likely ignored. At the same time, although we can align the shapes before training by doing translation and scaling, it is still difficult to avoid other kinds of transformations, such as rotation and shearing. Actually if rotation and shearing exist, the scaling step may even produce error.

B. Frey and N. Jojic proposed the transformed component analysis (TCA) method in 1999 [19]. They set up a mixture model to jointly estimate the image components and the spatial transformations, such as translation, scaling, rotation and shearing. Motivated by this translation invariant component analysis algorithm for images and videos, we formulate a rotation invariant component analyzer for the shape vectors, because in most medical image cases, rotation distortion is a common phenomenon and difficult to eliminate.

Since the shape of the mid portion of the heart in short axis (SA) images is consistent and topologically fixed (one left ventricle (LV) and one right ventricle (RV)), it is reasonable to implement an active shape model [6] to represent the desired boundary contours.

We acquired two image data sets each from two normal subjects, using two slightly different imaging techniques. The data sets were acquired in the short axis plane. There are two sets of tagging line orientations (0° and 90°, or -45° and 45°) and slightly different tag spacings. Each data set included images acquired at phases through systole into early diastole, and at positions along the axis of the LV, from near the apex to near the base, but without topological changes. An expert was asked to segment the epicardium (Epi), the left ventricle (LV) endocardium and the right ventricle (RV) endocardium from the datasets. In total we obtained 220 sets (each set includes one LV, one RV, and one Epi) of segmented contours to use as the training data.

Segmented contours were centered and scaled to a uniform size. Landmark points were placed automatically by finding key points with specific geometric characteristics. As shown in Figure 4.2, the black points are the key points, which were determined by the curvatures and positions along the contours. For instance, $P1$ and $P2$ are the highest curvature points of the RV; $P7$ and $P8$ are on opposite sides of the center axis of the LV. Then, fixed numbers of other points are equally placed in between. In this way, the landmark points were registered to the corresponding locations on the contours. Here, we used 50 points to represent the shape.

The shape of the mid portion of the heart in short axis images is not significantly varying and topologically fixed (one LV and one RV). Therefore it is reasonable to implement a linear subspace shape model to represent prior shape knowledge and constrain the shape variations during the myocardial wall tracking process. This is particularly desirable in tagged cardiac MRI images. Due to the noisy and complex nature of tagged MR images, without the use of prior knowledge, *accurate* boundary tracking based solely on the MR images is usually not possible. We use the expert's manual contours as the training input. We use a set of points to describe a shape. That is, in each input image, we choose a set of landmark points on the object contours to represent the object's shape. These landmark points should be consistently located from one input image to another. In practice, this landmark choosing process would be very time consuming. I used an automatic method to allocate these points. The manual contours are first centered and transformed to the same scale. Then a few feature points are selected automatically using geometry features such as maximum curvature, and all the other landmark points are chosen automatically by equally spacing them between feature points. In this application, the SA contours are automatically discretized into 50 ordered landmark points. Then the coordinates of the 50 landmark points are reshaped to a 100-element vector X, where,

$$X = (x_1, y_1, x_2, y_2, ..., x_{50}, y_{50})^T \tag{4.1}$$

Given s training examples, we generate s such vectors x_j. Then we perform TCA on these vectors.

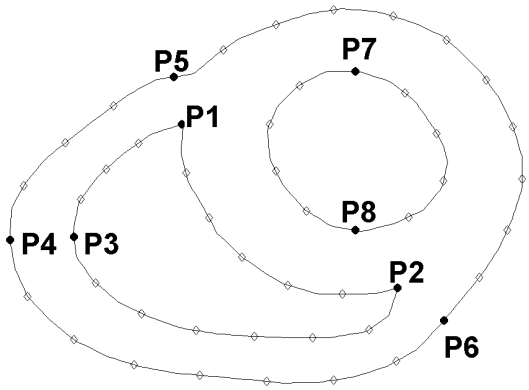


Figure 4.2: The illustration of the automatic method used to place the landmark points.

Allowing possible rotation transformations, X can be approximately represented by $R \cdot (\mu + Pb)$, where R is a rotation matrix, μ is the mean shape, P is the shape variation components matrix, and b is the component parameter vector, which determines a subspace representation of the shape X.

We adapt the same graphic model (see Fig. 4.3) of TCA [19]. This is a probability model that jointly estimates the rotation matrix R and the component model P and b via an EM approach. Assume Z is a latent shape determined by the mean shape and the component parameters. Then we assume the following Gaussian distributions:

$$\begin{aligned} p(b) &= \mathcal{N}(b; 0, I) \\ p(Z|b) &= \mathcal{N}(Z; \mu + Pb, \Phi) \\ p(X|l, Z) &= \mathcal{N}(X, R_l Z, \Psi) \end{aligned} \tag{4.2}$$

where I is the identity covariance matrix, and Φ, Ψ are two diagonal covariance matrices. The goal of the EM algorithm is to maximize the joint distribution over the training shape, the rotation angle, the latent shape and the component model. Based on the graphic model depicted in Fig. 4.3, we get:

$$\begin{aligned} p(X, l, Z, b) &= p(b)P(l)p(Z|b)p(X|l, Z) \\ &= P(l)\mathcal{N}(b; 0, I)\mathcal{N}(Z; \mu + Pb, \Phi)\mathcal{N}(X, R_l Z, \Psi) \end{aligned} \tag{4.3}$$

A transformed component analyzer (TCA) is a probability model that jointly estimates the spatial transformations and the components model [19]. The set of components vectors are transformed in different ways to eliminate transformation effects before they are linearly summed up to model the input shape. Below is a graphic model of TCA.

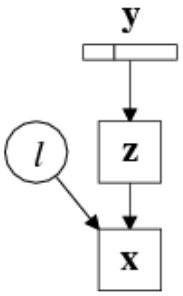


Figure 4.3: The graphic model of our method.

In the graph above, b is the component activities and forms a subspace representation of the input shape. b should be independent and initialized to be a Gaussian with equal standard deviations:

$$p(b) = \mathcal{N}(b; 0, I) \tag{4.4}$$

where I is the identity covariance matrix. Obviously, it is also a diagonal matrix which means b is independent. Modulating b will change the parameters of the linear combination of different components, thus to change the output shape z. z is a latent shape produced by combining the components linearly using a "factor loading matrix" P, plus a mean shape μ and a independent Gaussian noise Ψ. Ψ is a diagonal covariance matrix, which means the noise at each landmark point is independent.

$$p(Z|b) = \mathcal{N}(Z; \mu + Pb, \Phi) \tag{4.5}$$

Then we think about the latent shape z to be further transformed by a transformation l to obtain the observed shape x. Although continuous-valued rotation angles are more preferable, they will introduce complicated integrals into the EM algorithm that

appears in the later section. Thus we assume all the transformations are discrete. For instance, in our case, we assume the possible rotation angles vary from -30° to 30°, and the variation step is 1°, thus we get altogether 61 possible rotation angles, i.e., 61 rotation matrices. These angles are discrete. To increase accuracy, we may decrease the angle step, but this will also lead to more computation. Let $l \in \{1, 2, 3, \ldots, 61\}$ index the set of 61 rotation angles represented by the matrices $R_1, R_2, \ldots, R_{61}$. And R_l is given by:

$$R_l = \begin{bmatrix} \cos(\theta_l) & -\sin(\theta_l) & 0 & \ldots & 0 \\ \sin(\theta_l) & \cos(\theta_l) & 0 & \ldots & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & \ldots & 0 & \cos(\theta_l) & -\sin(\theta_l) \\ 0 & \ldots & 0 & \sin(\theta_l) & \cos(\theta_l) \end{bmatrix} \tag{4.6}$$

The probability density of the shape vector X for the shape corresponding to transformation l and latent shape z is

$$p(X|l, Z) = \mathcal{N}(X, R_l Z, \Psi) \tag{4.7}$$

where I is a diagonal covariance matrix that specifies the noise on the observed shape landmarks. Each of the transformations l has a prior probability $P(l) = p_l = 1/L$. This prior probability may be updated later.

The joint distribution over the observed shape, the transformation index, the latent shape, and the component activity is:

$$\begin{aligned} &p(X, l, Z, b) = p(b)P(l)p(Z|b)p(X|l, Z) \\ &= P(l)\mathcal{N}(b; 0, I)\mathcal{N}(Z; \mu + Pb, \Phi)\mathcal{N}(X, R_l Z, \Psi) \end{aligned} \tag{4.8}$$

4.2.1 Inferring hidden variables

For a given observed shape, we would like to know what kind of transformation makes it as the observed shape. And it is also useful to know overall how well the components combination matches the observed shape. We call the first as the "responsibility" of

the transformation and the second as the "likelihood" of the shape. The responsibilities are the posterior probabilities of the transformation indices, that is:

$$\begin{aligned} P(l|X) &= \tfrac{p(X,l)}{p(X)} \\ p(X) &= \textstyle\sum_{l=1}^{L} p(X,l) \end{aligned} \tag{4.9}$$

where $p(X)$ is the likelihood. To computer the above two terms, we can first obtain $p(X,l)$ by integrating $p(X,l,Z,b)$ over Z and b:

$$\begin{aligned} P(X|l) &= \textstyle\int_Z \int_b \lceil Z \lceil b p(X,l,z,b) \\ &= p_l \mathcal{N}(X; R_l\mu, R_l(PP'+\Phi)R_l' + \Psi) \\ &= p_l \textstyle\prod_{i=1}^{L} \mathcal{N}(X_i; R_l\mu, R_l(PP'+\Phi)R_l' + \Psi) \end{aligned} \tag{4.10}$$

For a given shape X, the posterior distribution over the component activities is given by:

$$p(b|x) = \sum_{l=1}^{L} p(b|X,l)P(l|X) \tag{4.11}$$

Given the transformation l and X, the expectation of b is:

$$E[b|X,l] = \beta_l P' \Phi^{-1}[\Omega_l R_l' \Psi^{-1} X - (I - \Omega_l \Phi^{-1})\mu] \tag{4.12}$$

where:

$$\begin{aligned} \Omega_l &= cov(Z|X,b,l) = (\Phi^{-1} + R_l'\Psi^{-1}R_l)^{-1} \\ \beta_l &= (I + P'\Phi^{-1}P - P'\Phi^{-1}\Omega_l\Phi^{-1}P \end{aligned} \tag{4.13}$$

4.2.2 EM Algorithm

In the expectation step, the sufficient statistics required in the maximization step are computed. For each l , we compute:

$$E[Z|X,l] = \mu + \Omega_l R_l' \Psi^{-1}(X_l - R_l\mu) + \Omega_l \Psi^{-1} P \beta_l P' \Phi^{-1} R_l' \Psi^{-1}(X_l - R_l\mu) \tag{4.14}$$

where X_t means t^{th} input training shape, $t \in 1...s$. And the following expectation is also computed:

$$E[Z - Pb|X_t] = \sum_{l=1}^{L} P(l|X_t)(E[Z|X_t, l] - PE[b|X_t, l]) \tag{4.15}$$

and

$$\begin{aligned} &E[(Z-\mu)\circ(Z-\mu)|X_t, l] = (E[Z|X_t, l]-\mu)\circ(E[Z|X_t, l]-\mu) \\ &+diag(\Omega_l) + diag(\Omega_l\Phi^{-1}P\beta_l P'\Phi^{-1}\Omega_l) \end{aligned} \tag{4.16}$$

$$E[(Z-\mu)b'|X_t, l] = (E[Z|X_t, l]-\mu)E[b|X_t, l]' + \Omega_l\Phi^{-1}P\beta_l \tag{4.17}$$

are computed to obtain following expectations, where $a \circ b$ means the element-wise product of a and b:

$$\begin{aligned} &E[(Z-\mu-Pb)\circ(Z-\mu-Pb)|X_t] = \textstyle\sum_{l=1}^{L} P(l|x_t)\cdot\{E[(Z-\mu)\circ(Z-\mu)|X_t, l] \\ &+diag(P\beta_l P') - 2diag(PE[(Z-\mu)b'|X_t, l]' + (PE[b|X_t, l])\circ(PE[b|X_t, l])\} \end{aligned} \tag{4.18}$$

$$\begin{aligned} &E[(X_t - R_l Z)\circ(X_t - R_l Z)|X_t] = \textstyle\sum_{l=1}^{L} P(l|X_t)\cdot \\ &\{(X_t - R_l E[Z|X_t, l])\circ(X_t - R_l E[Z|X_t, l]) \\ &+diag(R_l\Omega_l R_l') + diag(R_l\Omega\Phi^{-1}P\beta_l P'\Phi^{-1}\Omega_l R_l')\} \end{aligned} \tag{4.19}$$

$$E[(Z-\mu)b'|X_t] = \sum_{l=1}^{L} P(l|X_t)E[(Z-\mu)b'|X_t, l] \tag{4.20}$$

$$E[bb'|X_t] = \sum_{l=1}^{L} P(l|X_t)(\beta_l + E[b|X_t, l]E[b|X_t, l]') \tag{4.21}$$

In maximization step, following parameters are updated as:

$$\tilde{\mu} = \frac{1}{s}\sum_{t=1}^{s} E[Z - Pb|X_t] \tag{4.22}$$

$$\tilde{\Phi} = diag(\frac{1}{s}\sum_{t=1}^{s} E[(Z-\mu-Pb)\circ(Z-\mu-Pb)|X_t]) \tag{4.23}$$

$$\tilde{\Psi} = diag(\frac{1}{s}\sum_{t=1}^{s} E[(X_t - R_l Z) \circ (X_t - R_l Z)|X_t]) \tag{4.24}$$

$$\tilde{P} = \frac{1}{s}\sum_{t=1}^{s} E[(Z-\mu)b'|X_t] \cdot (\frac{1}{s}\sum_{t=1}^{s} E[(bb'|X_t])^{-1} \tag{4.25}$$

$$p_l = \frac{1}{s}\sum_{t=1}^{s} P(l=1|X_t) \tag{4.26}$$

4.2.3 Experiments and results

This algorithm is first implemented in a simple test. In this toy implementation, the input data have 20 training shape, which have different orientations and two different shapes, one is a triangle and the other is a hexagon:

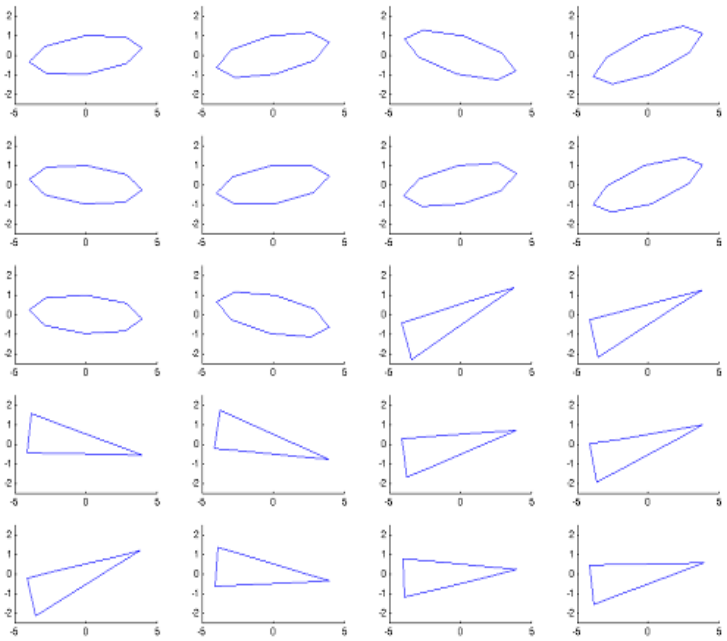


Figure 4.4: A set of training shapes of the simple test.

Figure 4.6 is a set of training shapes of the simple test. We applied both PCA and TCA to this data set, and get following results, left side is the result of PCA and

the right one is of TCA. we can find TCA method can eliminate the effect brought by rotation, and capture the more interesting structure information.

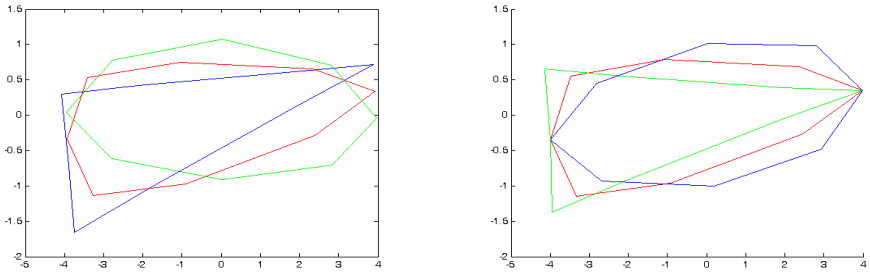


Figure 4.5: results of PCA(left) and TCA(right). The red shape is the mean shape, the blue and green are mean shape plue or minus a certain amount of the main component.

Then we applied this rotation invariant shape modeling method to some real heart shape data, which I arterially added some rotation distortion to so that make the results more obvious. The following is the input training data set. They are translated and scaled before the training process. We can find they have different orientations:

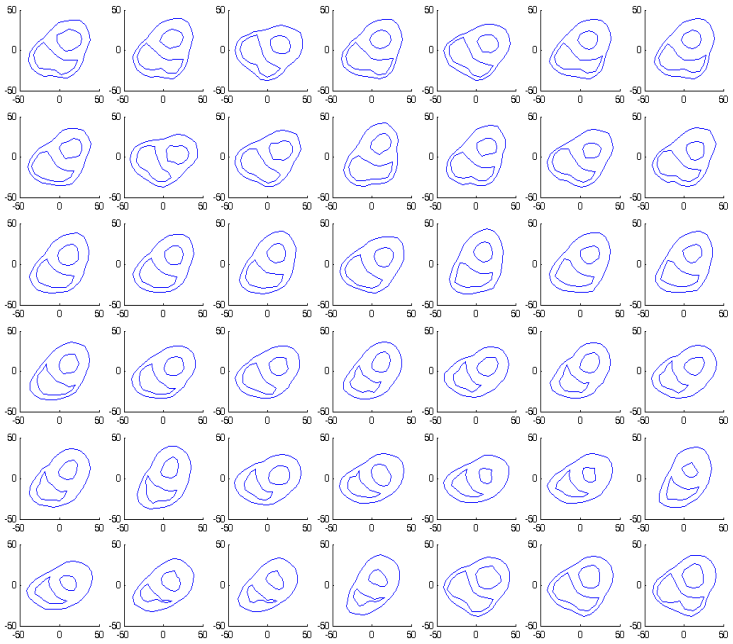
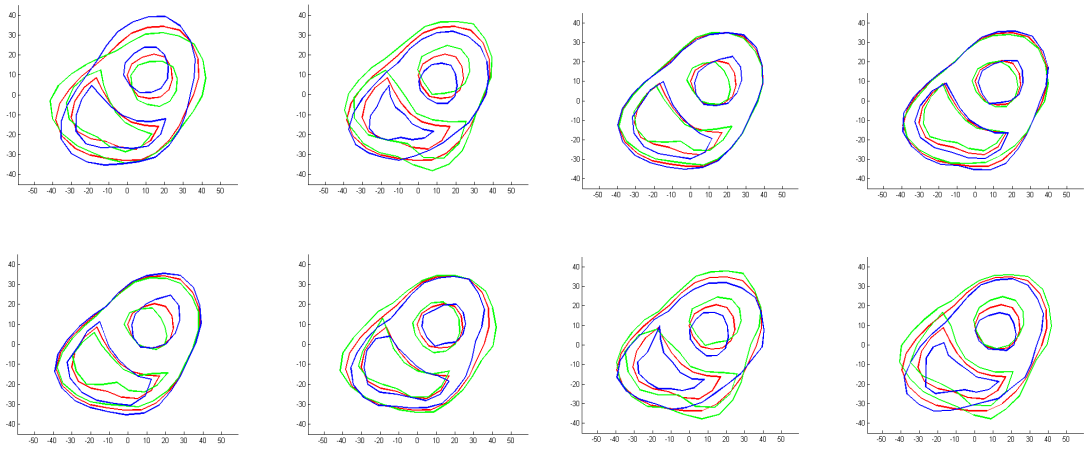


Figure 4.6: The input heart shape training data.

Again, PCA results are used to compare with the results of the rotation invariant

shape model. For a training set of totally 42 different shapes, each shape described by 50 landmark points, and the possible rotation matrix consists of totally 61 possible rotation matrixes, the computation time of the rotation invariant method is about 10 minutes.



(a) (b)

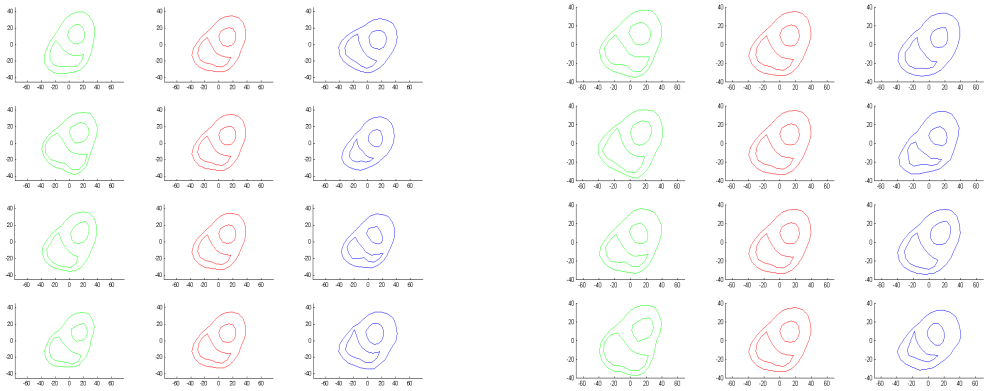
Figure 4.7: (a) is the result of PCA; (b) is from TCA. The red curves are the mean shape, and the green and blue are mean shape plus or minus a small amount of a certain component vector. Here the main 4 components are displayed for the 2 methods. We can find in PCA, the 1st component partially describes the variation of rotations. At the same time, we find the 4th component of PCA has also a variation of rotations. This illustrates that the PCA result is corrupted by the orientation variations. However in TCA, we can find the results have no orientation variations and the shape variation information are captured well and not blurred.

Here are some more results in figure 4.8.

We also generate some simulated heart shapes from PCA and TCA in figure 4.9.

The TCA method jointly learns components from data and normalizes for transformations. It especially works well in the data set that is not aligned well before training, which may introduce blurred or less interesting components in conventional PCA or FA methods, because PCA and FA methods are sensitive to the initialization of the input training data.

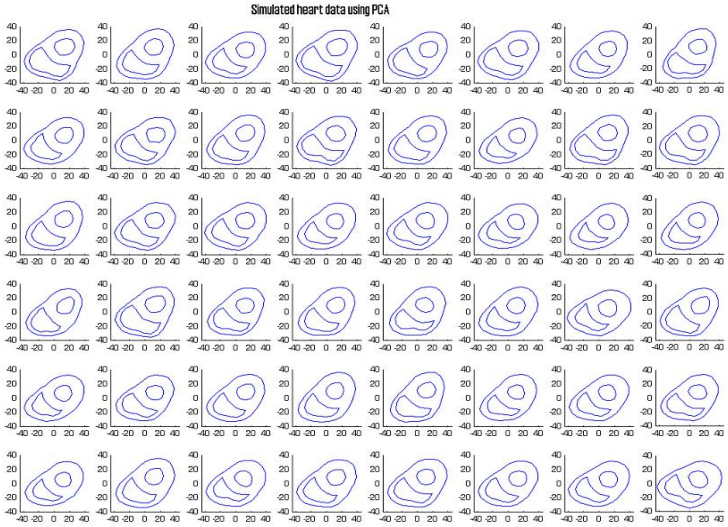
After we find the image forces at each landmark point, from section 4.3, the rotation invariant shape model evolves iteratively. In each iteration, the model deforms under the influence of the image forces to a new location; the image forces are then calculated



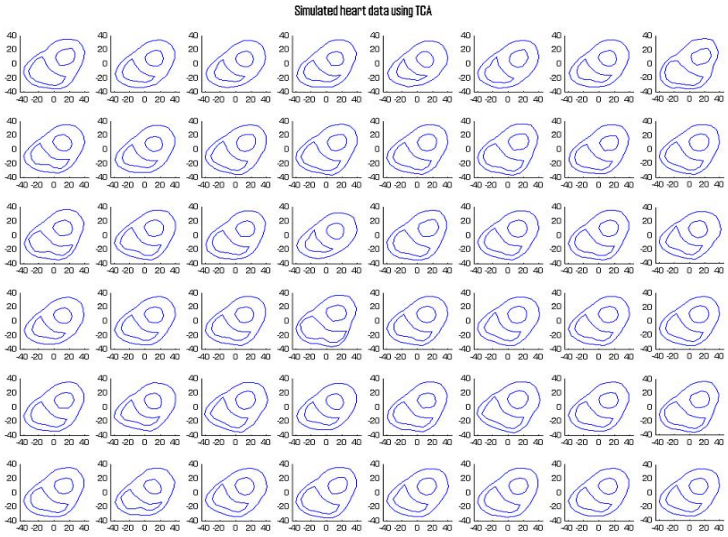
(a) (b)

Figure 4.8: Component analysis using PCA and TCA. Left is PCA, right is TCA.

at the new locations before the next iteration.



(a)



(b)

Figure 4.9: Simulated results of PCA and TCA. We can find in PCA, the simulated shapes have orientation variations, while the TCA results have the same orientation.

4.3 Local Appearance Modeling Through Adaboost

There has been some previous research on linear shape model segmentation methods based on local features modeling. In [20], a statistical analysis was performed, which used sequential feature forward and backward selection to find the set of optimal local features. In [21], an EM algorithm was used to select Gabor wavelet-based local features. In [22], an Adaboost learning method was proposed to find the optimal edge features. In our method, similarly using Adaboost, the rotation invariant shape model deforms based on a more *complex* rule, which is learned from the local appearance, not only of the edges, but also ridges and tagging line breakpoints.

4.3.1 Feature Design

To capture the local appearance characteristics, we designed three different kinds of steerable filters. We use the derivatives of a 2D Gaussian to capture the edges, we use the second order derivatives of a 2D Gaussian to capture the ridges, and we use half-reversed 2D Gabor filters to capture the tagging line breakpoints.

Assume $G = G((x-x_0)\cos(\theta), (y-y_0)\sin(\theta), \sigma_x, \sigma_y)$ is an asymmetric 2D Gaussian, with effective widths σ_x and σ_y, a translation of (x_0, y_0) and a rotation of θ. We set the derivative of G to have the same orientation as G:

$$G' = G_x \cos(\theta) + G_y \sin(\theta) \tag{4.27}$$

The second derivative of a Gaussian can be approximated as the difference of two Gaussians with different σ. We fix σ_x as the long axis of the 2D Gaussians, and set $\sigma_{y2} > \sigma_{y1}$. Thus:

$$G'' = G(\sigma_{y1}) - G(\sigma_{y2}) \tag{4.28}$$

In the previous two equations, we set $x_0 = 0$, and tune y_0, θ, σ_x, σ_y, σ_{y1} and σ_{y2} to generate the desired filters.

The half-reversed 2D Gabor filters are defined as a 2D sine wave multiplied with

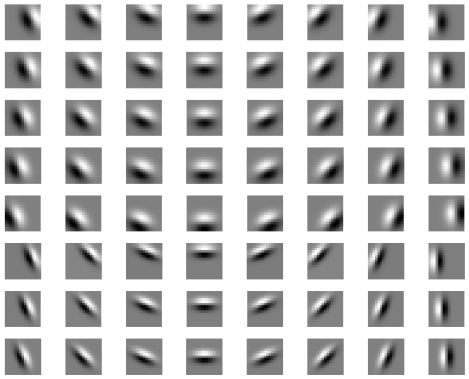


Figure 4.10: The first derivatives of Gaussian used for edge detection.

the 2D derivative of a Gaussian:

$$F = G'(x, y) \cdot \mathbb{R}\{e^{-j[\phi+2\pi(Ux+Vy)]}\} \tag{4.29}$$

where G' is the derivative of a 2D Gaussian. U and V are the frequencies of the 2D sine wave, $\psi = arctan(V/U)$ is the orientation angle of the sine wave, and ϕ is the phase shift. We set $x_0 = 0$, $\sigma_x = \sigma_y = \sigma$,$-45° \leq \psi - \theta \leq 45°$, and tune y_0, θ, σ, ϕ, U and V to generate the desired filters.

For a 15x15 sized window, we designed 1840 filters in total. See Figure 4.10, 4.11, 4.12 for some sample filters.

4.3.2 Adaboost Learning

In the learning section, each training image is scaled proportionally to the scaling of its contours. At each landmark point of the contours, a small window (15x15) around it was cut out as a positive appearance training sample for this particular landmark point. Then along the normal of the contour, on each side of the point, we cut out two 15x15-sized windows as negative appearance training samples for this particular landmark point. Thus for each training image, at a particular landmark point, we got one positive sample and four negative samples (shown in Figure 4.13(a).) We also randomly selected a few common negative samples outside the heart or inside the blood

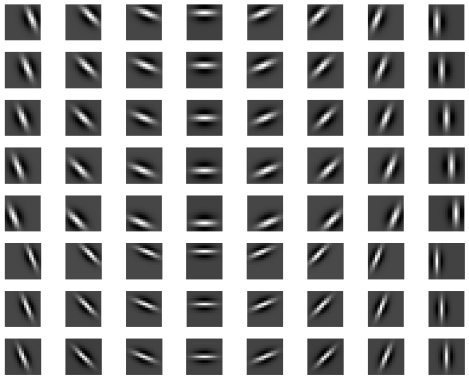


Figure 4.11: The second derivatives of Gaussian used for ridge detection.

area, which are suitable for every landmark point. For image contrast consistency, every sample was histogram equalized.

The function of the Adaboost algorithm [23, 24] is to classify the positive training samples from the negative ones by selecting a small number of important features from a huge potential feature set and creating a weighted combination of them to use as an accurate strong classifier. During the boosting process, each iteration selects one feature from the total potential features pool, and combines it (with an appropriate weight) with the existing classifier that was obtained in the previous iterations. After many iterations, the weighted combination of the selected important features can become a strong classifier with high accuracy. The output of the strong classifier is the weighted summation of the outputs of each of its each selected features, or, the weak classifiers:

$$F = \Sigma_t \alpha_t h_t(x) \tag{4.30}$$

where α are the weights of weak classifiers, and h are the outputs of the weak classifiers.

We call F the boundary criterion. When $F > 0$, Adaboost classifies the point as being on the boundary. When $F < 0$, the point is classified as off boundary. Even when the strong classifier consists of a large number of individual features, Adaboost encounters relatively few overfitting problems [25]. We divided the whole sample set into one training set and one testing set. The function of the testing set is critical. It

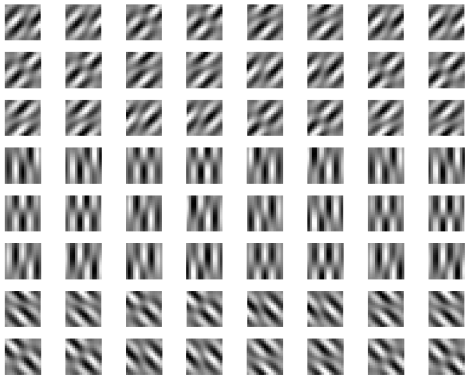


Figure 4.12: The half-reversed Gabor filters used for tag line breakpoint detection.

gives a performance measure and a confidence level that tells us how much we should trust its classification result. Figures 4.14 4.15 shows the learning error curve versus the boosting iteration numbers at two selected landmark points. Remarkably, every landmark point i has its own α, h and F_i.

4.3.3 Segmentation

In the segmentation stage, we first select a initial location and scale, and then overlay the mean shape $\bar{X}$, which is obtained from the linear shape model, onto the task image. In section 4.4 we describe an automatic initialization method.

At a selected landmark point i on the shape model, we select several equally spaced points along the normal of the contour on both sides of i, and use their F values to examine the corresponding windows centered on these points. In [25], a logistic function was suggested to estimate the relative boundary probabilities:

$$Pr(y = +1|x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}} \tag{4.31}$$

We find a point j whose test window has the highest probability of being on the heart boundary. Thus an image force $\vec{f}$ should push the current landmark point i toward j. Recall that, as discussed in the previous subsection, Adaboost gives the errors of the

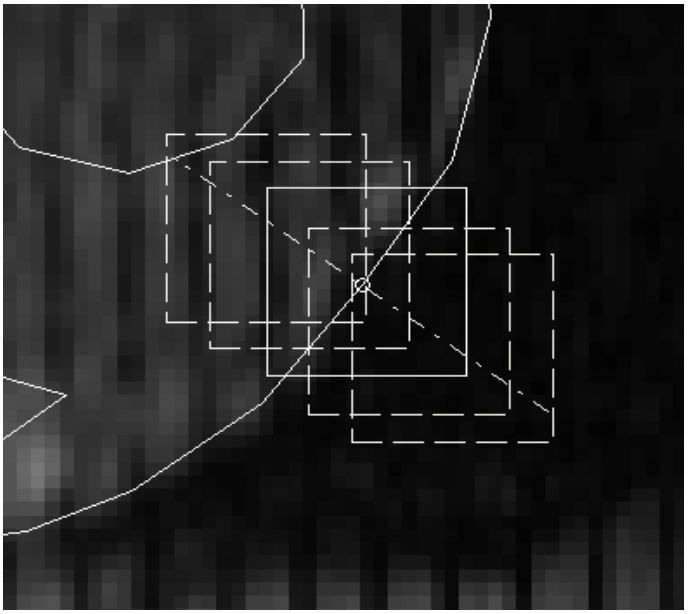


Figure 4.13: The method of setting the training data. The solid box is the positive sample around the landmark points. The four dash-line boxes along the normal are the negative samples. This way of setting the negative samples is chosen to make the classifier more adaptive to the particular landmark position.

testing data e_i. We define the confidence rate as:

$$c_i = \ln \frac{1}{e_i}; \tag{4.32}$$

Intuitively, when c_i is big, we trust its classification and increase the image force $\vec{f}$, and *vice versa*. Thus we define the image force at landmark point i as:

$$\vec{f} = \mu \cdot \frac{[\vec{x}(j) - \vec{x}(i)] \cdot c(i)}{||\vec{x}(j) - \vec{x}(i)||_2} \tag{4.33}$$

where μ is a scale as a small step size.

The detail algorithm to update the parameters of the linear shape model with the image force $\vec{f}$ can be found in [6].

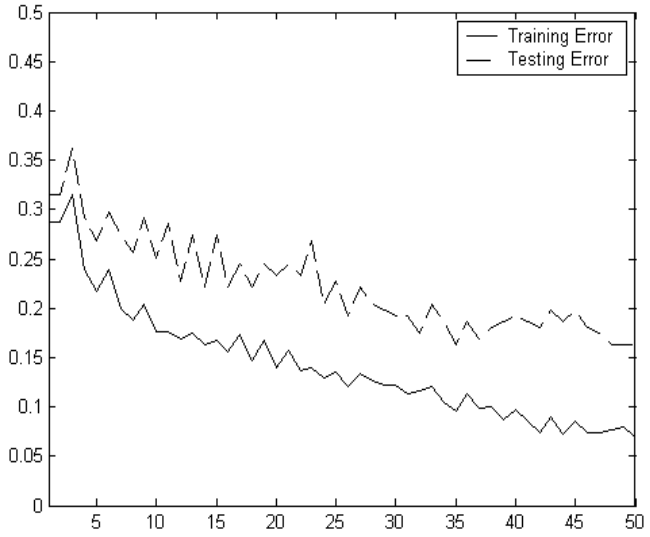


Figure 4.14: The training error (solid lines) and testing error (dash lines) of a landmark point on the LV versus Adaboost iteration times. Note how the training and testing error decrease as Adaboost iterates.

4.4 Automatic Initialization

Detection of the organ of interest in a medical image is often the first step of many medical image processing tasks. When we are dealing with medical image processing tasks, such as segmentation, registration, and tracking, first of all, we need to know where the interested organ locates and how much areas the organ covers. Usually this detection task is done manually by human experts clicking on the organ location or cropping out the region of interest.

To achieve automatic initialization for the following segmentation, our goal is to automatically detect the heart in the tagged MRI images. A closely related problem is face detection. We find face detection problem shares many similarities with our heart detection task. Usually there are a lot of variations among different faces, which come from different facial appearance, lighting, expression, etc. While in heart detection, we also have the same challenges: the heart has different tag pattern, shape, position, rotation, phase, etc. We can adopt the ideas from the face detection technique.

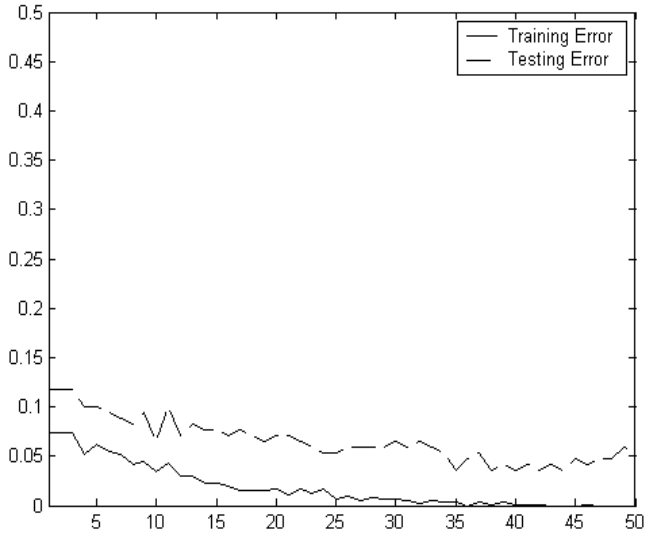


Figure 4.15: The training error (solid lines) and testing error (dash lines) of a landmark point on the Epi versus Adaboost iteration times. Note how the training and testing error decrease as Adaboost iterates. Also note the testing error of the LV landmark point is higher than this Epi landmark point: we are more confident of landmark point this Epi landmark point's classification result.

There are many existing face-detecting work. [26, 27] were using correlation-templates-based methods. [28] used view-based eigenspaces to reduce the high dimensional vector space of all possible face patterns to a low dimensional linear subspace. [29] modeled a deformable templates. Sung, at el [30] generated two distribution models of 'face-pattern' and 'non-face-pattern' from a set of training examples. The classifier is based on the difference feature vector which is computed between the local image pattern and the distribution-based model. Papageorgiou [31] set up an over-completed Haar wavelet representation of the object class. Then they reduced the dimension and select the most important features. They trained a support vector machine as the final classifier. Viola and Jones [32] used Adaboost method on an over-completed Haar-like features to generate an accurate strong classifier from a set of weak classifiers. They also implemented a cascade detection method to achieve high computation speed. Here we adopted Viola and Jones's Adaboost learning framework, because human's prior

knowledge tends to add bias constraints on the detection model, and we hope the algorithm totally learn its rules from the training data without any *a priori*. Furthermore, their implementation is accurate and relatively fast.

As mentioned in the previous section, Adaboost algorithm [23] selects a small number of important features from a huge feature set and generates an accurate strong classifier. During the boosting process, each iteration selects one feature from the total potential features, and weighted combines it with the existing classifier that obtained in the previous iterations. After many iterations, the weighted combination of the selected important features turns to be a strong classifier with high accuracy.

4.4.1 Features

The similar Haar wavelet function in Viola's paper are used as features of the weak classifiers, as seen in Fig. 4.16. The filtered result of a feature is by convolving the input image with the feature window. These rectangle features have many advantages. First, they are able to encode ad-hoc domain knowledge that is difficult to learn using pixel-based features. Second, they can be computed very fast using *Integral Image*, which is much faster than convolution.

Figure 4.16: Example features. They are two-rectangle and four-rectangle features with different orientations. The white-colored pixels equal 1, the black-colored pixels equal -1, and the gray equal 0. Totally there are 62208 features in a (24x24) sized image.

The Integral Image at pixel x, y is the summation of the pixels above and to the left of x, y, including x, y itself in the original input image, as seen in Fig. 4.17, 4.18:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{4.34}$$

Where ii is the integral image and i is the original image.

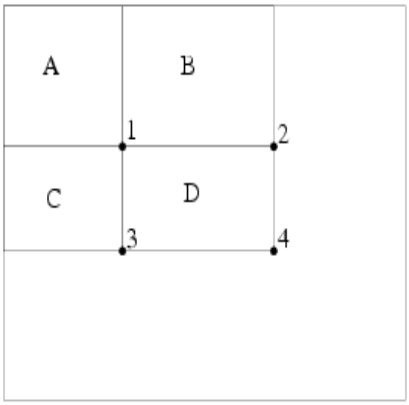


Figure 4.17: An illustration of integral image. Sum within rectangle $D = ii_4 + ii_1 - ii_2 - ii_3$.

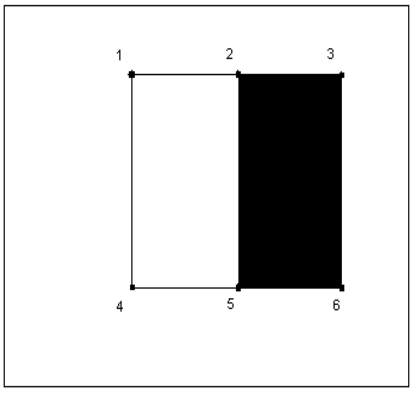


Figure 4.18: A feature example, whose filtered result $= ii_5 + ii_1 - ii_2 - ii_4 - (ii_6 + ii_2 - ii_3 - ii_5)$.

4.4.2 The Attentional Cascade Detection

In the detection process, the input image is divided to a huge number of sub-images, *e.g.*, a 192x192 sized image is divided to more than 120,000 sub-images. If each sub-images goes through the strong classifier, which consists of hundreds or thousands of weak classifiers, the computation is expensive. Alternatively, I used attentional cascade technique. This technique is based on three facts: 1st, the first few features in the strong classifier have relatively lower classifying error; 2nd, we can lower the threshold of the

weak classifier to achieve a very low false negative rate at the expense of an increased false positive rate; 3rd, most of the sub-images don't contain any heart. (Actually there should be only one optimal detection). Thus by using the first few weak classifiers, we can reject most of the sub-images, which saves a lot of time. Figure 4.19 is an example of the attentional cascade algorithm.

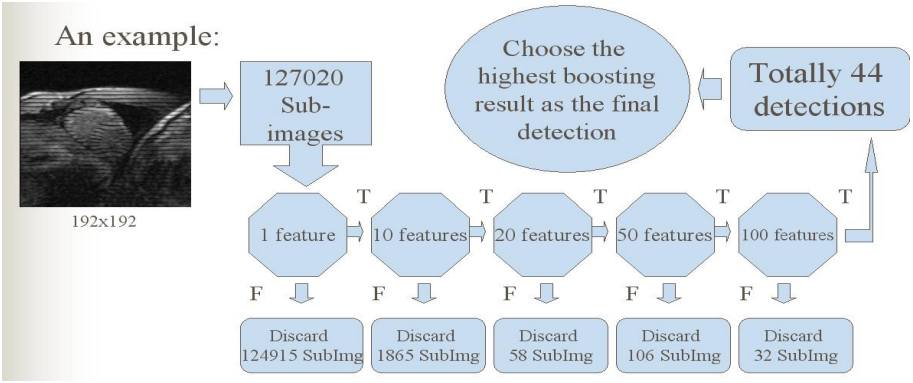


Figure 4.19: Five cascade stages with a total number of 100 features are used. At the first stage which consists of only one feature, 124915 out of 127020 candidate sub-images are rejected. At the second stage, 1865 sub-images are rejected, and so on. Finally in a certain image we have total 44 detections. The highest boosting result is chosen as the final detection. We can see during the first two stages, most sub-images are rejected, which makes the computation faster.

4.4.3 Experiments and Results

In the experiment, the training data consist of 297 heart images and 459 non-heart images; the testing data consist of 41 heart images and 321 non-heart images. All images are resized to 24x24 pixels and rotated to a same angle. Image intensities and contrasts are normalized. Figure 4.20 is a random sample from the heart training set.

The first few features Adaboost chooses are meaningful and easy to interpret. As shown in figure 4.21, the first five features mostly represent the boundary information of the heart, because the intensity inside the heart region varies too much and has no obvious patterns. Figure 4.22, 4.23 and 4.24 are a comparison of the error of the weak classifier, and the error of the strong classifier performing on the training data and the testing data.

After the final stage of the attentional cascade, the final detector usually produces

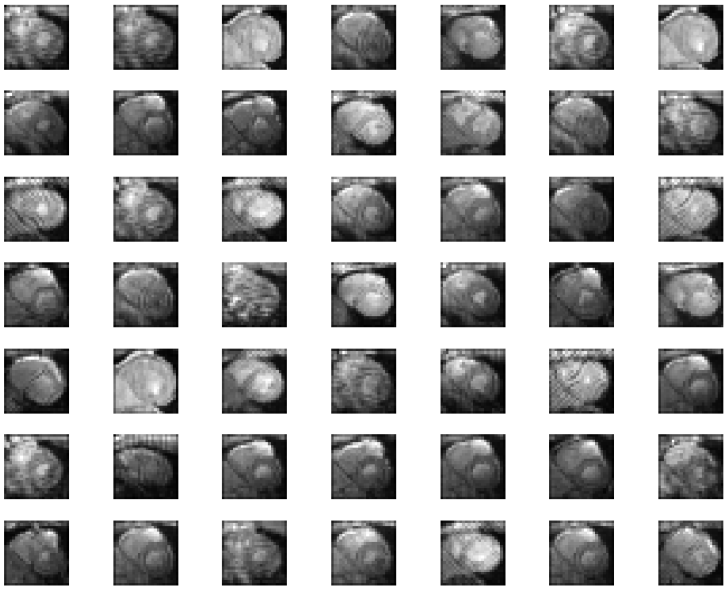


Figure 4.20: A random sample of the heart training set.

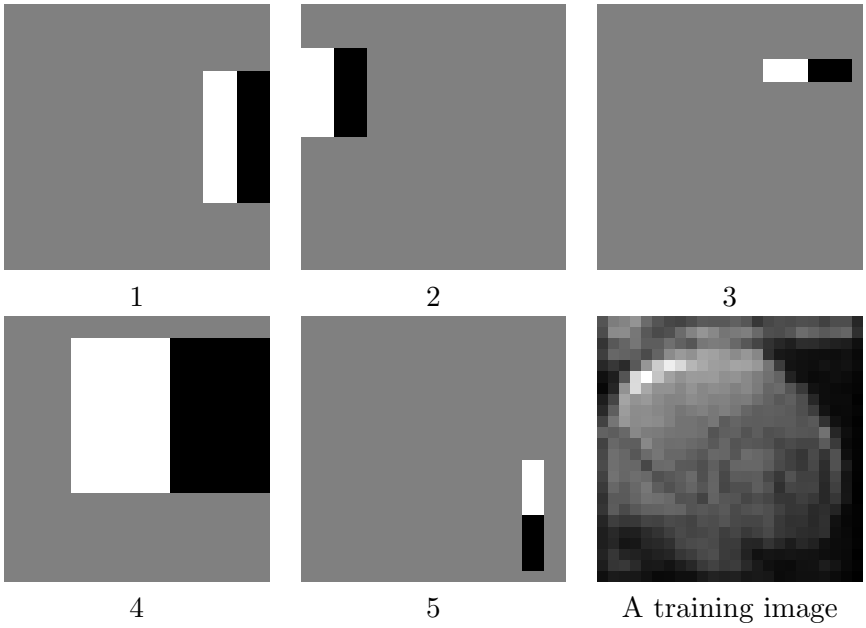


Figure 4.21: The first five features Adaboost selects comparing with a training image.

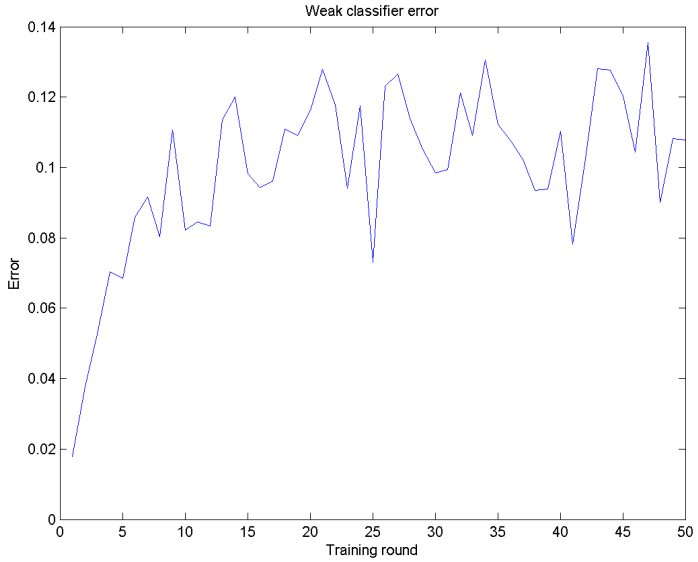


Figure 4.22: This figure shows the error of the weak classifier that Adaboost selects at each boosting round. The error increases non-monotonously as the distribution of the training examples become more difficult to classify.

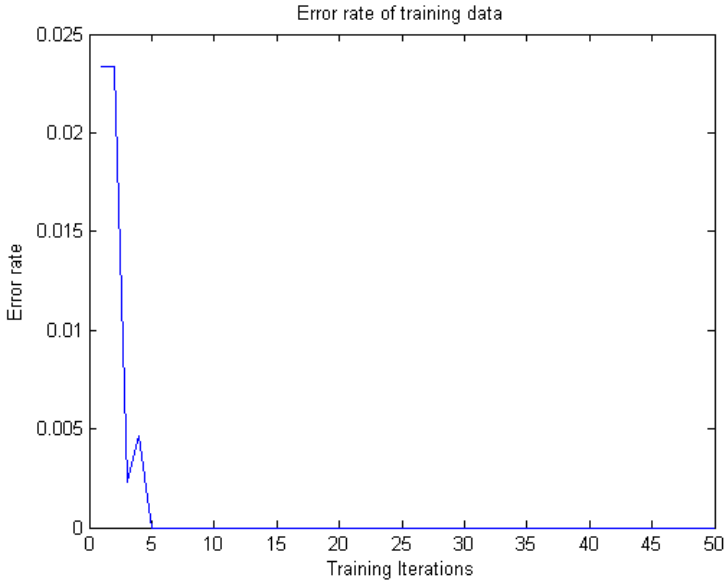


Figure 4.23: This figure shows the error of the strong classifier on the training data. The error drops to zero after five rounds.

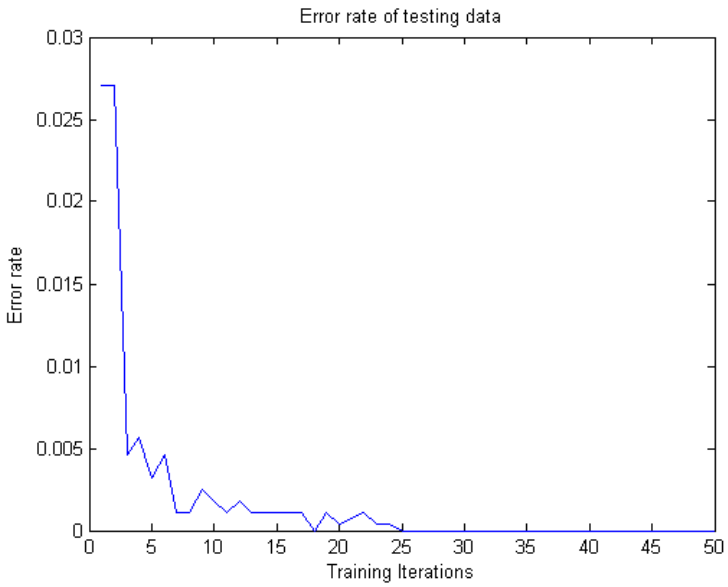


Figure 4.24: This figure shows the error of the strong classifier on the testing data. The testing error continues to decrease after the training error approaches zero, which means more iterations leads to larger margin and higher accuracy.

a number of detections, because this strong classifier is insensitive to small changes in translation and scale. However, as we know, there can be only one heart in a image. So in the case of multiple detections, we have to discard most of the detections and keep only one that we are most confident in. We select the detection with the highest boosting result, which means it has the maximum margin and Adaboost is most confident that it is a heart. Figure 4.25 shows some of our detection results. The blue boxes are the multiple detections. The yellow box is the final detection.

The face detection algorithm using Adaboost works quite well in our heart detection task. For the (41 positive + 321 negative)-sized testing data, our detector achieves a zero error rate.

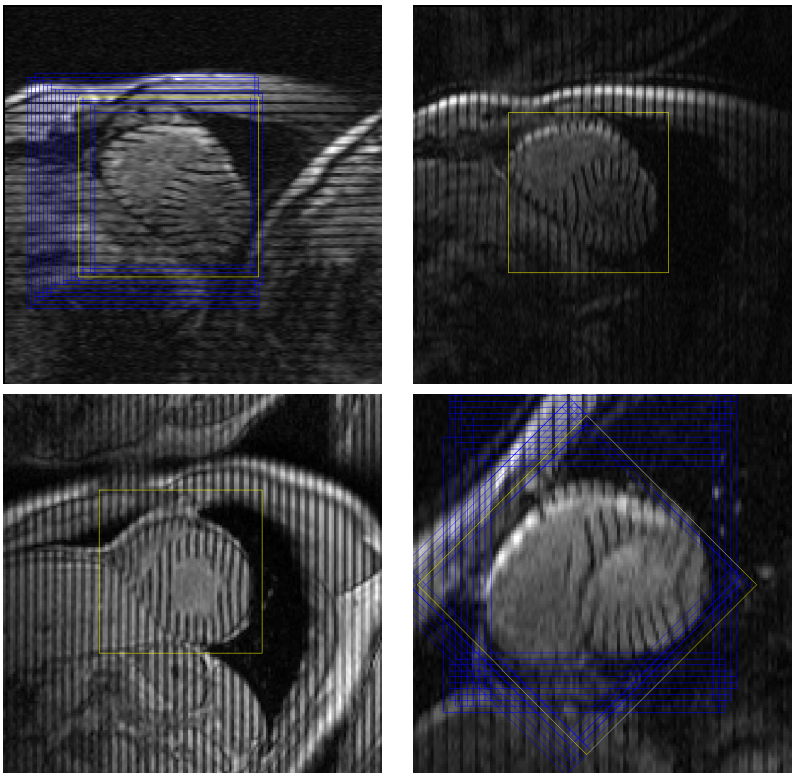


Figure 4.25: Four detection results. The first one shows that it has multiple detections and the yellow box is the final detection. The second and the third only show the final detections. The forth image shows detections at different rotation angles. The final detection is the rotated box which are most similar with the training data.

4.5 Results and Validation

We applied our segmentation method to two data sets, one from the same subject and with the same imaging settings as the training data, and the other a novel data from a different subject and with slightly different imaging settings. Each task image was scaled to contain a 80x80-pixel-sized heart before the segmentation. Each segmentation took 30 iterations to converge. Our experiment was coded in Matlab 6.5 and run on a PC with dual Xeon 3.0G CPUs and 2G memory. The whole learning process took about 20 hours. The segmentation process of one heart took 120 seconds on average. See Figure 4.26 for representative results.

For validation, we used the manual segmentation contours as the ground truth for the first data set. For the second data set, since we don't have indenpendent manual contours, we used cross validation, because we know that at the same position and phase, the heart shapes in the vertical-tagged and horizontal-tagged images should be similar. We denote the ground truth contours as T and our segmentation contours as S. We defined the average error distance as $\bar{D}_{error} = mean_{s_i \in S}(min||T - s_i||_2))$. Similarly the cross distance is defined as $\bar{D}_{cross} = mean_{s_i^{vertical} \in S^{vertical}}(min||S^{horizontal} - s_i^{vertical}||_2))$. In a 80x80 pixel-sized heart, the average error distances between the automatically segmented contours and the contours manually segmented by the expert were: $\bar{D}_{error}(LV) = 1.12$ pixels, $\bar{D}_{error}(RV) = 1.11$ pixels, $\bar{D}_{error}(Epi) = 0.98$ pixels. In the second dataset, the cross distances are: $\bar{D}_{cross}(LV) = 2.39$ pixels, $\bar{D}_{cross}(RV) = 1.40$ pixels, $\bar{D}_{cross}(Epi) = 1.94$ pixels. The cross distance arises in part from underlying misregistration between the (separately acquired) horizontal and vertical images. Thus, the true discrepancy due to the segmentation should be smaller.

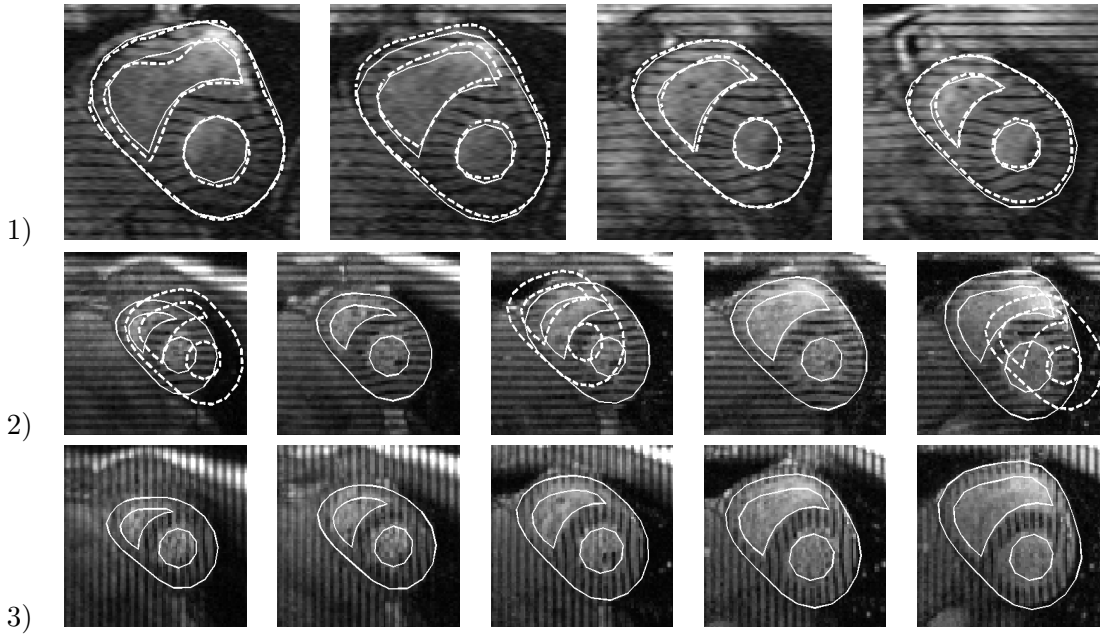


Figure 4.26: The first row of images comes from the the first dataset. The solid contours are from our automatic segmentation method; the dashed contours are manual. Notice that the papillary muscles in LV are excluded from the endocardium. The second and third rows are from the second dataset. Manual contours are not available for this dataset, so we compare our segmentation results between the the horizontal and vertical tagged images that are at same position and phase. Qualitatively, the contours are quite consistent, allowing for possible misregistration between the nominally corresponding image sets. In (2a), (2c) and (2e) the dashed contours are the initialization shapes, while the final contours are solid. Although the initialization is fay away from the target, the shape model moves and converges well to the target.

Chapter 5

Conclusion

5.1 Discussion

Because medical image segmentation needs high level medical and anatomic knowledge, model-based segmentation methods are highly desirable. In this thesis, we first give a short survey of current approaches of medical image segmentation. Then we specifically develop appearance and shape models for two different segmentation tasks, of which one is 3D MR colonography and the other is 2D cardiac tagged MRI. These models are either obtained from visual observation and prior human expertise, or from certain automatic machine learning methods. For 3D MR colonography, we manually build a flexible tube-shaped shape model and an intensity model that combines both edge and region information. The 3D segmentation is formulated in a non-parametric tracking framework. For 2D tagged MRI, we learn the shape and local appearance model from a training set. This shape model is a linear rotation invariant model, on which each control point has its own local appearance model. This appearance model is learned from an Adaboost process, which integrates many features. In each application, besides the models, we also give complete details in solving the segmentation problems, such as how we correct the MR image intensity inhomogeneity and how we automatically initialize the segmentation.

From the experimental results, we find that both model-based methods perform well on real medical image data. They are robust for noisy images. However they also have limitations.

For the 3D colonography segmentation, the shape of the colon sometimes is too complicated for a tube-shaped model, or even a bendable tube-shaped model to represent. Our current method has difficulties when the colon's shape changes rapidly. From

this point of view, the shape model may put too much constraints to the segmentation. In the future work, we could make this shape model more flexible and thus relax the shape constraints.

In the 2D cardiac tagged MRI segmentation, the size of training data set is critical for the segmentation performance, since all the models are obtained from learning. We find that if the segmentation method is applied to images at phases or positions that are not represented in the training data, the segmentation process tends to get stuck in local minima. Thus the training data need to be of sufficient size to cover all possible variations that may be encountered in practice.

5.2 Possible Future Work

We find in both applications, a main limitation of model-based segmentation methods is their inflexibility. Models tends to constrain the segmentation by certain prior knowledge. But when new image data come in, because of biological diversity, it is dangerous to strictly rely on the old model. There are several possible way to solve this problem in the future work. First, we can try make a better model, which can more precisely describe the anatomic structure. For manual parameterized model design, we may add more parameters. For learning approaches, we may increase the training size, or choose more complicated shape model, such as nonlinear manifold model. Second, we can improve the learning scheme. When new data come in, the learning approach should be able to augment them to form a new better model. Third, we may add some noise to the model to make it more flexible.

References

[1] Forsyth, D., Ponce, J.: Computer Vision - A Modern Approach. Prentice Hall (2003)

[2] Qian, Z., Huang, X., Metaxas, D., Axel, L.: Robust segmentation of 4d cardiac mri-tagged images via spatio-temporal propagation. In: Proceedings of SPIE Medical Imaging: Physiology, Function, and Structure from Medical Images. Volume 5746. (2005) 580–591

[3] McInerney, T., Terzopoulos, D.: Deformable models in medical image analysis: a survay. Medical Image Analysis **1** (1996) 91–108

[4] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int'l Journal of Computer Vision **1** (1988)

[5] Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Pro. of Fifth International Conference on Computer Vision. (1995) 694–699

[6] Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models - their training and application. Computer Vision and Image Understanding **61** (1995) 38–59

[7] Canny, J.: A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence **8** (1986) 679–698

[8] Xu, C., Prince, J.: Gradient vector flow: A new external force for snakes. In: IEEE Conf. on Computer Vision and Pattern Recognition. (1997) 66–71

[9] Dubes, R., Jain, A., Nadabar, S., Chen, C.: MRF model-based algorithm for image segmentation. In: IARP Int'l Conf. on Pattern Recognition. (1990)

[10] Jones, T., Metaxas, D.: Automated 3D segmentation using deformable models and fuzzy affinity. In: Proc. of Information Processing in Medical Imaging. (1997) 113–126

[11] Huang, X., Metaxas, D., Chen, T.: Metamorphs: Deformable shape and texture models. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (2004) 496–503

[12] Geenen, R., et al: CT and MR Colonography: Scanning techniques, postprocessing, and emphasis on polyp detection. Radiographics **24** (2003)

[13] Lauenstein, T.C., et al: MR Colonography with barium-based fecal tagging: Initial clinical experience. Radiology **223** (2002) 248–254

[14] Salvado, O., Hillenbrand, C., Zhang, S., Wilson, D.: Method to correct intensity inhomogeneity in mr images for atherosclerosis characterization. Medical Imaging, IEEE Transactions on **25** (2006) 539–552

[15] Styner, M., Brechbuhler, C., Szekely, G., Gerig, G.: Parametric estimate of intensity inhomogeneities applied to MRI. Medical Imaging, IEEE Transactions on **19** (2000) 153–165

[16] Axel, L., Costantini, J., Listerud, J.: Intensity correction in surface-coil mr iamging. American Journal of Radiology **148** (1987) 418–420

[17] Montillo, A., Metaxas, D., Axel, L.: Automated segmentation of the left and right ventricles in 4d cardiac spamm images. In: Medical Imaging Computing and Computer-Assisted Intervention. (2002) 620–633

[18] Huang, X., Li, Z., Metaxas, D.N.: Learning coupled prior shape and appearance models for segmentation. In: MICCAI (1). (2004) 60–69

[19] Frey, B., Jojic, N.: Transformed component analysis: Joint estimation of spatial transformations and image components. In: IEEE Int'l Conf. on Computer Vision. (1999)

[20] Ginneken, B.V., Frangi, A.F., Staal, J.J., et al: Active shape model segmentation with optimal features. IEEE Trans. on Medical Imaging **21** (2002)

[21] Jiao, F., Li, S., Shum, H., Schuurmans, D.: Face alignment using statistical models and wavelet features. In: IEEE Conf. on CVPR. Volume 1. (2003) 321–327

[22] Li, S., Zhu, L., Jiang, T.: Active shape model segmentation using local edge structures and adaboost. In: Medical Imaging Augmented Reality. (2004)

[23] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory. (1995) 23–37

[24] Schapire, R.E.: The boosting approach to machine learning: An overview. In: In MSRI Workshop on Nonlinear Estimation and Classification. (2002)

[25] R. E. Schapire, Y. Freund, P.B., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. Annals of Statistics **26** (1998) 1651–1686

[26] Bichsel, M.: Strategies of Robust Objects Recognition for Automatic Identification of Human Faces. PhD thesis (1991)

[27] Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE Trans. on Pattern Analysis and Machine Intelligence **15** (1993) 1042–1052

[28] Pentland, A., Moghaddam, B., T., S.: View-based and modilar eigenspaces for face recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition. (1997)

[29] Yuille, A., Hallinan, P., Cohen, D.: Face recognition: Features versus templates. Int'l Journal of Computer Vision **15** (1992) 99–111

[30] Poggio, T., Sung, K.K.: Example-based learning for view-based human face detection. In: Proc. of the ARPA Image Understanding Workshop. II. (1997) 843–850

[31] Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: IEEE Int'l Conf. on Computer Vision. (1998)

[32] Viola, P., Jones, M.: Robust real-time object detection. Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, And Sampling. Vancouer, Canada, July 13 (2001)