

ULTRASONIC HEART IMAGE SEGMENTATION USING ACTIVE CONTOUR  
MODEL

by

Jiuyu Sun

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

©Jiuyu Sun, 2014

# Abstract

This thesis is concerned with the ultrasonic heart image segmentation problem using parametric active contour model. Most of the existing parametric models consider only either the edge or the regional information. In this thesis, we propose a new parametric active contour model considering both the edge and regional information. The model is based on the vector field convolution (VFC) model and the Chan-Vese model and thus it is called as VFCCV model. The VFCCV not only has a large capture range and the ability to detect concavity, but is also robust to noise, cluttered image background, and weak boundary. Experiments on synthetic images, heart ultrasound images and MRI images show VFCCV has better performance than either VFC or Chan-Vese, and a state-of-the-art model combining edge and region information. In order to use VFCCV to segment the 3D ventricle ultrasound image, the 3D image is divided into 2D slices, and we segment each slice using VFCCV. However, the segmentation results of some slices may be inaccurate, while manual validation for every slice would be tedious. Therefore, we propose a novel automatic contour validation method, and introduce the validation step in the segmentation system. The validation step can judge whether the contour on each slice is accurate in order to make the whole process as automatic as possible while keeping a high segmentation accuracy. We extend an existing method that is based on the principal component analysis (PCA) for contour validation by performing the validation locally rather than globally in order to detect local errors. Experiments are conducted using the heart phantom ultrasound and human heart ultrasound images. Experimental results show that the VFCCV can generate correct segmentation results in most of the cases, and that the validation step can detect the errors when the contours are inaccurate. Also, the segmentation system is significantly faster than manual validation. For the heart phantom dataset, experiments show that the final ventricle volumes have high accuracy, and the system has high reproducibility.

# Acknowledgements

I would like to take this opportunity to express my sincere gratitude to those who have helped me through my Master studies.

Foremost, I would like to give my deepest appreciation to my supervisor, Professor Hong Zhang, the mentor who initiates me into the world of research. He has been patiently helping me grow, not only the research skills, but also the way of thinking and solving problems in life.

I would also like to express my gratitude to my co-supervisor, Professor Nilanjan Ray. His enthusiasm, inspiration, and patient guidance help me throughout the learning and research process.

I am also very grateful to my committee member, Professor Herb Yang, for spending time on reading my thesis and providing feedback on this research.

Also, thanks to Professor Harald Becher, Peter Wood, and Allen He in Mazankowski Alberta Heart Institute for their expertise and useful comments, and providing me the dataset. Thanks to Rajarshi Maiti for providing the user feedback and helpful remarks on the project. I would also like to acknowledge the support from all my colleagues and friends who have helped and encouraged me throughout the two years.

Last but not least, I would like to thank my beloved family for their unconditional support and endless love for my life, and their forgiveness of my inability to express my love.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Heart Image Segmentation . . . . .	1
1.2	Thesis Motivation and Objective . . . . .	2
1.3	Thesis Contribution . . . . .	3
1.4	Organization . . . . .	4
<b>2</b>	<b>Background and Related Work</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Parametric Active Contour Models . . . . .	6
2.2.1	Snake Model . . . . .	6
2.2.2	Balloon Force Model . . . . .	8
2.2.3	Gradient Vector Flow . . . . .	10
2.2.4	Generalized Gradient Vector Flow Model . . . . .	12
2.2.5	Other Parametric Active Contour Models . . . . .	13
2.3	Geometric Active Contours . . . . .	13
2.3.1	Curve Evolution-based Model . . . . .	14
2.3.2	Geodesic Active Contours . . . . .	16
2.3.3	Chan-Vese Model . . . . .	17
2.3.4	Other Geometric Active Contours . . . . .	19
2.4	Intelligent Scissors . . . . .	19
2.5	Summary . . . . .	21
<b>3</b>	<b>VFCCV Snake: A Novel Snake Model Combining Edge and Regional Information</b>	<b>25</b>
3.1	Introduction . . . . .	25
3.2	Related Works . . . . .	27
3.2.1	Vector Field Convolution . . . . .	27
3.2.2	Region-aided Geometric Snake . . . . .	29
3.3	VFC Chan-Vese (VFCCV) Model . . . . .	30
3.3.1	Propose the VFCCV Model . . . . .	30
3.3.2	Numerical Implementation . . . . .	32
3.4	Summary . . . . .	33
<b>4</b>	<b>3D Echocardiography Segmentation Embedded with Automatic Validation</b>	<b>34</b>
4.1	Introduction . . . . .	34
4.2	2D Contour Propagation and Validation . . . . .	37
4.2.1	Segmentation and Propagation . . . . .	37
4.2.2	Contour Validation . . . . .	38
4.3	Discussion . . . . .	42
4.4	Summary . . . . .	44

<b>5</b>	<b>Experimental Results</b>	<b>45</b>
5.1	Performance of VFCCV Model . . . . .	45
5.2	Performance of the Propagation and Validation Algorithm . . . . .	49
5.2.1	Dataset . . . . .	49
5.2.2	Performance of VFCCV in the Segmentation System . . . . .	50
5.2.3	Inaccurate Contours Detected by the Validation Step . . . . .	51
5.2.4	Accuracy of the Final Volume . . . . .	51
5.2.5	Reproducibility: Intra-Person Variability and Inter-Person Variability . .	54
5.2.6	Specificity and Sensitivity . . . . .	55
5.2.7	User Interface . . . . .	57
5.2.8	Performance on Human Echocardiography . . . . .	58
5.3	Summary . . . . .	59
<b>6</b>	<b>Conclusion and future work</b>	<b>62</b>
6.1	Conclusion . . . . .	62
6.2	Future work . . . . .	63
	<b>Bibliography</b>	<b>64</b>
<b>A</b>		<b>70</b>
<b>B</b>		<b>71</b>

# List of Tables

5.1	Quantitative Comparison of the Competing Algorithms in Terms of Pratt's Figure of Merit (FOM), Dice's coefficient (DC), and Hausdorff Distance (HD). Note that the higher the FOM or DC value is, the better the performance, and that the smaller the HD value, the better the performance. Each metric value is an average computed over 21 images that have ground truth segmentation and the metric values are followed by the respective standard deviations. . . .	49
5.2	Average accuracy of the volume for the five 4D images based on the five constant scalars respectively . . . . .	53
5.3	Coefficient of variability (CV) of the nine 3D images . . . . .	54
5.4	Percentage of difference of mean (MD) of the nine 3D images . . . . .	55
5.5	Average number of auto acceptances, manual acceptances, manual corrections, and inaccurate contours passing the validation of the five 4D images . . . . .	56
5.6	Sensitivity and specificity of the five image data and their average . . . . .	57

# List of Figures

1.1	Left ventricle in human echocardiography . . . . .	2
2.1	(a) The original image (b) Typical snake evolution (c) $-E_{ext}^4$ of the original image. $\sigma = 5, w_{edge} = 1$ . . . . .	8
2.2	(a) The U-shape image (b) The external force field of the snake model (c) The snake cannot evolve into the concave region . . . . .	9
2.3	(a) The U-shape image (b) The force field of the GVF model (c) The GVF snake can evolve into the concave region . . . . .	11
2.4	The evolution of the zero level set. The contour can split and detect the two objects. . . . .	15
2.5	The evolution of the level set function. The zero level set splits into two circles while the level set function is still a valid function. For a better view, the level set function is displayed upside down and thus has positive values inside and negative values outside. The zero level set is shown as the red curve. . . . .	23
2.6	The region-based energy is minimized when the contour locates on the object boundary. . . . .	24
2.7	A typical interface of the lasso tool, also called as intelligent scissors. . . . .	24
3.1	(a) VFC (b) Chan-Vese model implemented in parametric framework . . . . .	27
3.2	The vector field kernel. The centre point has the coordinate (0,0) . . . . .	28
4.1	(a) Without the validation step (b) With the validation step . . . . .	36
4.2	(a) The annular band for the whole contour (b) The annular band segment (c) Construct local and overlap segments from the whole unfolded annular band . . . . .	39
4.3	For a 4D image, the training data is taken from the first 3D image which is shown in red . . . . .	40
4.4	Find the mapped slice and take the neighbouring two slices . . . . .	40
4.5	(a) The eigenvalues (b) The derivative of the eigenvalues . . . . .	41
4.6	Flowchart of the validation step . . . . .	42
4.7	Red curves: intra-3D image difference. Blue curves: inter-3D image difference. (a) Difference between the means of each group of pattern images. (b) Angle between the hyperplanes defined by each group of PCs . . . . .	43
5.1	Comparison using image with U-shaped gradient object (a) VFC (b) Chan-Vese Snake (c) RAGS and (d) VFCCV . . . . .	46
5.2	Comparison on left ventricle of ultrasound image (a) VFC (b) Chan-Vese snake (c)RAGS and (d) VFCCV . . . . .	47
5.3	Comparison on human echocardiography (a) VFC (b) Chan-Vese snake (c) RAGS and (d) VFCCV . . . . .	47

5.4	Comparison on heart phantom images on ultrasound (a) VFC (b) Chan-Vese snake (c) RAGS and (d) VFCCV . . . . .	47
5.5	Comparison on left and right ventricles of MRI (a) VFC (b) Chan-Vese snake (c) RAGS and (d) VFCCV . . . . .	48
5.6	Our dataset consists of five 4D images . . . . .	50
5.7	One 4D image consists of nine 3D images indexed from two to ten . . . . .	50
5.8	Segmentation results on the heart phantom using VFCCV when all the contours are accepted by the validation step. Every five slice from the 54th slice to the 119th slice . . . . .	51
5.9	Before: the red curves show the inaccurate curves detected by the validation step. After: the contours after user correction . . . . .	52
5.10	The red curves show the volumes of the five image data respectively. The blue curve is the ground truth. The x-axis is the time sequence, and the y-axis shows the corresponding volumes. (a) The volumes are scaled based on the constant calculated using the first image. It obtains the minimum cumulative error among the five constants. (b) The scaler is calculated using the fourth image, which obtains the maximum cumulative error . . . . .	53
5.11	Bland-Altman Plot . . . . .	55
5.12	Interface . . . . .	58
5.13	Interface when the contour fails the validation . . . . .	59
5.14	A typical 3D visualization result . . . . .	60
5.15	Segmentation results on human ultrasonic heart images. The first row: along long-axis direction. The second row: along short-axis direction . . . . .	61
5.16	Before: the red curves show the inaccurate curves detected by the validation step. After: the contour after user correction . . . . .	61



# Chapter 1

## Introduction

In this thesis, we deal with image segmentation using active contour models, especially for ultrasonic heart images. In Section 1.1, we will first introduce the importance of heart image segmentation using computer-aided techniques. In Section 1.2, we will give the motivation and objective of this thesis. The contribution of this thesis is given in Section 1.3 and the thesis organization is given in Section 1.4.

### 1.1 Heart Image Segmentation

Medicine has been revolutionized by the fast development of medical imaging technologies [46]. Imaging modalities like ultrasound imaging, magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET), digital mammography, etc., provide an effective means for non-invasively mapping the anatomic structures [57]. Physicians and scientists can get information from medical images to diagnose and treat patients. Moreover, medical imaging is an important tool for structure visualization, treatment planning, surgical simulation, intra-operative navigation, and tracking the progress of disease [46]. With the rapid growth of the significance of medical imaging, the number and size of medical images are also increasing. This necessitates computer-assisted image processing technologies in order to facilitate and automate the analysis process and generate non-biased results.

Among all the computer-based technologies, medical image segmentation is an important task. In medical imaging, segmentation is important for image display, image measurements, feature extraction, etc. [59]. It is useful in applications like localization of pathology [84], study of anatomical structure [74], calculation of tissue volumes [35], disease diagnosis [66], treatment planning [28] and computer-integrated surgery [23] [57]. The principal goal of image segmentation is to partition an image into regions that have similar features [59], for example, partitioning an image into anatomical structures of bones, muscles, skin, or blood vessels. For another example, in cerebral images, the regions are partitioned into

white matter, grey matter and cerebrospinal fluid, fat and background. We can also use image segmentation techniques to delineate the anatomical structures and other regions of interest in order to, for example, detect the endocardium and segment the left ventricle.

Particularly, we only deal with heart images in this thesis, and mainly focus on the ultrasonic heart image segmentation problem. In cardiology, detecting the endocardium is significant for the estimation of left ventricle volume from which the ejection fraction<sup>1</sup> is derived, and also for the assessment of the regional wall motion [53]. Moreover, ventricular volume is an important diagnostic indicator of disease like dilated cardiomyopathy, etc. [33, 34, 73]. Also, the heart wall thickness is a crucial indicator to disease like heart muscle injury, hypertension, etc. These cardiac functional estimations all require accurate ventricular boundary detection, which leads to the requirement of computer-aided cardiology segmentation. A typical human left ventricle echocardiography image<sup>2</sup> is shown in Fig. 1.1. It is a 2D slice taken from a 3D echocardiography.

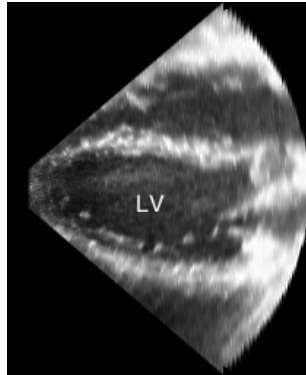


Figure 1.1: Left ventricle in human echocardiography

## 1.2 Thesis Motivation and Objective

As mentioned above, physicians and cardiologists need measurements like ventricle volume and ejection fraction to treat patients. Clearly, computer softwares can dramatically save their manual work. However, some of the existing softwares cannot generate satisfactory results in practise. This motivates us to design a robust ventricular segmentation method which will be helpful for physicians and cardiologists. Some of the image data we use is provided by the Echocardiography Laboratory, Mazankowski Alberta Heart Institute, and the scientists and cardiologists there need accurate ventricle volumes derived from these heart images.

<sup>1</sup>ejection fraction =  $\frac{\text{end-diastolic volume} - \text{end-systolic volume}}{\text{end-diastolic volume}} \times 100$

<sup>2</sup>The image is provided by Dr. Harold Becher of Echocardiography Laboratory, Mazankowski Alberta Heart Institute

Compared with 3D cardiography, 2D cardiography is easier to be displayed and observed, and certain 2D anatomical structures are better defined. Also, due to the practical reasons like lower memory requirements, reduced computational complexity, ease of implementation, 2D segmentation methods are often applied sequentially to the slices of 3D images [57]. These observations motivate us to work on a 2D segmentation method and apply it to the 3D images slice by slice. Our segmentation method is based on the active contour model for its popularity and potency in image segmentation. It can effectively find the object's boundary, and it has constraints on the shape of the contour in order to regularize the shape of the contour. Particularly, the parametric active contour model has strict topology control on the contour, which is extremely suitable for detecting the object with single topology, like the heart ventricle.

Instead of setting up the experiment for each slice separately, we want to make the process as automatic as possible. Since we want to segment the 3D image slice by slice, it is straightforward to use the segmentation result of the previous slice as the initialization for the next slice and then evolve the initialized contour in the new slice. The process should ideally be fully automatic. However, the segmentation method cannot be expected to be perfect. Neither do we want to check the result of every slice for this would be much manual work. Therefore, an automatic validation step is necessary. The validation step should have the ability to judge whether the segmentation result of each slice is accurate.

The first objective of the thesis is to design a robust 2D segmentation method based on the active contour model. Secondly, we apply this 2D segmentation method to 2D slice sequence in 3D image. During the segmentation process, a validation step is applied in order to automatically judge whether the segmentation on each slice is accurate.

### **1.3 Thesis Contribution**

Two main contributions are made in this thesis.

1. First we propose a novel parametric active contour model that is robust to noise, cluttered image background, and weak boundary. The model is called as VFCCV. Both edge and regional information are considered in the evolution function of VFCCV. The edge information is taken into account within the vector field convolution (VFC) model, and the regional information within the Chan-Vese model. Although several active contour models considering both the edge and regional information have been studied, most of them are in the geometric framework and combining the VFC and Chan-Vese types of forces has never been studied.

The VFC model only considers edge information, and thus it is sensitive to noise and easily trapped in cluttered regions. The Chan-Vese model only considers regional

information and thus is easy to pass the weak boundary. However, our VFCCV model considers both edge and regional information and thus it overcomes the weaknesses of both VFC and Chan-Vese. The VFCCV model also has a large capture range and the ability to detect concavity. We use experimental results to show the superiority of this model.

2. We perform 3D heart image segmentation by dividing a 3D image into 2D slices. As has been done in the literature, the segmentation result of the previous slice is used as the initialization of the next slice. The contour on the new slice is evolved using VFCCV, and thus the segmentation propagates along the 2D slice sequence. We introduce a validation step in the propagation process to automatically judge whether the contour on each slice is accurate. We apply PCA-based snake validation method in our validation method. We extend the existing method by performing the validation locally instead of globally to detect local errors. Also, we propose to build the PCA training data in a location-specific manner in order to avoid the influence of dissimilar image features. By embedding the validation step in the segmentation system, we make the segmentation process as automatic as possible while keeping a high accuracy.

Experiments are conducted on heart phantom ultrasound images and human heart ultrasound images. The validation step can effectively detect the inaccurate contours. Also, the segmentation system is significantly faster than manual validation. The heart phantom dataset can provide the ground truth regarding the volume measurement with which we can compare. Experimental results show that the final 3D heart volumes have high accuracy. The segmentation system also has high reproducibility. A user interface is built for the convenient operation in practice.

## 1.4 Organization

In this thesis, we will first introduce the background and related work in Chapter 2. Then, we will introduce the proposed active contour model called VFCCV in Chapter 3. In Chapter 4, we will present the active contour automatic validation method. All the experimental results and analysis are given in Chapter 5. The conclusion and future work are given in Chapter 6.

## Chapter 2

# Background and Related Work

In this chapter, we will present the background and related work of this thesis. We will mainly focus on introducing different kinds of active contour models. A snake-like interactive segmentation technique is also introduced.

### 2.1 Introduction

Low-level image segmentation methods like thresholding, region growing, etc., are easy to generate wrong segmentation results, especially for the relatively complex medical images. As a result, those model-free methods require a large amount of manual intervention in order to get the correct results [46]. Model-based methods like deformable models [27], active shape models [69], active appearance model [49], Markov random field models [47], artificial neural networks [24], atlas-guided approaches [1], etc., have better performance compared to those low-level techniques.

The deformable model is a vigorously studied and promising method to computer-aided medical image segmentation, and its widely recognized potency stems from its ability to segment and trace the anatomic structures [46]. In this thesis, we will mainly focus on the echocardiography segmentation using deformable models. The deformable model can effectively find the object's boundary, and it has constraints on the shape of the contour in order to regularize the shape of the contour. Both 2D and 3D deformable models have been studied in the literatures. We will only focus on the 2D models in this thesis.

The term “deformable models” stems primarily from the use of elasticity theory at the physical level [46]. In physics, the deformable model is regarded as an elastic body that can respond naturally to the internal constraints and external forces. In computer vision, the deformable model is also called the active contour model, that is, a curve that can move according to constraints and forces. In the following parts of this thesis, we will use the term “active contour model”.

In this chapter, we will introduce the mathematical foundations of different kinds of the

active contour models. The active contour models can be categorized into two classes, the parametric active contour model and the geometric active contour model [76]. In this chapter, we will mainly introduce four kinds of parametric active contour models and three kinds of geometric active contour models, followed by the introduction of a snake-like technique called intelligent scissors. The summary is given in the last section.

## 2.2 Parametric Active Contour Models

The model is named as parametric active contour model because its evolution is associated with the contour's parameters. In this section, we will introduce four popular parametric active contour models. The first one is the traditional snake model [27], the most famous parametric active contour model. The second one is the balloon force model that enlarges the capture range of the traditional snake model. The third one is the gradient vector flow (GVF) model [77], which solves the main problems of the traditional snake model, followed by its variation, the generalized gradient vector flow (GGVF) model.

### 2.2.1 Snake Model

The snake model was first proposed by Kass *et al.* in 1988 [27]. A contour is first initialized within the image domain. The contour can evolve in the image domain according to the energy function associated with it. The process of evolution is the process of minimizing the energy function. When the active contour locates on the image boundary, the energy function reaches its minimum.

The active contour is represented parametrically by  $\mathbf{v}(s) = (x(s), y(s))$ ,  $s \in [0, 1]$ . Its energy functional is the integration along the contour, including both internal energy and external energy, which can be written as

$$\begin{aligned} E_{snake} &= \int_0^1 E_{snake}(\mathbf{v}(s)) ds \\ &= \int_0^1 [E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s))] ds \end{aligned} \quad (2.1)$$

where  $E_{int}$  is the internal energy, and  $E_{ext}$  is the external energy. The internal energy controls the first-order and second-order continuity of the contour, which can be defined as

$$E_{int} = \frac{1}{2}(\alpha(s) |\mathbf{v}'(s)|^2 + \beta(s) |\mathbf{v}''(s)|^2) \quad (2.2)$$

The first-order derivative controls the tension of the contour. Minimizing the first-order derivative discourages ripples and loops, and the contour tends to be circular. The second-order derivative controls the bending of the contour. Minimizing the second-order derivative discourages the bending rigidity, and the contour tends to be smoother.  $\alpha(s)$  and  $\beta(s)$  are

weights controlling the model's tension and rigidity respectively. They are often chosen as contents in practice.

The external energy takes the advantage of the image information and gets smaller values at object boundary or other regions of interest. For a binary image  $I(x, y)$ , the external energy can be defined as

$$E_{ext}^1 = w_{line}[I(x, y)] \quad (2.3)$$

$$E_{ext}^2 = w_{line}[G_\sigma(x, y) * I(x, y)] \quad (2.4)$$

where  $w_{line}$  is the parameter. For black lines on white background,  $w_{line}$  should be positive. For white lines on black background,  $w_{line}$  should be negative.  $G_\sigma$  is the two dimensional Gaussian function with standard deviation  $\sigma$ .  $*$  is the two-dimensional convolution operator. To be consistent with common representation of pixels on images, here we use  $(x, y)$  instead of  $v(s)$  defined before to represent the point on the parametric contour.

For a greyscale image, the external energy can be defined as

$$E_{ext}^3 = -w_{edge} |\nabla[I(x, y)]|^2 \quad (2.5)$$

$$E_{ext}^4 = -w_{edge} |\nabla[G_\sigma(x, y) * I(x, y)]|^2 \quad (2.6)$$

where  $w_{edge}$  is a positive parameter and by taking the gradient of the image, the external energy takes smaller values at object boundaries. Note that enlarging  $\sigma$  broadens the capture range of the edge, however, at the same time blurs the edge and the segmentation may be shifted from the real boundary. A large capture range means the range of influence of the force is large, and thus even though the initialization is far away from the boundary, the method can still detect the object's boundary. Fig. 2.1(c) shows the edge map,  $-E_{ext}^4$ , of the original grey-level image in Fig. 2.1(a). The edge map has larger value near the object boundary, that is, the energy is small near the object boundary.

Kass *et al.* [27] also proposed the constraint energy, which includes the spring energy and volcano energy, and the termination energy. These energies can be added to the external energy. We will not include the details here.

By minimizing the total energy defined in Eq. 2.1, we can get the evolved snake. It is known as a variational problem [12] and the curve minimizing  $E_{snake}$  should satisfy the Euler-Lagrange equation shown as follows [7, 27, 76]

$$\frac{\partial}{\partial s} \left( \alpha \frac{\partial v}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 v}{\partial s^2} \right) - \nabla E_{ext} = 0 \quad (2.7)$$

The detailed deduction of the internal energy part can be found in Appendix A. By regarding  $v$  as a function of time  $t$  as well as  $s$ , the active contour is dynamic and thus we can solve Eq. 2.7 by iteration. The partial derivative of  $v$  with respect to  $t$  is set to be equal

to the left-hand side of Eq. 2.7 as follows

$$\gamma \frac{\partial \mathbf{v}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) - \nabla E_{ext} \quad (2.8)$$

The parameter  $\gamma$  is introduced to make the units on the left side consistent with the right side [76].  $\gamma$  is often set as 1 and thus been ignored in many applications [77, 36]. The contour is updated iteratively until it converges. When the contour stops moving,  $\partial \mathbf{v} / \partial t$  reaches 0 and at the same time, the right-hand side of Eq. 2.8 also reaches 0, which satisfies the Euler-Lagrange equation in Eq. 2.7.

Fig. 2.1(b) shows a typical evolution process of a snake model. The initialization is a circle outside the elliptical object and contour can finally locates on the boundary.

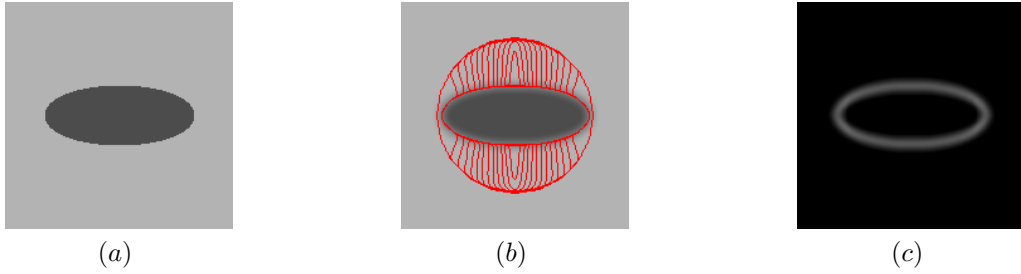


Figure 2.1: (a) The original image (b) Typical snake evolution (c)  $-E_{ext}^4$  of the original image.  $\sigma = 5, w_{edge} = 1$

## 2.2.2 Balloon Force Model

In the snake model introduced in the previous section, since the edge information is taken locally, and the Euler-Lagrange equation aims at finding the local minimum, the initialized contour needs to be close to the real boundary, or it will evolve to the wrong location. Also, the snake has difficulties in converging to concave regions. This is because in concave regions, the directions of the external force inside the concave region are in the opposite with each other, and thus the influences of the edge cancel out with each other. This effect can be illustrated in Fig. 2.2. The balloon force model [7] is a popular model to solve these problems.

Unlike the above snake model, which is in energy minimization formulation, the balloon force model is in dynamic force formulation. Before we introduce the balloon force model, we will first introduce the dynamic force formulation of the parametric active contour models. The dynamic force formulation is derived from the Newton's second law. The dynamics of a contour should satisfy the following equation [76]

$$\mu \frac{\partial^2 \mathbf{v}}{\partial t^2} = F_{damp}(\mathbf{v}) + F_{int}(\mathbf{v}) + F_{ext}(\mathbf{v}) \quad (2.9)$$



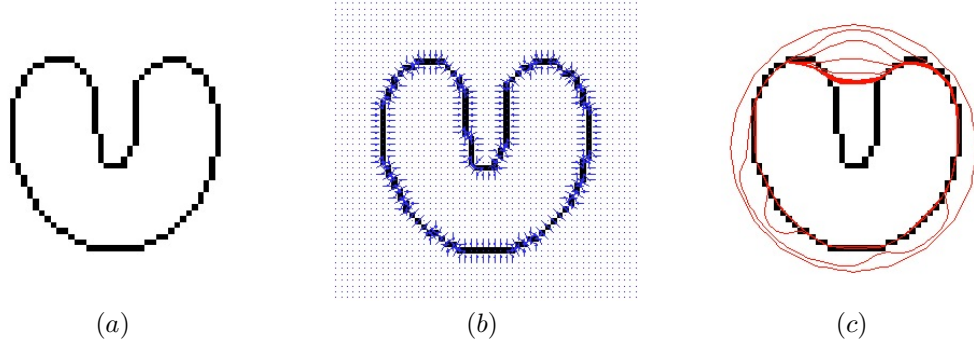


Figure 2.2: (a) The U-shape image (b) The external force field of the snake model (c) The snake cannot evolve into the concave region

where  $\mu$  is a parameter that can be regarded as the mass in Newton's second law.  $F_{damp}$ , the damping force, defined as  $-\gamma \partial \mathbf{v} / \partial t$ , where  $\gamma$  is the damping coefficient.  $F_{int}$  is the internal force and  $F_{ext}$  is the external force. By setting  $\mu$  as 0, we can get the following equation

$$\gamma \frac{\partial \mathbf{v}}{\partial t} = F_{int}(\mathbf{v}) + F_{ext}(\mathbf{v}) \quad (2.10)$$

In fact, Eq. 2.10 is the same with Eq. 2.8, the evolution function derived in energy minimization formulation, by setting the internal force as follows

$$F_{int} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) \quad (2.11)$$

and the external force

$$F_{ext} = -\nabla E_{ext} \quad (2.12)$$

So in fact the right side of the evolution function in Eq. 2.8 can be regarded as two forces. The contour will stop moving when the internal force and the external force reach the balance status.

Note that the external force in the energy minimization formulation in Eq. 2.8 has to be written as the negative gradient of a potential energy function. However, the external force in dynamic force formulation in Eq. 2.10 can be more general types of forces, and it can be either potential force or non-potential force. The balloon force model is just derived from the dynamic force formulation taking this convenient.

As mentioned above, the balloon force model is proposed to solve the problems of the snake model. This is done by changing the external force of snake. The external force of the balloon force model is the combination of the original external force and the pressure force. The pressure force which is defined as

$$F_p(\mathbf{v}) = w_p \mathbf{N}(\mathbf{v}) \quad (2.13)$$

where  $\mathbf{N}(\mathbf{v})$  is the inward normal vector at point  $\mathbf{v}$ .  $w_p$  is the weight, and the sign of it decides whether to enlarge or shrink the contour. With the pressure force, the contour can

evolve towards the object even when the initialization is far away from the object. Also, the pressure force can drag the contour into concave region. However, although the sign of  $w_p$  is easy to choose, the magnitude of it is hard to decide. It should be smaller than the Gaussian potential force in order to stop at the true boundary, but large enough to pass through noise or spurious edges [76]. Either too small or too large pressure force will result in undesirable results [37, 82].

### 2.2.3 Gradient Vector Flow

Except for the balloon force model, many other models have also been proposed to solve the two main problems of snake model, for example, the multi-scale Gaussian potential force model [67], the distance potential force model [8], the dynamic distance force model [15, 41], the interactive force model [27], the control points model [14], etc. However, just like the balloon force model, they either generate new problems or only solve one of the two main problems [76]. The active contour model that will be introduced in this section solves these two problems and has good segmentation performance, which is called the gradient vector flow (GVF) model [77].

The GVF model is also formed in dynamic force formulation. It replaces the original external force field in Eq. 2.10 by the GVF force field. In the regions near the object boundary, the GVF field points toward the boundary as the original external force field does. However, the GVF field diffuses from the boundary regions to homogeneous regions in the image domain and finally gets a smooth force field. This is done by defining an energy functional of the GVF field and minimizing this functional.

The edge map  $e(x, y)$  is first been derived from the image  $I(x, y)$ . The edge map has larger values near the object boundary. It can simply be the negative of the external energy

$$e(x, y) = -E_{ext} \quad (2.14)$$

The external energy is defined from Eq. 2.3 to Eq. 2.6. The original external force field is just  $-\nabla E_{ext}$  which is taken locally and thus only has small capture range. There is no force field at all in homogeneous regions. The GVF field is defined to be the vector field  $\mathbf{f}_{gvf}(x, y) = (u_{gvf}(x, y), v_{gvf}(x, y))$  that minimizes the energy functional

$$E_{gvf} = \int \int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla e|^2 |\mathbf{f}_{gvf} - \nabla e|^2 dx dy \quad (2.15)$$

Different from the  $E_{snake}$  introduced in the previous section, the  $E_{gvf}$  here is not the energy of the GVF snake, but is the energy of the force field whose minimum is the GVF force field.  $\mu$  is a parameter controlling the tradeoff between the first term and the second term.  $e$  is the edge map of the image. When  $|\nabla e|$  is large, that is, near the object boundary, the integral is dominated by the second term. Therefore, the GVF field is approximately equal to  $\nabla e$ .

When  $|\nabla e|$  is small, the integral is dominated by the first term. By minimizing the partial differential in both  $x$  and  $y$  directions, the GVF field tends to be smoothly diffused from the boundaries to homogeneous regions.

In this way, the GVF field propagates from the boundary regions to the homogeneous regions and thus enlarge the capture range. Also, the force field can drag the contour to concave regions. The GVF field is shown in Fig. 2.3(b). The evolved snake is shown in Fig. 2.3(c) from which we can see that the GVF snake can detect the concave region.

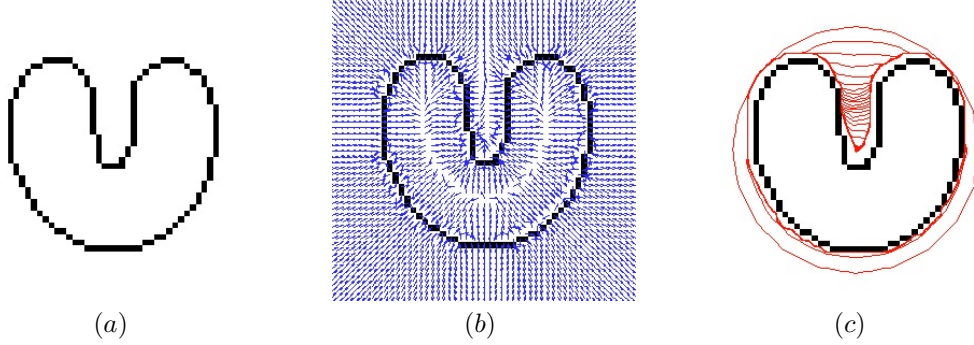


Figure 2.3: (a) The U-shape image (b) The force field of the GVF model (c) The GVF snake can evolve into the concave region

The GVF field can be solved by Euler equation using the calculus of variations [12, 77]

$$\mu \nabla^2 u - (u - e_x)(e_x^2 + e_y^2) = 0 \quad (2.16)$$

$$\mu \nabla^2 v - (v - e_y)(e_x^2 + e_y^2) = 0 \quad (2.17)$$

Here  $\nabla^2$  is the Laplacian operator. Similar to treating the active contour as a function of time  $t$  discussed before, the GVF field can also be solved in this way through iteration

$$\frac{\partial u}{\partial t} = \mu \nabla^2 u - (u - e_x)(e_x^2 + e_y^2) \quad (2.18)$$

$$\frac{\partial v}{\partial t} = \mu \nabla^2 v - (v - e_y)(e_x^2 + e_y^2) \quad (2.19)$$

By replacing the original external force field and set the  $\gamma$  in Eq. 2.8 as 1, the final evolution function of GVF model is

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) + \mathbf{f}_{gvf} \quad (2.20)$$

In the next section, we will introduce the improved version of the GVF model which is called the generalized gradient vector flow(GGVF) model.

## 2.2.4 Generalized Gradient Vector Flow Model

We can rewrite Eq. 2.18 and Eq. 2.19, the evolution functional of the GVF model, in a simple form shown as follows

$$\frac{\partial \mathbf{f}_{gvf}}{\partial t} = \mu \nabla^2 \mathbf{f}_{gvf} - |\nabla e|^2 (\mathbf{f}_{gvf} - \nabla e) \quad (2.21)$$

where  $\mathbf{f}_{gvf}$  is the GVF force field. The first term can be regarded as the smoothing term. By minimizing the derivatives of the force field, the field is smoothed and thus the information near the boundary is propagated to far away homogeneous regions. The second term can be regarded as the data term. It forces the field close to  $\nabla e$  especially when near the boundary. Although the GVF field can detect concave objects, it is still hard for the GVF snake to locate on thin and long indentations. It is because the constant weight  $\mu$  smoothes the field everywhere including the regions inside the thin and long indentations. Therefore, the GVF model smoothes the nearby forces with opposite directions and thus the force is not strong enough to drag the contour inside the thin and long indentation.

To solve this problem, the generalized gradient vector flow (GGVF) [78] is proposed. The GGVF model replaces the two weights of the smoothing and data terms by weighting functions and generalizes the weighting functions. Then, we can get the generalized form of Eq. 2.21 as follows

$$\frac{\partial \mathbf{f}_{gvf}}{\partial t} = g(\cdot) \nabla^2 \mathbf{f}_{gvf} - h(\cdot) (\mathbf{f}_{gvf} - \nabla e) \quad (2.22)$$

Here,  $g(\cdot)$  and  $h(\cdot)$  are the two weighting functions. In the original GVF model, we choose  $g(\cdot) = \mu$  and  $h(\cdot) = |\nabla e|^2$ . However, in regions near the boundary, the force field should be close to  $\nabla e$  and has small smoothing effect, which leads to a small  $g(\cdot)$  and large  $h(\cdot)$ . In regions far away from boundary, we need more smoothing effect in order for the edge information to be propagated to homogenous regions, which leads to a large  $g(\cdot)$  and small  $h(\cdot)$ . Based on the above analysis,  $g(\cdot)$  and  $h(\cdot)$  should be spatially varying functions because they depend on the magnitude of the gradient of the edge map which is spatially varying. Also,  $g(\cdot)$  should be designed as monotonically non-increasing of  $|\nabla e|$  and  $f(\cdot)$  is monotonically non-decreasing of  $|\nabla e|$ . In the GGVF model, the original weight functions are replaced by the following new weight functions

$$\begin{aligned} g(|\nabla e|) &= \exp\left(-\frac{|\nabla e|}{K}\right) \\ h(|\nabla e|) &= 1 - g(|\nabla e|) \end{aligned} \quad (2.23)$$

where  $K$  is the tuning parameter controlling the tradeoff between the smoothing term and the data term. We can see that  $g(\cdot)$  gets smaller as close to the edges and  $h(\cdot)$  gets larger as  $g(\cdot)$  getting smaller.

Above all, the GGVF model solves the problem of detecting the thin and long indentations. However, although the GVF and GGVF have the flexibility for initialization and can

force the contour to concave regions or even thin and long indentations, they still have some disadvantages. For example, the sensitivity to noise, the high computational cost, the ambiguous relationship between parameters and capture range [36]. The vector field convolution (VFC) model is further proposed to solve the problems of GVF and GGVF while keeping their advantages. We will introduce the VFC model in detail in the next chapter.

### 2.2.5 Other Parametric Active Contour Models

Besides the above parametric active contour models, many other kinds of parametric active contour models are also been studied. For example, the probabilistic deformable model [65], models incorporating prior information [40, 81], models based on Markov Random Field [47], GICOV based snake solved by dynamic programming [3], etc. We will not introduce these models in this thesis.

## 2.3 Geometric Active Contours

In the previous section, we have introduced the first kind the active contour model, the parametric active contour model. In this section, we will introduce the second kind of active contour model, the geometric active contour model. Different from the parametric active contour model, the evolution of the geometric active contour is independent from the parameters of the contour. The contour is evolved using geometric measures.

The geometric active contour model is also called the level set model. The level set method was introduced by Osher and Sethian [55] in 1987. In image processing area, the level set model was introduced independently by Caselles *et al.* [4] and Malladi *et al.* [43] based on the curve evolution theory [62, 30, 31, 2] and the level set method [55, 63]. The level set of a 2D scalar function is the set of the points that have the identical function value [76]. In the geometric active contour model, the contour is represented as the zero level set of a higher dimensional function, called the level set function (LSF). This leads to the advantage that the geometric active contour over the parametric active contour, that is, it can handle topological changes of the active contour automatically. For a parametric active contour, due to the limits of the first-order and second-order continuity, the topology of the contour cannot be changed.

In this section, we will first introduce the early geometric active contours that based on the curve evolution theory. Then, we will introduce the geodesic active contour model, followed by a famous geometric active model, called the Chan-Vese model.

### 2.3.1 Curve Evolution-based Model

The first geometric active contour is based on the curve evolution theory. Similar as before, we first denote a curve as a function of both the parameter  $s$  and the time  $t$ , that is,  $v(s, t)$ . The evolution of the curve can be expressed as

$$\frac{\partial v}{\partial t} = FN \quad (2.24)$$

where  $F$  is the speed function controlling the motion of the curve and  $N$  is the inward normal vector of the curve. The intuition behind this fact is that the tangent deformation affects only the curve's parameterization, not its geometry and shape [76].

One formulation of  $F$  is to design it as a function of the curvature, for example

$$\frac{\partial v}{\partial t} = \alpha \kappa N \quad (2.25)$$

where  $\alpha$  is a positive constant, and  $\kappa$  is the curvature of the curve. The above function has large speed where the curvature of the curve is large so that it smoothes the curve and eventually shrinks it to a small circle [22].

Based on the above curve evolution theory, we further embed the dynamic curve as the zero level set of the 2D scalar level set function (LSF),  $\phi(x, y, t)$ . Here  $x$  and  $y$  are the coordinate on the image grid, and  $t$  indicates the time index. In this way, the evolution of the active contour is transformed to the evolution of the LSF, and thus the topological changes of the active contour can be handled automatically. Even if the 2D curves split or merge, the LSF keeps a valid function.

Based on Eq. 2.24 and embedding the curve as the zero level set of  $\phi(x, y, t)$ , we have

$$\phi[v(s, t), t] = 0 \quad (2.26)$$

Differentiating the above equation with respect to  $t$ , we can obtain

$$\frac{\partial \phi}{\partial t} + \nabla \phi \frac{\partial v}{\partial t} = 0 \quad (2.27)$$

where  $\nabla$  is the gradient operator [72]. Assume that the LSF has negative values inside the zero level set, and positive values outside. The inward normal vector can be written as [71]

$$N = -\frac{\nabla \phi}{|\nabla \phi|} \quad (2.28)$$

Based on the Eq. 2.24 and Eq. 2.28, Eq. 2.27 can be expressed as

$$\frac{\partial \phi}{\partial t} = F|\nabla \phi| \quad (2.29)$$

which is the basic form of the level set evolution equation.

Fig. 2.4 and Fig. 2.5 show the evolution of the zero level set and the corresponding LSF. The contour can split to detect multiple objects. For a better view, the level set function

is displayed upside down and thus has positive values inside and negative values outside. The initialized LSF can also be the signed distance function towards the zero level set. Irregularities may appear during the evolution which would result in numerical errors and even destroy the stability of the LSF [39]. Therefore reinitialization is often required to periodically replace the evolved LSF as the signed distance function. We will not include the reinitialization details in this thesis.

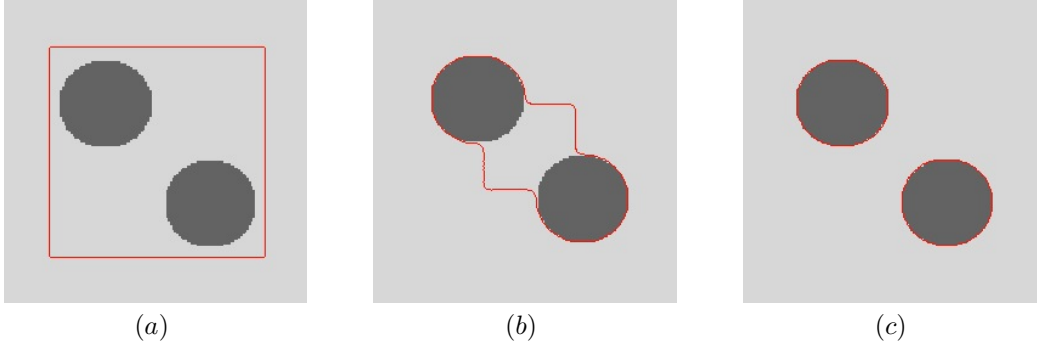


Figure 2.4: The evolution of the zero level set. The contour can split and detect the two objects.

The basic idea of the geometric active contour is to design the speed function  $F$  using the image data and thus the contour can stop at the object boundary. Inspired by the speed function in Eq. 2.25, the speed function can be designed based on both curvature and image information which is proposed by Caselles *et al.* [4] and Malladi *et al.* [43]. The speed function can be expressed as

$$\frac{\partial \phi}{\partial t} = g(\kappa + V_0)|\nabla \phi| \quad (2.30)$$

where

$$g \triangleq \frac{1}{1 + |\nabla(G_\sigma * I)|} \quad (2.31)$$

$$\kappa = \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right) \quad (2.32)$$

Here  $g$  incorporates the image information, and decreases the speed near the object boundary.  $G_\sigma *$  is the 2D Gaussian convolution as defined before.  $\kappa$  is the curvature for all the level sets,  $\operatorname{div}(\cdot)$  is the divergence operator [70], and  $\nabla$  is the gradient operator as usual.  $V_0$  is a constant parameter. Based on the assumption that the level set function is negative inside the zero level set and positive outside, positive  $V_0$  increase the value of LSF and thus shrink the zero level set, and negative  $V_0$  enlarge the zero level set. The constant  $V_0$  provides an extra speed that can accelerate the evolution. Also, it can prevent the curve stopping when the curvature is zero but the curve has not arrived at the boundary. For example, it can detect concave objects.

There are many other ways to design the speed function  $F$  in Eq. 2.29 [5, 80, 29, 64]. We will not include the details in this thesis.

### 2.3.2 Geodesic Active Contours

The geodesic active contour was proposed by Caselles *et al.* in 1997 [5]. It is also a geometric active contour model that can naturally handle the topological changes and segment multiple objects simultaneously. It has the advantages of fewer parameters compared with many other geometric models, and the ability to detect weak boundary.

The geodesic active contour model can be regarded as the connection between the curve evolution based level set model and the classical energy minimization snake model. Mathematically, it is derived from the classical snake model, but the problem is transformed to computing the geodesics, or the minimal distance curve between two points on a chosen Riemannian space. So in this section, we will first introduce the geodesic active contour model based on the curve evolution based model introduced in the previous section. Then, we will introduce its mathematical formulation derived from the energy minimizing snake model.

The geodesic active contour model can be regarded as the variation of the curve evolution based model proposed in Eq. 2.30. The evolution function of the geodesic active contour is represented as follows

$$\frac{\partial \phi}{\partial t} = g(\kappa + V_0)|\nabla \phi| + \nabla \phi \cdot \nabla g \quad (2.33)$$

Here the notations are the same as defined in Eq. 2.30. In the original evolution function proposed in Eq. 2.30, the evolution stops when  $g$  is very small, which just holds for strong edges. So the original model is easy to pass weak boundary. However, with the term  $\nabla \phi \cdot \nabla g$ , which can make difference even near weak boundary, the model can detect object with a weak boundary. But it also makes the model sensitive to the noise and cluttered background.

In fact, the geodesic active contour can be derived from the snake model introduced before, whose energy function can be rewritten as

$$E_{gac} = \alpha \int_0^1 |\mathbf{v}'(s)|^2 ds + \beta \int_0^1 |\mathbf{v}''(s)|^2 ds - w \int_0^1 |\nabla I(\mathbf{v}(s))| ds \quad (2.34)$$

where  $\alpha, \beta, w$  are positive parameters. It has been shown that we can get a smooth contour without the second term [4]. So  $\beta$  is set to 0 and we get

$$E_{gac} = \alpha \int_0^1 |\mathbf{v}'(s)|^2 ds - w \int_0^1 |\nabla I(\mathbf{v}(s))| ds \quad (2.35)$$

Define an edge detector function,  $g : [0, +\infty) \rightarrow R^+$  be a monotonically decreasing function,  $g(r) \rightarrow 0$  as  $r \rightarrow \infty$ , and  $g(0) = 1$ . For example,  $g(|\nabla I(\mathbf{v}(s))|) = \frac{1}{1+|\nabla I|}$  or



$g(|\nabla I(\mathbf{v}(s))|) = \frac{1}{1+|\nabla I|^2}$ , etc. Then  $|\nabla I(\mathbf{v}(s))|$  can be replaced by  $g(|\nabla I(\mathbf{v}(s))|)^2$  without influencing the result. So we have

$$E_{gac} = \alpha \int_0^1 |\mathbf{v}'(s)|^2 ds + w \int_0^1 g(|\nabla I(\mathbf{v}(s))|)^2 ds \quad (2.36)$$

Using theorems on Riemannian calculus, the problem of minimizing Eq. 2.36 is transformed to the problem of geodesic computation in a Riemannian space according to a new metric [5]. We will not include the mathematical details here. The new minimization problem is shown as follows

$$\text{Min } E_{gac} = \text{Min} \int_0^1 g(|\nabla I(\mathbf{v}(s))|) |\mathbf{v}'(s)| ds \quad (2.37)$$

In the above equation,  $s$ , the contour's parameter, is the variable. Since the Euclidean length of the curve can be represented by

$$L = \int_0^1 |\mathbf{v}'(s)| ds = \int_0^{L(\mathbf{v})} dq \quad (2.38)$$

where  $dq$  is the Euclidean arc-length. So Eq. 2.37 can be written as

$$\text{Min} \int_0^{L(\mathbf{v})} g(|\nabla I(\mathbf{v}(s))|) dq \quad (2.39)$$

Based on the Euler-Lagrange equation, the above minimization problem can be solved by the steepest descent method [5]

$$\frac{\partial \mathbf{v}}{\partial t} = g\kappa \mathbf{N} - (\nabla g \cdot \mathbf{N}) \mathbf{N} \quad (2.40)$$

where  $\kappa$  is the curvature, and  $\mathbf{N}$  is the unit inward normal vector of the curve. Representing the curve as the zero level set of the level set function, we have the following level set evolution function

$$\frac{\partial \phi}{\partial t} = g\kappa |\nabla \phi| + \nabla \phi \cdot \nabla g \quad (2.41)$$

We can add a constant  $V_0$  following  $\kappa$  in case we want to speed up the evolution or detect non-convex objects. Then we can get the same formulation as proposed in Eq. 2.33.

### 2.3.3 Chan-Vese Model

All the active contour models mentioned above are based on the edge information. However, the Chan-Vese model [6] is a region-based model implemented in geometric active contour framework. Instead of directly designing the evolution function in Eq. 2.29, the Chan-Vese model is formed in an energy minimization way.

Denote the active contour as  $\mathbf{v}$ , the original image as  $I$ , the region inside the contour in the image domain as  $inside(\mathbf{v})$ , and the region outside as  $outside(\mathbf{v})$ . A region-based energy of the active contour is first defined as

$$F_1(\mathbf{v}) + F_2(\mathbf{v}) = \int_{inside(\mathbf{v})} |I(x, y) - c_1|^2 dx dy + \int_{outside(\mathbf{v})} |I(x, y) - c_2|^2 dx dy \quad (2.42)$$

where  $c_1$  and  $c_2$  are two constants.  $c_1$  is the average greyscale inside  $v$ , and  $c_2$  is the average greyscale outside  $v$ . We use  $F$  instead of  $E$  representing the energy to indicate they are the fitting energy, just like how other papers do. The energy gets its minimum when the contour locates on the object boundary. This can be illustrated by the toy example in Fig. 2.6. Assume the image  $I$  is formed by two regions, and each region has approximately constant greyscale. The dark grey region is the object, and the light grey region is the background. When the contour is outside the object,  $F_1(v) > 0$  and  $F_2(v) \approx 0$ . When the contour is inside the object,  $F_1(v) \approx 0$  and  $F_2(v) > 0$ . When the contour goes across the object,  $F_1(v) > 0$  and  $F_2(v) > 0$ . Only when the contour locates on the object boundary, we get  $F_1(v) \approx 0$  and  $F_2(v) \approx 0$  reaching the minimum of the region-based energy.

The Chan-Vese model also adds two regularization terms, the length term and the area term, in the energy function. The final energy function of the active contour is shown as follows

$$E_{Chan-Vese} = \mu Length(v) + \nu Area(inside(v)) + \lambda_1 \int_{inside(v)} |I(x, y) - c_1|^2 dx dy + \lambda_2 \int_{outside(v)} |I(x, y) - c_2|^2 dx dy \quad (2.43)$$

where  $\mu, \nu, \lambda_1, \lambda_2$  are positive parameters controlling the strength of each term. The contour tends to be smoothed by minimizing the length of the contour. The area term is introduced to accelerate the evolution of the zero level set, which is necessary when the contour is initialized far away from the object [39].

Then, the energy of the 2D contour can be transformed to the energy of the zero level set in the LSF. Before we introduce the transformation, we first introduce the Heaviside function  $H$ , and the Dirac function  $\delta$  which are shown as follows

$$H(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2.44)$$

$$\delta(x) = \frac{d}{dx} H(x) \quad (2.45)$$

We can then transform Eq. 2.43 into the level set formulation by replacing the contour  $v$  into the LSF  $\phi$ . Denote the image domain as  $\Omega$ . Assume that the LSF has positive values inside the zero level set and negative outside. Then, we have

$$\begin{aligned} Length(\phi = 0) &= \int_{\Omega} |\nabla H(\phi(x, y))| dx dy \\ &= \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy \end{aligned} \quad (2.46)$$

$$Area(\phi \geq 0) = \int_{\Omega} H(\phi(x, y)) dx dy \quad (2.47)$$

$$\begin{aligned} F_1(\phi \geq 0) &= \int_{\phi \geq 0} |I(x, y) - c_1|^2 dx dy \\ &= \int_{\phi} |I(x, y) - c_1|^2 H(\phi(x, y)) dx dy \end{aligned} \quad (2.48)$$

$$\begin{aligned}
F_2(\phi < 0) &= \int_{\phi < 0} |I(x, y) - c_2|^2 dx dy \\
&= \int_{\phi} |I(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy
\end{aligned} \tag{2.49}$$

Then, the energy in Eq. 2.43 can be expressed as

$$\begin{aligned}
E_{Chan-Vese} &= \mu \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\
&\quad + \lambda_1 \int_{\Omega} |I(x, y) - c_1|^2 H(\phi(x, y)) dx dy + \lambda_2 \int_{\Omega} |I(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy
\end{aligned} \tag{2.50}$$

First keeping  $c_1$  and  $c_2$  fixed, based on the Euler-Lagrange equation, the energy  $F$  can be minimized through the following evolution function

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ \mu \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2 \right] \tag{2.51}$$

where  $\operatorname{div}(\cdot)$  is the divergence operator.  $c_1$  and  $c_2$  are updated every iteration as the zero level set evolving.

### 2.3.4 Other Geometric Active Contours

As mentioned before, reinitialization is often required to avoid irregularities during the evolution process. However, the reinitialization step will also generate problems in practice [39]. A distance regularized level set model is proposed in [39] which incorporates the shape regularization term in the energy function, and thus removes the need of reinitialization. Many other geometric active contour models have also been proposed. For example, a region-based model is proposed in [38]. The region size is scalable in order to deal with intensity inhomogeneities. We will not include the details of the other models in this thesis.

## 2.4 Intelligent Scissors

We have introduced different kinds of active contour models in the previous sections including both parametric and geometric models. In this section, we will introduce a snake-like interactive segmentation tool called intelligent scissors [50].

Softwares like Gimp and Photoshop have the lasso tool which is used for delineate the object boundary. It is an interactive segmentation method. The contour can snap to the boundary while the user clicks several points along the boundary to give hints to the lasso tool. A typical interface of the lasso tool is shown in Fig. 2.7. The fully-automatic or semi-automatic segmentation methods might be inaccurate, and usually they cannot be applied to all image contents. With the user guidance, the intelligent scissors can segment object from arbitrarily complex background. The reason we will introduce the intelligent scissors

is that it serves a perfect tool to segment the ultrasonic heart images, and we will use it in our future experiments.

Different from the active contour models whose results are obtained from minimizing the energy of all the points on the contour, the results of intelligent scissors are formulated stage by stage by finding the optimal path from the current mouse-clicked point to the previous mouse-clicked point. Each pixel in the image can be regarded as a node, and the pixel and its eight neighbours form the edges between nodes. The path is defined to be a series of connected edges, and the optimal path is the one with the minimum cumulative cost from the starting point to the end point. Here, the cumulative sum is the sum of the costs of edges along the path. To find the optimal path between the two points, the problem is formulated as a graph search problem solved by dynamic programming, for example, the Dijkstra algorithm [17].

In order to make the path along the object boundary has the minimum cost, pixels with strong edge information should have low costs. The cost for the directed edge from  $p$  to its neighbour  $q$  is defined as

$$l(\mathbf{p}, \mathbf{q}) = w_Z f_Z(\mathbf{q}) + w_G f_G(\mathbf{q}) + w_D f_D(\mathbf{p}, \mathbf{q}) \quad (2.52)$$

where each  $w$  is the corresponding weight to each term.  $f_Z$  is the laplacian zero-crossing cost. The convolution result of an image and a laplacian kernel approximates the second order partial derivative of the image [50]. The zero crossing of the convoluted image corresponds to pixels with maximum or minimum gradient. So the zero crossing points should have small cost, and thus  $f_Z$  is defined as

$$f_Z(\mathbf{q}) = \begin{cases} 0, & I_L(\mathbf{q}) = 0 \\ 1, & I_L(\mathbf{q}) \neq 0 \end{cases} \quad (2.53)$$

where  $I_L$  is the laplacian convoluted image. Note that  $I_L(\mathbf{q}) = 0$  does not mean the point with zero value exactly since it rarely appears. Between two neighbouring pixels that goes from positive to negative, the one close to zero is regarded as the zero crossing point.

Since the zero laplacian cost cannot represent the strength of the gradient, the second cost is introduced. The second cost is defined based on the gradient of the image directly, shown as follows

$$f_G = \frac{\max(G) - G}{\max(G)} = 1 - \frac{G}{\max(G)} \quad (2.54)$$

$$G = |\nabla I| = \sqrt{I_x^2 + I_y^2} \quad (2.55)$$

where  $I_x$  and  $I_y$  are the partial derivatives of image  $I$  in  $x$  and  $y$  directions respectively. The gradient cost is further scaled by the Euclidean distance. If  $q$  is a diagonal neighbour to  $p$ ,  $f_G$  is scaled by 1. If  $q$  is a horizontal or vertical neighbour of  $p$ ,  $f_G$  is scaled by  $1/\sqrt{2}$ .

The last cost  $f_D$  is the gradient direction cost [52]. Let  $D(\mathbf{p})$  be the unit vector with the gradient direction at point  $p$ , and  $D'(\mathbf{p})$  is the unit vector perpendicular to  $D(\mathbf{p})$ . That is,

$D(\mathbf{p}) = [I_x(\mathbf{p}), I_y(\mathbf{p})]$ ,  $D'(\mathbf{p}) = [I_y(\mathbf{p}), -I_x(\mathbf{p})]$ . In fact,  $D'(\mathbf{p})$  points along the boundary direction. The direction cost is defined as

$$f_D(\mathbf{p}, \mathbf{q}) = \frac{2}{3\pi} \{ \text{acos}[d_p(\mathbf{p}, \mathbf{q})] + \text{acos}[d_q(\mathbf{p}, \mathbf{q})] \} \quad (2.56)$$

where the vector dot products are

$$\begin{aligned} d_p(\mathbf{p}, \mathbf{q}) &= D'(\mathbf{p}) \cdot L(\mathbf{p}, \mathbf{q}) \\ d_q(\mathbf{p}, \mathbf{q}) &= D'(\mathbf{q}) \cdot L(\mathbf{p}, \mathbf{q}) \end{aligned} \quad (2.57)$$

and the unit link vector is

$$L(\mathbf{p}, \mathbf{q}) = \frac{1}{\|\mathbf{p} - \mathbf{q}\|} \begin{cases} \mathbf{q} - \mathbf{p}, & D'(\mathbf{p}) \cdot (\mathbf{q} - \mathbf{p}) \geq 0 \\ \mathbf{p} - \mathbf{q}, & D'(\mathbf{p}) \cdot (\mathbf{q} - \mathbf{p}) < 0 \end{cases} \quad (2.58)$$

Here the unit link vector either points from  $\mathbf{p}$  to  $\mathbf{q}$ , or points from  $\mathbf{q}$  to  $\mathbf{p}$ , depending on the boundary direction at point  $\mathbf{p}$ . By making the direction of the link vector approximately consistent with  $D'(\mathbf{p})$ , the sum  $\{ \text{acos}[d_p(\mathbf{p}, \mathbf{q})] + \text{acos}[d_q(\mathbf{p}, \mathbf{q})] \}$  in Eq. 2.56 calculates the sum of the two angles, that is, the angle between the link vector and  $D'(\mathbf{p})$ , and the angle between the link vector and  $D'(\mathbf{q})$ .

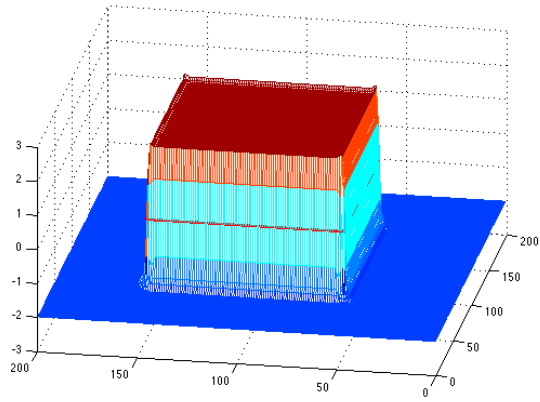
When the boundary directions at neighbouring points are similar with the direction of the link vector between them, the direction cost of the edge between these two points is small because the two included angles are small. However, if the boundary direction at the two points goes across the link vector, the cost will be larger. If the three directions are all significantly different from each other, the cost reaches the largest. In this way, the gradient direction cost makes the contour go along the boundary direction, and adds a smooth constraint by giving high cost to sharp changes in gradient direction [50]. Since the maximum of the sum of the two angles mentioned above is  $3\pi/2$ , the gradient cost is normalized by  $2/3\pi$  to make the maximum unity.

The laplacian zero-crossing cost, the gradient cost, and the direction cost all range from 0 to 1. Based on the above three kinds of cost, we can calculate the cost of every edge in the image. Given a starting point, we can get the optimal path from the starting point to every other points in the image using Dijkstra algorithm [17, 51]. The first starting point is just the first user-clicked point, and by clicking the second point, the intelligent scissors pick the optimal path from the first point to the second point from a large collection of the optimal paths. By using the next point as the new starting point, the paths are constructed stage by stage. Finally, we can get a close contour by finding the optimal path from the last clicked point to the first clicked point.

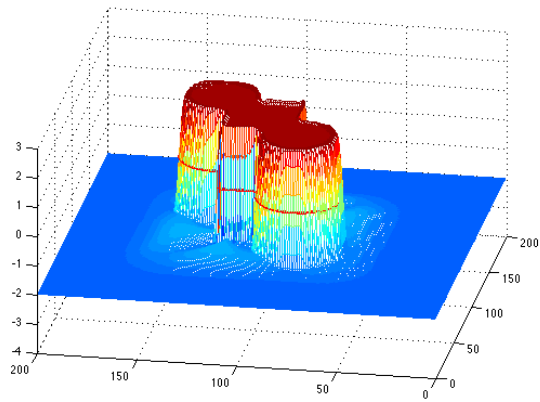
## 2.5 Summary

In this section, we mainly introduce the two categories of the active contour model, the parametric active contour model and the geometric active contour model. For paramet-

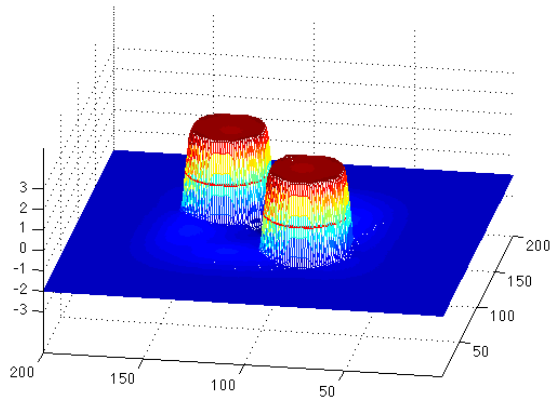
ric active contours, we have introduced the most popular snake model, the balloon force model, GVF model, and GGVF model. For geometric active contours, we have introduced the curve evolution-based model, the geodesic active contour model, and the Chan-Vese model. Although the geometric active contour model can handle topological changes, for noisy images, or images with the boundary gap, etc., the geometric models are easy to generate incorrect topology with respect to the real object. Therefore in these cases, it is better to choose parametric models due to their strict control in topology [76]. In the end of this chapter, we also introduce a snake-like technique called intelligent scissors. In the next chapter, we will introduce our proposed snake model called VFCCV.



(a)



(b)



(c)

Figure 2.5: The evolution of the level set function. The zero level set splits into two circles while the level set function is still a valid function. For a better view, the level set function is displayed upside down and thus has positive values inside and negative values outside. The zero level set is shown as the red curve.

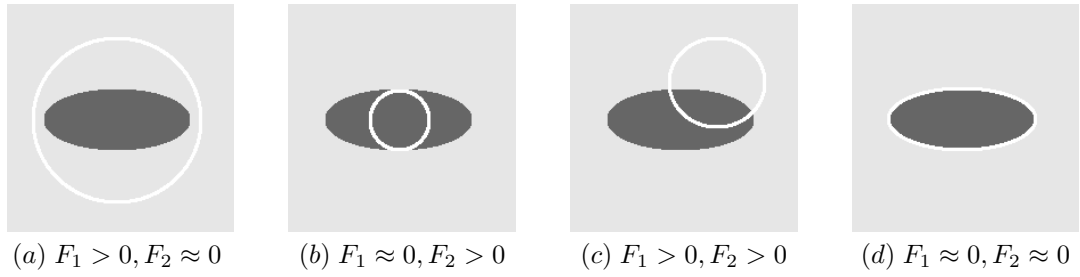


Figure 2.6: The region-based energy is minimized when the contour locates on the object boundary.

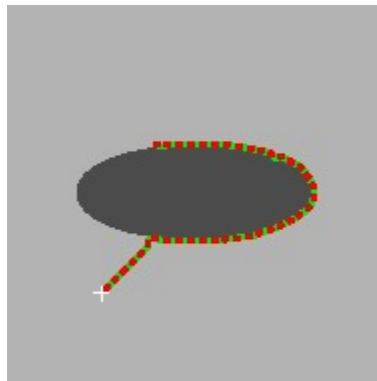


Figure 2.7: A typical interface of the lasso tool, also called as intelligent scissors.



## Chapter 3

# VFCCV Snake: A Novel Snake Model Combining Edge and Regional Information

As presented in the previous chapter, the active contour models have been widely used for image segmentation. We have introduced different kinds of active contour models in the previous chapter. However, most of the models either only consider edge information, or only consider regional information. In this chapter, we will introduce a new active contour model called VFCCV that takes both edge and regional information into consideration.

### 3.1 Introduction

The active contour models have been widely used in image segmentation [27] and object tracking [58]. As mentioned before, active contour models can be categorized into parametric active contour and geometric active contour. The most popular snake model defines a parametric active contour that evolves under the influence of internal energy and the external energy [27]. However, the classic snake model is known to have certain limitations, e.g., close initialization requirement and difficulty of convergence into concavities. To overcome these limitations, the gradient vector flow (GVF) snake model was proposed [77]. GVF creates a vector force field by a spatial diffusion of the edge map of the image so that the active contour can be driven toward strong edges far away from the current contour location. However, GVF suffers from its own weaknesses such as high computational cost, susceptibility to noise, and sensitivity to parameter selection. The vector field convolution (VFC) snake model [36] was later proposed to tackle the problems of GVF. VFC is also a parametric active contour model. With the VFC model, the active contour is evolved within an edge map generated with a user-defined vector field kernel. Unfortunately, VFC remains sensitive to noise because it only considers the edge information and ignores the regional information, and the resulting contour can be easily distracted by noise or trapped in a clut-

tered environment. As illustrated in Fig 3.1(a), the VFC snake is easy to be influenced by a single noisy spot.

In addition to the edge information, regional information is also important in the formulation of an active contour model that is robust to both noise and to weak edges [79]. Among different kinds of active contour models using region information [60, 83], the model proposed by Chan and Vese [6] has been most widely used. Based on the model proposed by Chan and Vese, we propose a novel active contour model that incorporates regional information in VFC. We refer to our model as VFC Chan-Vese (VFCCV) in the rest of this paper.

As introduced before, the Chan-Vese model detects the object boundary by minimizing the region-based fitting energy and the shape regularization energy. However, it only considers regional information, which makes the model pass the weak boundary easily, as illustrated in Fig. 3.1(b). Therefore it is also natural to consider incorporating edge information into a region-based active contour model. The VFC model is implemented in the parametric active contour framework, while the Chan-Vese model is in the geometric framework. [79] suggests that the parametric active contour model and the geometric active contour model are mathematically equivalent. In fact, the use of parametric versus geometric active contour models largely depends on the application. If the application is to delineate a single object, then it is easy to use a parametric active contour model, since the contour never changes its topology. On the other hand, using geometric active contour model for such application is often troublesome, because the program needs to prevent the occurrence of topological changes that is often performed in an ad-hoc manner. The geometric active contour models may easily generate shapes that have an inconsistent topology with respect to the actual object, especially when applied to noisy images with significant boundary gaps [76]. Since the VFCCV model is mainly applied on heart images to segment the ventricle which is obvious a single object, to ensure a correct topology, we implement our VFCCV model in the parametric active contour framework. Also, computationally speaking, the parametric active contour evolution involves solving a sparse and positive definite linear system, which is extremely efficient to solve these days. On the contrary, the geometric active contour model solves a gradient descent method, often much more computationally demanding than a comparable parametric active contour model evolution equation.

Although the idea to combine edge and regional information is common in the active contour framework [75, 68], most of them are geometric active contours and, to the best of our knowledge, incorporating the fitting energy in the VFC active contour model has never been studied. Specifically, the region-aided geometric snake (RAGS) model [75], a widely used method using both edge and region information, first gets the region boundary



Figure 3.1: (a) VFC (b) Chan-Vese model implemented in parametric framework

map through the region segmentation map. The boundary gradient of this region map can be regarded as the region force, to be diffused using the same technique as GVF, and thus the region force has enlarged capture range. The final external force is the sum of this diffused region force and the original gaussian external force in the snake model. Obviously, since this method requires the region boundary map in the preprocessing step, its performance is largely dependent on the results of region segmentation. In contrast, our VFCCV seamlessly integrates regional and edge information consistently in a single parametric active contour framework.

The rest of this chapter is organized as follows. In Section 3.2, we introduce the VFC model on which the VFCCV model is based. Then, we introduce the RAGS model with which we will compare the VFCCV model. In Section 3.3, we propose our VFCCV model followed by its numerical implementation. The summary of the chapter is drawn in Section 3.4.

## 3.2 Related Works

### 3.2.1 Vector Field Convolution

As introduced in Chapter 2, the energy functional of the snake model can be written as

$$E_{snake} = \int_0^1 \left[ \frac{1}{2}(\alpha|\mathbf{v}'(s)|^2 + \beta|\mathbf{v}''(s)|^2) + E_{ext}(\mathbf{v}(s)) \right] ds \quad (3.1)$$

Here, the first two terms in the integral are internal energy that exerts control on the topology of the contour and regularizes its shape. The last term is the external energy that attracts the contour to the region of interests. To minimized the energy, the snake should satisfy the Euler-Lagrange equation

$$\frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) - \nabla E_{ext} = 0 \quad (3.2)$$

By treating the snake as a function time  $t$ , we can get the solution to the above equation by the following gradate descent equation

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) - \nabla E_{ext} \quad (3.3)$$

As introduced before, the two main problems of the above snake model, i.e., the close initialization requirement and the difficulty to converge into concavities, are solved by the gradient vector flow (GVF) model. However, GVF still has disadvantages, such as noise sensitivity, parameter sensitivity, high computational cost, and the ambiguous relationship between parameters and the capture range [36]. The vector field convolution (VFC) model is proposed to solve the problems of GVF. Similar to the GVF model, the force field of VFC is also in the dynamic force formulation. The VFC field is calculated by convolving the edge image derived from the image with a vector field kernel. A user can define their own vector field kernels.

In the vector field kernel [36], all the vectors point to the centre of the kernel. The kernel can be defined as

$$\mathbf{k}(x, y) = m(x, y)\mathbf{n}(x, y) \quad (3.4)$$

where  $m(x, y)$  is the magnitude of the vector at point  $(x, y)$ .  $\mathbf{n}(x, y)$  is the unit vector pointing to the kernel centre  $(0, 0)$

$$\mathbf{n}(x, y) = (-x/(r + \epsilon), -y/(r + \epsilon)) \quad (3.5)$$

Here  $r$  is the distance from point  $(x, y)$  to the centre of the kernel, which can be defined as the Euclidean distance  $r = \sqrt{x^2 + y^2}$ .  $\epsilon$  is a small number to prevent dividing by zero at the centre point. Fig. 3.2.1 shows an example of the vector field kernel.

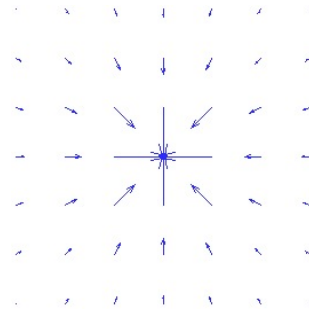


Figure 3.2: The vector field kernel. The centre point has the coordinate  $(0,0)$

The VFC field is calculated by convoluting the above kernel with the edge map derived from the image. In this way, the edge information is also propagated from the boundary regions to the homogeneous regions. So both the GVF and VFC fields are propagated from the edge, however, they propagate in different ways. The VFC force field can be defined as

$$\begin{aligned} \mathbf{f}_{vfc} &= (u_{vfc}(x, y), v_{vfc}(x, y)) \\ &= e(x, y) * \mathbf{k}(x, y) \\ &= (e(x, y) * u_k(x, y), e(x, y) * v_k(x, y)) \end{aligned} \quad (3.6)$$

Here  $e(x, y)$  is the edge map.

[36] provides two options for the magnitude of the vector field. They are both decreasing positive functions of  $r$ , the distance from the centre of the kernel. The two kinds of magnitude functions are:

$$m^1(x, y) = (r + \epsilon)^{-\gamma} \quad (3.7)$$

$$m^2(x, y) = \exp(-r^2/\zeta^2) \quad (3.8)$$

where  $\gamma$  and  $\zeta$  are controlling parameters.

Replacing the GVF field in Eq. 3.3 by the VFC force field, we get the evolution function of the VFC model as follows

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) + \mathbf{f}_{vfc}(\mathbf{v}) \quad (3.9)$$

The VFC model not only has the large capture range and the ability to converge to concave regions, but also has reduced computational cost and the flexibility of tailoring the force field [36].

### 3.2.2 Region-aided Geometric Snake

In this section, we will introduce a model that also combines region information and edge information, called the region-aided geometric snake (RAGS) [75]. In the experiment section, we will compare our method with this method.

The RAGS is a geometric active contour that combines the original gaussian edge force and regional force. The regional force is diffused from the gradient of the region segmentation map using similar technique as how the GVF force field is diffused. The regional force gives a global view of the edge information, and the edge force helps to detect local edges. The traditional active contour models suffer from either leaking into a weak boundary or stopping at a fake boundary or noise. By incorporating regional information with edge information, the RAGS model attempts to solve both of these problems.

The RAGS model requires a region segmentation technique. The goal of the region segmentation technique is to partition the original image into regions, and each region corresponds to homogeneous area in the original image. Different from an edge-based segmentation technique, for example, the active contour model whose output are edges between regions, region segmentation methods identify regions directly. Many region segmentation methods have been proposed [54, 19, 9, 16]. We will not include the details of these techniques in this thesis.

Based on any of the region segmentation methods, an image can be split into several regions distinguished by, for example, their dominant grayscale. Then, the RAGS calculates the magnitude of the gradient of the region segmentation map, which is called the region

boundary map. The gradient of the region boundary map can be regarded as the region force. However, this region force only takes local gradient into consideration and thus has small capture range. Using similar techniques as proposed in GGVF introduced in Section 2.2.4, the region force of RAGS is diffused into homogenous regions. Denote the region boundary map as  $R$ , and the region force field as  $\mathbf{f}_{rags}(x, y) = (u_{rags}(x, y), v_{rags}(x, y))$ . The RAGS regional force field is obtained by solving the following equations

$$\begin{aligned} g(|\nabla R|)\nabla^2 u_{rags} - h(|\nabla R|)(u_{rags} - \nabla R_u) &= 0 \\ g(|\nabla R|)\nabla^2 v_{rags} - h(|\nabla R|)(v_{rags} - \nabla R_v) &= 0 \end{aligned} \quad (3.10)$$

Similar to the GGVF model, here the weighting functions  $g(\cdot)$  and  $h(\cdot)$  are defined as

$$\begin{aligned} g(|\nabla R|) &= \exp\left(-\frac{|\nabla R|}{K}\right) \\ h(|\nabla R|) &= 1 - g(|\nabla R|) \end{aligned} \quad (3.11)$$

where  $K$  is a tuning parameter.

The RAGS model is constructed on the GAC model introduced in Section 2.3.2. The internal and external force of the snake formulation of the GAC model can be represented as

$$\begin{aligned} F_{int} &= g\kappa\mathbf{N} \\ F_{ext} &= gc\mathbf{N} - \nabla g \end{aligned} \quad (3.12)$$

The snake formulation of the RAGS model is built by adding the regional force in the original external force. So the internal force and external force of the snake formulation of RAGS can be represented as

$$\begin{aligned} F_{int} &= g\kappa\mathbf{N} \\ F_{ext} &= gc\mathbf{N} + \beta\mathbf{f}_{rags} - \nabla g \end{aligned} \quad (3.13)$$

where  $c$  and  $\beta$  are the weights of the corresponding terms. Then, we can get the evolution function

$$\frac{\partial \mathbf{v}}{\partial t} = g(\kappa + c)\mathbf{N} - \nabla g + \beta\mathbf{f}_{rags} \quad (3.14)$$

The model can be further transformed into level set representation. Since the performance of the RAGS model is largely influenced by the region force, the performance is highly dependent on the results of the region segmentation.

### 3.3 VFC Chan-Vese (VFCCV) Model

#### 3.3.1 Propose the VFCCV Model

As mentioned before, VFC snake is still sensitive to noise as illustrated in Fig. 3.1(a) where the final contour could be influenced by just a single noisy spot. One way to overcome the sensitivity of the VFC snake to noise is to not only consider edge information, but also

regional information when it is available. This is the key insight that motivates our proposed VFCCV snake. As well, it is important to incorporate edge information into a region-based model in order to overcome the problem of passing through a weak boundary in the Chan-Vese model as shown in Fig. 3.1(b).

Our regional force is defined based on the fitting energy proposed in the Chan-Vese model. As introduced in Section 2.3.3, the Chan-Vese model [6] defines a fitting energy as follows:

$$F_1(\mathbf{v}) + F_2(\mathbf{v}) = \int_{inside(\mathbf{v})} |I(x, y) - c_1|^2 dx dy + \int_{outside(\mathbf{v})} |I(x, y) - c_2|^2 dx dy \quad (3.15)$$

where  $c_1$  and  $c_2$  are the average intensities of the regions inside and outside contour  $\mathbf{v}$  respectively. Inspired by this fitting energy, we define a region-information map as  $-\lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2$ . For a point on the contour  $\mathbf{v}$ , we multiply its outward normal with the value of this map at this point to generate a valid force field in order to embed it into a parametric active contour model. We define the region force as follows.

$$\begin{aligned} \mathbf{f}_{region}(x, y) &= \mathbf{n}(x, y)(f_1(x, y) + f_2(x, y)) \\ &= \mathbf{n}(x, y)(-\lambda_1(I(x, y) - c_1)^2 + \lambda_2(I(x, y) - c_2)^2) \end{aligned} \quad (3.16)$$

where  $\mathbf{n}(x, y)$  is the outward normal vector at point  $(x, y)$ ,  $f_1(x, y) = -\lambda_1(I(x, y) - c_1)^2$ , and  $f_2(x, y) = \lambda_2(I(x, y) - c_2)^2$ . For a point  $(x, y)$  within the image domain, if this point is inside the desired object,  $I(x, y)$  is similar to  $c_1$ . In contrast, there is a relatively large difference between  $I(x, y)$  and  $c_2$ . As a result,  $f_1(x, y)$  is approximately 0 and  $f_2(x, y)$  is larger than 0 assuming  $\lambda_1 = \lambda_2$ .  $f_1 + f_2$  is positive, when multiplied by the outward normal, and the direction of the region force points outwards to pull this point toward the object boundary. Conversely, if the point is outside the object,  $f_2(x, y)$  is approximately 0 and  $f_1(x, y)$  is smaller than 0.  $f_1 + f_2$  is negative and the direction of the force points inwards.  $\lambda_1$  and  $\lambda_2$  are positive weighting parameters controlling the strength of shrinking or expanding the contour respectively. The introduction of  $\lambda_1$  and  $\lambda_2$  can improve the model's flexibility.

For us to compute Eq.3.16, we need to define  $c_1$  and  $c_2$ . Generally, the background is often cluttered especially in medical images, while the intensity of the foreground object is distinct from the background and often homogeneous. Therefore, to implement our VFCCV snake, we use the average intensity inside the contour for  $c_1$  as usual, while we use as  $c_2$  the average intensity of the annular band immediately outside the contour. As in the Chan-Vese snake, both  $c_1$  and  $c_2$  are updated iteratively as the contour evolves. This is fine when the initialization is outside the object. When the initialization is inside the object, even though the annular band is not large enough to include a portion of the background, the diffused edge force, which enlarges the capture range, can still drag the contour outwards to the boundary until the annular band includes the background and allow the region force to take effect.

In the parametric active contour model, external forces can be expressed as the superposition of several different forces where each force is manageable [76]. In our case when we add the regional force with the VFC edge force to generate the final external force, the final evolution function becomes

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) + \gamma \mathbf{f}_{vfc}(\mathbf{v}) + \lambda \mathbf{f}_{region}(\mathbf{v}) \quad (3.17)$$

where  $\gamma$  and  $\lambda$  are weighting parameters controlling the strength of VFC edge force and regional force.

### 3.3.2 Numerical Implementation

Eq.3.17 can be solved by the finite difference method [27]. We use the vector notation to represent a point,  $\mathbf{v}_i^n = (x_i^n, y_i^n) = (x(ih, n\Delta t), y(ih, n\Delta t))$ . Here  $i$  indicates the index of the location of point on the contour and  $n$  indicates the index of time.  $h$  is the space step size, and  $\Delta t$  is the time step size. By discretizing the active contour, we can represent Eq. 3.17 as

$$\begin{aligned} \frac{\mathbf{v}_i^n - \mathbf{v}_i^{n-1}}{\Delta} &= \frac{\alpha}{h^2} [(\mathbf{v}_{i+1}^n - \mathbf{v}_i^n) - (\mathbf{v}_i^n - \mathbf{v}_{i-1}^n)] \\ &\quad - \frac{\beta}{h^4} [(\mathbf{v}_{i-2}^n - 2\mathbf{v}_{i-1}^n + \mathbf{v}_i^n) - 2(\mathbf{v}_{i-1}^n - 2\mathbf{v}_i^n + \mathbf{v}_{i+1}^n) + (\mathbf{v}_i^n - 2\mathbf{v}_{i+1}^n + \mathbf{v}_{i+2}^n)] \\ &\quad + \gamma \mathbf{f}_{vfc}(\mathbf{v}_i^{n-1}) + \lambda \mathbf{f}_{region}(\mathbf{v}_i^{n-1}) \end{aligned} \quad (3.18)$$

The detailed derivation of the above equation can be found in Appendix B. The external force  $\mathbf{F}_{ext} = \gamma \mathbf{f}_{vfc} + \lambda \mathbf{f}_{region}$  can be stored as a  $height \times width \times 2$  vector field defined on the image domain, where  $height \times width$  is the size of the image. The first layer records the force component in  $x$  direction, and the second layer records the force component in  $y$  direction. In this way, we can get the external force at any point in the image from the vector field map.

Eq. 3.18 can be written in a compact matrix form shown as follows [27, 76, 36]

$$\frac{\mathbf{v}^n - \mathbf{v}^{n-1}}{\tau} = \mathbf{A}\mathbf{v}^n + \gamma \mathbf{f}_{vfc}(\mathbf{v}^{n-1}) + \lambda \mathbf{f}_{region}(\mathbf{v}^{n-1}) \quad (3.19)$$

where  $\tau = \Delta t$ .  $\mathbf{v}^n, \mathbf{v}^{n-1}$ , and the external force  $\mathbf{F}_{ext}(\mathbf{v}^{n-1}) = \gamma \mathbf{f}_{vfc}(\mathbf{v}^{n-1}) + \lambda \mathbf{f}_{region}(\mathbf{v}^{n-1})$  are  $m \times 2$  matrices, where  $m$  is the number of points on the contour.  $\mathbf{A}$  is an  $m \times m$  pentadiagonal banded matrix used to compute the internal force [76, 36]. Based on Eq. 3.19, the contour can be solved iteratively through the following equation

$$\mathbf{v}^n = (\mathbf{I} - \tau \mathbf{A})^{-1} [\mathbf{v}^{n-1} + \tau (\gamma \mathbf{f}_{vfc}(\mathbf{v}^{n-1}) + \lambda \mathbf{f}_{region}(\mathbf{v}^{n-1}))] \quad (3.20)$$



## 3.4 Summary

In this chapter, we have introduced a new active contour model called VFCCV snake, which adds Chan-Vese type regional information to the VFC active contour model. We have also introduced the numeral implementation of VFCCV. Also, we have reviewed a competitive model called RAGS which also considers both edge and regional information. We will provide experimental results comparing the performance of the VFCCV, VFC, RAGS, and Chan-Vese methods in Chapter 5. A system that segments the 3D ultrasonic image of the left heart ventricle using VFCCV will be presented in the next chapter.

## Chapter 4

# 3D Echocardiography Segmentation Embedded with Automatic Validation

In the previous chapter, we have introduced a new 2D active contour model. In this chapter, we will talk about how to use this model to segment 2D echocardiography slice sequence and finally get the 3D ventricle volume. A validation step is embedded in the segmentation system to automatically decide whether the segmentation result on each slice is correct.

### 4.1 Introduction

Currently, the left ventricle ejection fraction<sup>1</sup> is one of the most validated and valuable measurements for determining a patient's health [32]. There is strong evidence showing that the decrease in ejection fraction is associated with an increase in mortality [13, 73]. Obviously, only an accurate ventricular volume calculation can generate an accurate ejection fraction. Also, the volume of ventricle is a significant diagnostic indicator of diseases like dilated cardiomyopathy, etc. [33, 34, 73]. The importance of volume calculation to these cardiac functional parameters leads to the requirement of computer-aided cardiology segmentation. In this chapter, we will only deal with the echocardiography segmentation for the reason that the echocardiography is relatively inexpensive and easy to obtain in practice.

Clinically speaking, there are three main methods used by physicians to obtain the 3D left ventricle volume from the echocardiography [73]. The first one is the Simpson's bi-plane method [33]. The echocardiography is first cut into several stacked discs and the final volume is the sum of these discs volumes. Each of the disc volumes is determined by manually delineated borders based on the assumption that the ventricle is an ellipsoid and the border is an ellipse. The Simpson's method is available on both 2D and 3D echocar-

---

<sup>1</sup>ejection fraction =  $\frac{\text{end-diastolic volume} - \text{end-systolic volume}}{\text{end-diastolic volume}} \times 100$

diography. However, the assumption of an ellipsoid shaped ventricle makes the calculation insufficiently accurate. The second method is a manual method. Similar to the first method, the left ventricle is first sliced into several discs and the ventricle volume is also the sum of these disc volumes. Each disc volume is determined by direct manual area tracing, which requires lots of manual work. The result of this method is neither accurate nor reproducible as the third method, which is a semi-automatic method based on border identification. In the third method, the ventricle volume is calculated by the number of voxels inside the segmented ventricle. A user are required to input five landmarks, and the method can automatically detect the 3D ventricular border and calculate the volume. However, this method has problems delineating some abnormal left ventricle shapes, e.g., apical infarcts.

In medical imaging research, echocardiography segmentation methods have also been studied in both 2D and 3D. For example, 3D active appearance model [48], 3D level set model [10], etc. Compared with 3D echocardiography, the anatomic structure is more easily defined and displayed on a 2D slice. Also, due to practical reasons like lower memory requirements, reduced computational complexity, ease of implementation, it is an intuitive way to apply a 2D segmentation method sequentially to the slices of a 3D images [57]. Therefore it is a good way to segment the 3D image data through 2D slices.

The 3D echocardiography data is first divided into a sequence of 2D slices. Due to the spatial continuity between neighbouring slices, the segmentation result of the previous slice can be copied to the next slice serving as a reasonable initialization. Then, a contour is evolved in the new slice. In this way, the segmentation propagates along the 2D slice sequence. This propagation idea has also been mentioned in earlier papers [7, 57]. In this chapter, we will also use this propagation idea. Specifically, the propagation method is used in [42]. It segments 2D prostate ultrasound images slice by slice to obtain the 3D prostate volume. Initialized points are selected in the mid-gland slice which normally contains the largest and the most visible part of anatomic structure. The method uses a warped and tapered ellipse to fit each slice and an ellipsoid to fit the entire prostate. The ellipse is propagated from the middle to the base and apex slices in opposite directions. For our echocardiography case, it is not accurate to approximate each slice using an ellipse. However, we will also initialize from the middle slice and propagate the result upwards and downwards until the apex and base slices are reached. We will apply the VFCCV model proposed in the previous chapter as the 2D segmentation method.

The propagation approach, including contour initialization and evolution, should ideally be fully automatic. However, the 2D segmentation of a slice may not be accurate. If the segmentation of one slice deviates from the true boundary, the segmentation of the following slices is also most likely erroneous, as illustrated in Fig. 4.1(a), where the middle ellipse in solid blue is initialized manually, the hollow blue ellipse immediately below it is

segmented correctly but not the red ellipses thereafter. In this case, we could either tune the parameters or change the initialization and redo the experiments. However, such a process might never produce the correct final volume nor would it be automatic. One alternative would be to manually check the segmentation result of every slice. This however would be tedious and in fact unnecessary when most of the slices are correctly segmented. Therefore, we propose to introduce a validation step which aims at automatically judging whether the segmentation on each slice is correct. With the validation step, the system can detect the potentially wrong segmentation and stop to ask the user for a correction. The propagation continues after the correction is given, as illuminated in Fig. 4.1(b) where the green curve is the correction for the red curve and leads to correct segmentations of the subsequent slices below it due to the correct initialization.

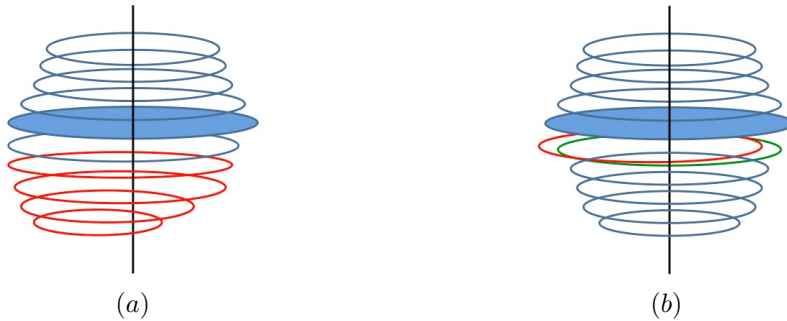


Figure 4.1: (a) Without the validation step (b) With the validation step

In the literature, the active contour validation is often ignored and left for the application to decide [61]. A snake validation method is proposed in [61]. This method builds a classifier to classify the contours as desired objects and non-objects. The classifier is based on the principal component analysis (PCA). An annular band around the active contour is first constructed and each annular band is a pattern image for PCA. Then, the new pattern image is projected into the already trained PC space and the reconstruction error is computed. The contours with a low reconstruction error are identified as objects and others are identified as non-objects. We adapt this method to our case. Rather than classifying the entire contour, the validation is done locally to detect local errors. In addition, we choose the PCA training data based on the location of each slice, to take into account the spatial variance of PC's along the vertical axis of 3D image. Our method will be introduced in detail in the next section. For each newly evolved contour, the validation step is performed to automatically decide whether the contour is accurate. Therefore, the validation step is embedded in the propagation system to make the whole segmentation process as automatic as possible while maintaining a high accuracy.

A 3D left ventricle segmentation method is proposed in [11]. They provide a 3D initial-

ization method for the 3D ventricle. A few slices among the sequence are first selected. On each slice, an approximate polygon is drawn by the physician. Then, they use linear interpolation among these slices to fill up the whole ventricle. Although their method is performed in 3D directly and our method is in 2D, we both provide a way to prevent the segmentation from deviating from the true boundary. [11] does it by initializing the 3D shape close to the real boundary and we do it through propagation and validation. However, the linear interpolation in [11] is difficult to implement because they need to pair the corresponding points between slices.

Our overall system is introduced in Section 4.2. The discussion is given in Section 4.3 followed by a summary of the chapter.

## 4.2 2D Contour Propagation and Validation

### 4.2.1 Segmentation and Propagation

The 3D echocardiography data is first divided into a sequence of 2D slices. The process begins by the user to choose the base and apex slices between which the left ventricle is contained. Usually, the middle slice between the two extreme slices contains the largest and the most visible part of the ventricle. The segmentation process begins at the middle slice. It is selected as the initial slice which needs a manual initialization. In our system, we use intelligent scissors [50] introduced in Chapter 2 for the initial segmentation, which can trace the boundary intelligently. The segmentation propagates upwards and downwards from the middle slice. Due to the spatial continuity of the cross-sectional areas of neighbouring slices, the segmentation of the previous slice can be copied to the next slice to serve as a reasonable initialization. On each new slice, an active contour model is run to evolve the initialized contour.

Although the neighbouring slices look similar with each other, there can still be a large change in the ventricle boundaries of two neighbouring slices. So we need to choose an active contour model that has a large capture range. As mentioned in Section 2, many models have been proposed to enlarge the capture range. Among these models, the GVF [77] model can obtain better results than the other models. The VFC [36] model was later proposed to tackle the problems of the GVF model. Therefore, we choose to take advantage of the VFC model. Also, we need to take the region information into consideration to make the system more robust to noise and not easily trapped in cluttered regions. The Chan-Vese model is a widely-recognized model as a regional model. So we use the VFCCV model proposed in the previous chapter, which is a combination of the VFC and the Chan-Vese. The evolution function is defined as

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial v}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 v}{\partial s^2} \right) + \gamma f_{vfc}(v) + \lambda f_{region}(v) \quad (4.1)$$

All the symbols and parameters here are the same as have been introduced before. The propagation stops until the base or apex slices are reached.

### 4.2.2 Contour Validation

The validation step is in fact an image classification process classifying a contour as accurate and inaccurate based on the principal component analysis (PCA). If we construct an annular image band around the contour, as shown in Fig. 4.2(a), valid and invalid boundaries are visually distinct. For example, the greyscale inside the ventricle is darker, and the greyscale outside is brighter. So if we unfold the annular band, one side should be darker, and the other side should be brighter, as shown in the top image in Fig. 4.2(c). Even if only part of the contour deviates from the true boundary, the whole contour should be classified as inaccurate. Therefore, we should not use the annular band for the whole contour to detect this small error. Instead, the annular band is divided into local segments, and the validation is also performed locally for each segment. The local annular band segment is shown in Fig. 4.2(b). If a part of the contour is inaccurate and we just cut this part into two non-overlapping segments, the PCA reconstruction errors in both the left segment and the right segment might be too small for the validation to detect the mistake. Therefore, each segment is overlapped with its neighbours by, for example, 67%, in order to be sensitive enough to detect all the local errors, as shown in Fig. 4.2(c). In our case, each annular band segment is considered as a pattern image.

The training pattern images should have the same format as the testing pattern images. To clearly introduce how to construct all the training pattern images, we will first introduce the format and characteristics of our dataset. From the end-diastolic status to the end-systolic status in each heart cycle, the image is 4D with dimensions  $w \times h \times d \times t$ . Here  $t$  is the number of 3D images in the time series from the end-diastolic status to the end-systolic status.  $w \times h \times d$  is the size of a 3D image of one heart. Due to the different imaging environments for different 4D images, each 4D image may look different from one another. Therefore, different 4D images should use different training data for PCA. In each 4D image, the  $t$  3D images look similar with each other because they are generated under the same imaging environment. So different 3D images in one 4D image can use the same set of training data. For example, in a certain 4D image, the training data for the 3D images from index two to index  $t$  can all come from the 3D image with index one. However, in each 3D image, the top few slices may look very different from the bottom few slices. So different slices in one 3D image require different PCs. The observation - the slices at the same location across different 3D images look more similar than the slices at difference locations in the same 3D image - can be demonstrated in Section 4.3.

Based on the above analysis, for each 4D image, the training data for all the 3D images

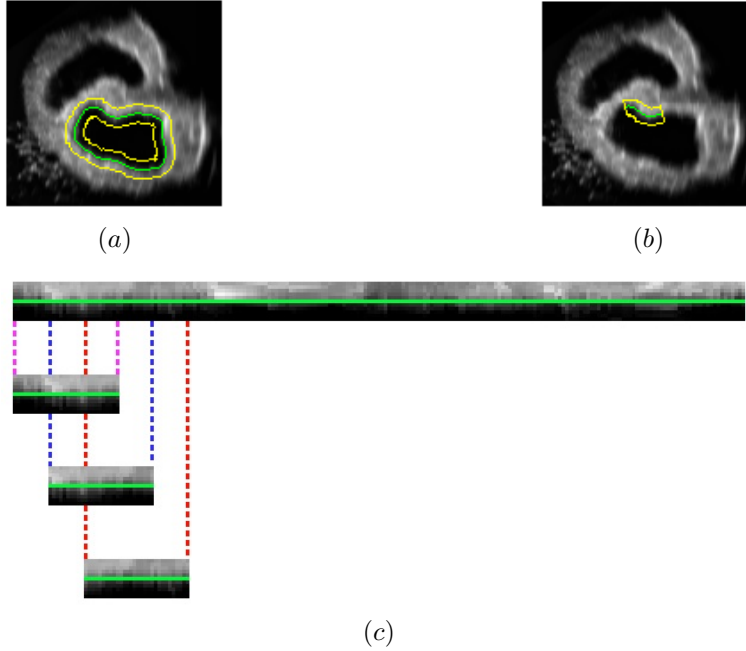


Figure 4.2: (a) The annular band for the whole contour (b) The annular band segment (c) Construct local and overlap segments from the whole unfolded annular band

are taken from the first 3D image in the time series, which is shown in Fig 4.3. Since the heart volume of the left ventricle decreases from the end-diastolic status to the end-systolic status, for the current slice in a certain 3D image, we need to first find its mapped slice in the first 3D image. The mapped slice for the current slice  $j$  is  $i$ , where

$$i = i_1 + (i_n - i_1)(j - j_1)/(j_n - j_1) \quad (4.2)$$

where  $i_1$  and  $i_n$  are the apex and base slices of the first 3D image respectively,  $j_1$  and  $j_n$  are the apex and base slices of the current 3D image, as shown in Fig. 4.4. The mapped slice  $i$  is rounded to the nearest integer. We take the mapped slice together with its upper and lower slices to form the training set of the current slice. If the mapped slice is the apex slice, the top three slices are chosen as the training slices. If it is the base slice, the bottom three are chosen.

For the current slice in the current 3D image, the local and overlapped segments are constructed from the three training neighbouring slices in the first 3D image. Each segment in the training data is a pattern image in the training set. Each pattern image is a feature vector, and all the pattern images form the feature matrix. Assume we have  $p$  pattern images in the training set for the current slice. Each pattern image,  $I_i$ , with size  $m \times n$ , is reshaped into a vector  $f_i$ , with size  $M \times 1$ , where  $M = mn$ . Then, the  $p$  pattern images form a matrix,  $[f_1, f_2, \dots, f_p]$ , with size  $M \times p$ . Then, this matrix is centred by subtracting

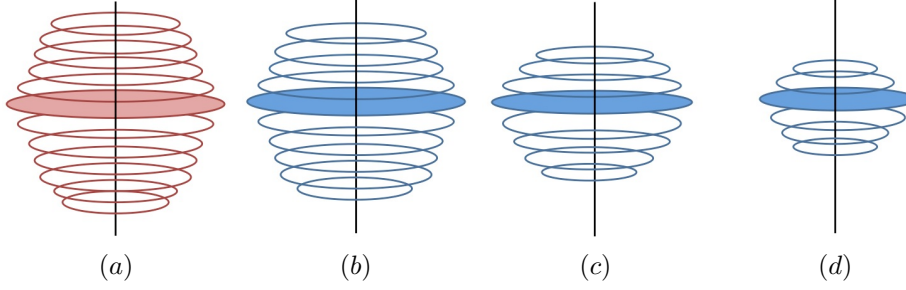


Figure 4.3: For a 4D image, the training data is taken from the first 3D image which is shown in red

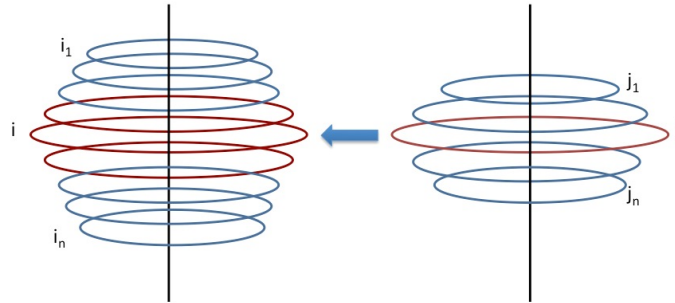


Figure 4.4: Find the mapped slice and take the neighbouring two slices

their mean. That is,

$$F = [f_1 - \bar{f}, f_2 - \bar{f}, \dots, f_p - \bar{f}] \quad (4.3)$$

where  $\bar{f} = (1/p) \sum_{i=1}^p f_i$ .

Then, we perform PCA on the centred feature matrix. Through singular value decomposition [21, 61], the matrix  $F$  can be decomposed as

$$F_{M \times p} = U_{M \times M} \Sigma_{M \times p} (V_{p \times p})^T \quad (4.4)$$

Here  $\Sigma$  is the diagonal matrix whose diagonal elements  $s_i$  are the singular values of  $F$ . The  $s_i$  are ordered so that  $s_i \leq s_{i+1}, i = 1, 2, \dots, p - 1$ .  $U$ 's columns are eigenvectors of  $FF^T$  and  $V$ 's columns are eigenvectors of  $F^T F$ . Both  $U$  and  $V$  are unitary matrices. We perform PCA on the matrix by keeping the first  $d$  columns of  $U$ ,  $d \ll M$ .

There are several ways of choosing the number of components. For example, retaining the component with singular value greater than 1 (the value-one criterion), keeping the number of components to the number that account for a certain cumulative percentage (percentage of variance method), plotting the singular values and finding a break point after which the values begin to level off (the scree test), etc. [25, 26]. Since the value-one criterion may generate lots of problems [25] and the percentage of variance method needs an extra subjective parameter for the amount of the percentage, we choose the number  $d$ , that is, the number of PCs, based on the scree plot of the singular values. A typical



scree plot in our case is shown in Fig. 4.5(a). Normally the top few values are significantly larger than the others. Instead of choosing the numbers by simply visual observation, we calculate the derivative of each point in the scree plot, and the PCs are kept up until the one with the largest derivative. The derivative is calculated using both the left and right neighbours of each point. We duplicate the first and last points in order to calculate the derivative for the first and last points. Fig. 4.5(b) shows the derivative of each singular value in Fig. 4.5(a). In Fig. 4.5(b), the point with the maximum derivative is shown in red and its corresponding singular value is also shown in red in Fig. 4.5(a). In this case, the second singular value has the largest derivative and thus the number of PCs is chosen as two.

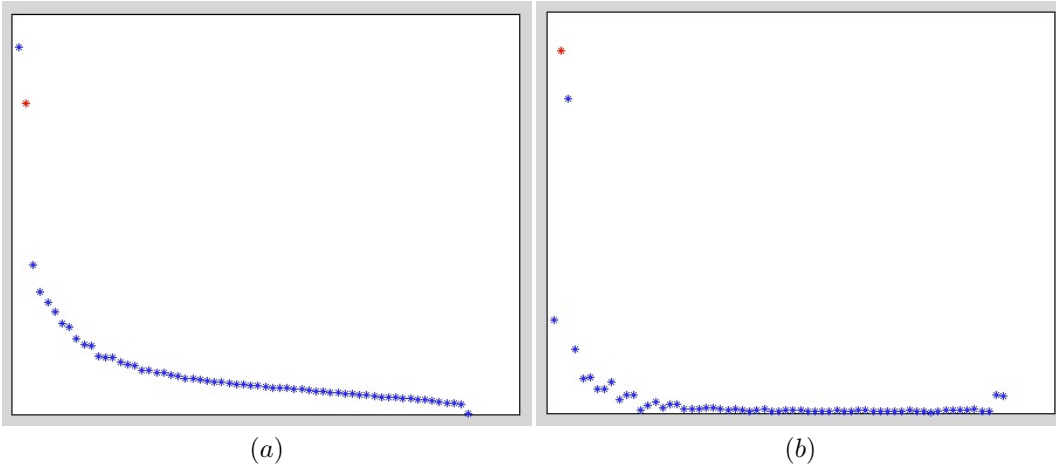


Figure 4.5: (a) The eigenvalues (b) The derivative of the eigenvalues

After PCA on the matrix  $F$ , the new reshaped pattern image  $f$  on the testing contour is then projected into the PC-subspace composed of the first  $d$  columns of  $U_{M \times p}$ . Note that the new reshaped pattern image is also constructed from the local and overlap segment on the testing contour. The projected result  $\tilde{f}$ , is calculated by

$$\tilde{f} = \bar{f} + U_{M \times d}(U_{M \times d})^T(f - \bar{f}) \quad (4.5)$$

Then, the PCA reconstruction error for the testing segment  $f$  is computed, which is defined as

$$\epsilon = \|f - \tilde{f}\|^2 \quad (4.6)$$

where  $\|\cdot\|$  is the Euclidean distance. If the reconstruction error is small, e.g., lower than a threshold, the contour segment is classified as an accurate contour segment. Otherwise the segment is classified as an inaccurate one.

If all the local and overlap pattern images on the contour are accurate, the whole contour is judged as accurate and passes the validation. The propagation and validation process

continues. If one or some of the pattern images are inaccurate, the contour fails the validation and the propagation process pauses to ask the user to decide. The flowchart of the above validation step is shown in Fig. 4.6. If the contour fails the validation, the user can choose to accept the current result or reject it. If it is been accepted, the propagation and validation process continues. If it is rejected by the user, the user needs to give a correction. The correction can be a reinitialization for either the whole contour or just the failed segments. We also use intelligent scissors [50] here for reinitialization. After the system produces a satisfactory contour, the propagation and validation process continues. The segmentation result first propagates from the middle slice to the apex slice, and then propagates from the middle slice to the base slice. The ventricle volume can be computed after we get all the 2D segmentations.

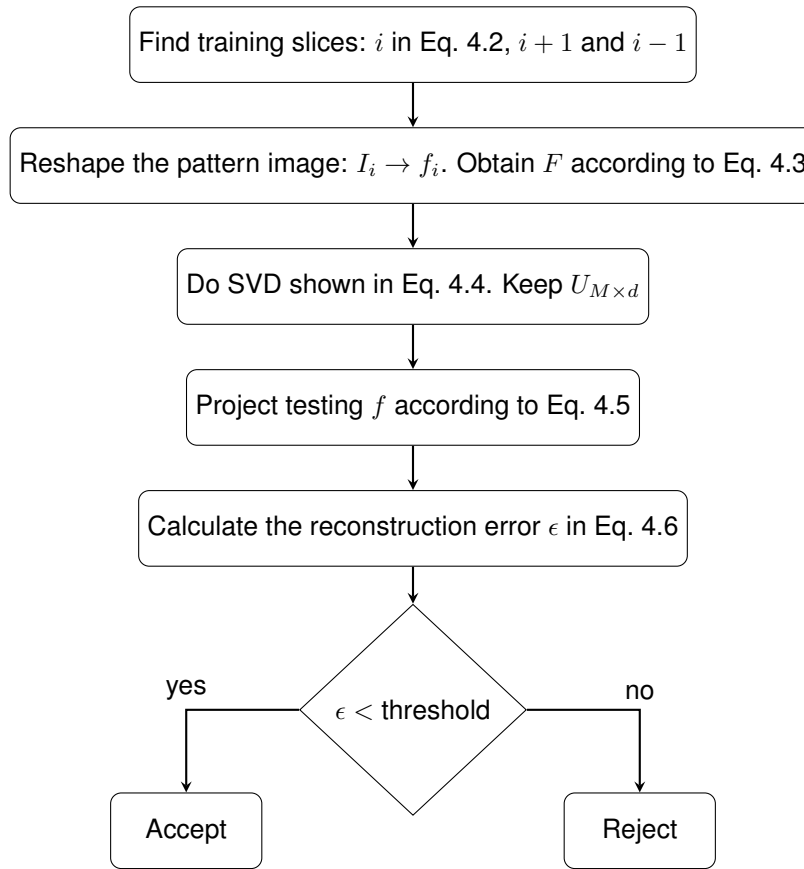


Figure 4.6: Flowchart of the validation step

### 4.3 Discussion

As mentioned above, we choose three neighbouring slices to form the training set, as shown in Fig. 4.4. In the experiments, we also test the performance of choosing just one

slice or five or more slices. However, the performance is not better than choosing three slices. The experiments and analysis will be given in the next chapter.

The observation mentioned above - the slices at the same location across different 3D images look more similar than the slices at difference locations in the same 3D image - can be demonstrated in Fig. 4.7. The red curves show the intra-3D image difference, and the blue curves show the inter-3D image difference. The difference is measured based on the local and overlap pattern images, and each group contains three neighbouring slices. The eight points on the red curves show the differences between the three slices at the relatively top location in a certain 3D image (called as “the standard three slices”) and the eight following groups of slices until the relatively bottom location in the same 3D image. The eight points on the blue curves show the differences between “the standard three slices” and the three mapped slices at the same location in the following 3D images. Fig. 4.7(a) measures the difference between the means of each group of pattern images. Since the length of the PC is normalized, the difference can be measured by the angle between the hyperplanes defined by each group of PCs. Fig. 4.7(b) shows the angles between the hyperplanes. The number of PCs is chosen as two.

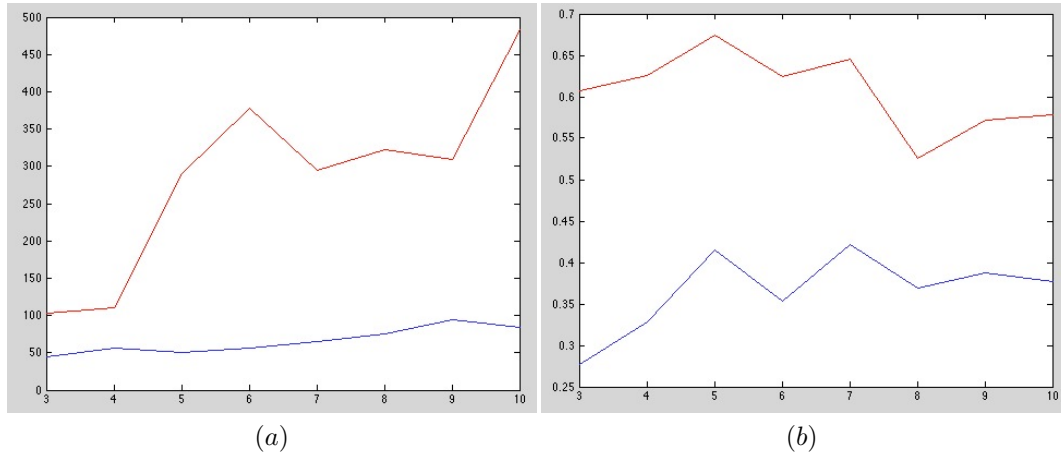


Figure 4.7: Red curves: intra-3D image difference. Blue curves: inter-3D image difference. (a) Difference between the means of each group of pattern images. (b) Angle between the hyperplanes defined by each group of PCs

The PCA validation step is in fact building a classifier to classify the contour as accurate or inaccurate. We can also use other techniques to build the classifier, for example, AdaBoost [20]. However, AdaBoost is a supervised binary classifier. In our propagation process, we can easily generate accurate contours. However, the inaccurate contour is difficult to obtain. Also, we cannot arbitrarily draw the inaccurate contours because they should be similar enough to the real inaccurate segmentations in order to simulate them. So it is difficult for us to build a balanced training data to train the AdaBoost classifier. Even

if we have a balanced training data, how to choose the images features is another problem, just like other one-class classifiers. For example, classifiers based on the difference between the grayscale outside the annular band and inside, entropy, gradient of the middle line of the annular band, GICOV [18], etc. They would all lead to the problem of how to choose image features to have a better performance. Also, if we need several features to perform the classification, we need to extract every feature from the images which would be time-consuming. However, the PCA validation provides a uniform and less complex way to perform the classification.

## **4.4 Summary**

In this chapter, we introduce an automatic validation step in the segmentation system to segment the 2D slice sequence. The validation step can judge whether the contour on each slice is accurate in order to make the whole process as automatic as possible while keeping a high segmentation accuracy. We use the VFCCV model introduced before to perform the segmentation. The validation is based on PCA. The pattern images are local and overlap to detect local segmentation errors. We also propose to choose the training data for each slice based on their location. Experimental results are shown in the next chapter.

# Chapter 5

## Experimental Results

Experimental results are shown in this chapter. The experiments are composed of two parts. The first part shows the performance of the VFCCV model introduced in Chapter 3. The second part investigates the performance of the propagation and validation algorithm introduced in Chapter 4.

### 5.1 Performance of VFCCV Model

Experimental results on both synthetic and real images are shown in this section, to establish the superiority of our method in comparison with the VFC snake, Chan-Vese snake and the RAGS model.

We first show in Fig. 5.1 the results on a toy example, a contrived image with a U-shaped gradient object and a single noisy spot. Using the red dashed curve as the initialization, Fig. 5.1(a) is the result of VFC. We can see that without region force, VFC is easily affected by noise. Note that the original Chan-Vese model uses the geometric active contour framework and is thus not directly comparable with parametric active contour models. In this example, the original Chan-Vese model would over-segment the U-shaped object. Specifically, Fig. 5.1(b) is the result of the Chan-Vese snake implemented in the parametric contour framework, which only considers the region force. Its evolution function is

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) + \lambda \mathbf{f}_{region}(\mathbf{v}) \quad (5.1)$$

We can see in Fig. 5.1(b) that without the help of edge information, the contour leaks through the right side with weak boundary information. Fig. 5.1(c) is the result of the RAGS model [75] which is also implemented in the parametric contour framework for a fair comparison. This method would not be influenced by the noisy spot because in the region segmentation map, the noise is so small that it is merged into the background. However, the contour fails to converge into the concavity even when the weight of the region force increases, in which case the contour would first leak through the right side of the U shape.

Fig. 5.1(d) is the result of our VFCCV method which converges to the correct U-shape boundary and is robust to the noise. In this experiment, we set  $\alpha = 1.5, \beta = 0, \gamma = 1, \tau = 0.5, \lambda = 1.5, \lambda_1 = \lambda_2 = 1$ . These parameters remain largely unchanged for the other experiments that we have conducted and described in the rest of this section.  $\gamma$  and  $\lambda$  are two key parameters in our method in Eq. 3.17. For the edge force to dominate the evolution, we need a larger  $\gamma$  and a smaller  $\lambda$ . Conversely, for the region force to dominate, we need a smaller  $\gamma$  and a larger  $\lambda$ . To be fair, all the parameters are optimized for all the competing algorithms separately in all the experiments.



Figure 5.1: Comparison using image with U-shaped gradient object (a) VFC (b) Chan-Vese Snake (c) RAGS and (d) VFCCV

Fig. 5.2 shows the segmentation results on an ultrasound image of the left ventricle of the heart.<sup>1</sup> The initialization is a circle inside the ventricle. Fig. 5.2(a) is the segmentation result of the VFC which is degraded by nearby stronger edges in cluttered regions. Fig. 5.2(b) is the result of the Chan-Vese snake, which leaks near the weak boundary. Fig. 5.2(c) is the result of RAGS, which also leaks through weak boundaries. Fig. 5.2(d) is the result of our VFCCV method, which in contrast produces a more accurate contour near the weak boundaries. This difference in performance is largely due to the manner in which the region information is expressed in the energy functions of RAGS and VFCCV. Even though the boundary information is weak, there is still a sufficiently strong VFC force field which our method is able to exploit to determine the contour and prevent the contour from leaking outward.

Experiments are also done on human echocardiography with results shown Fig. 5.3. The initialization for the four cases is the contour slightly larger than the real boundary. Fig. 5.3(a) is the result of VFC which is again caught in cluttered regions. It is because the VFC model only considers edge information and the edge force traps the boundary in the cluttered regions. Fig. 5.3(b) is the result of Chan-Vese snake which leaks through weak boundary. Fig. 5.3(c) is the result of the RAGS model. Since the weak boundary and the background are merged into one region in the region segmentation map, the region force is inaccurate and thus unable to drag the contour to the true boundary. Fig. 5.3(d) is the result of VFCCV which converges to the correct boundary.

<sup>1</sup>Cropped from a ultrasonic heart phantom image provided by Dr. Harold Becher of Echocardiography Labora-

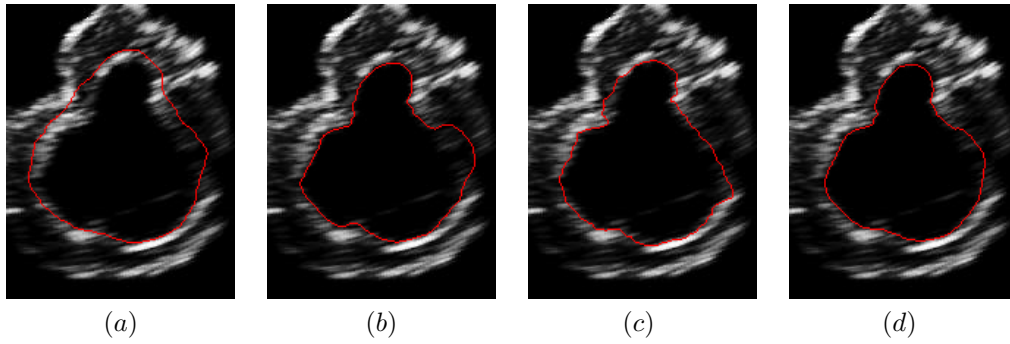


Figure 5.2: Comparison on left ventricle of ultrasound image (a) VFC (b) Chan-Vese snake (c)RAGS and (d) VFCCV

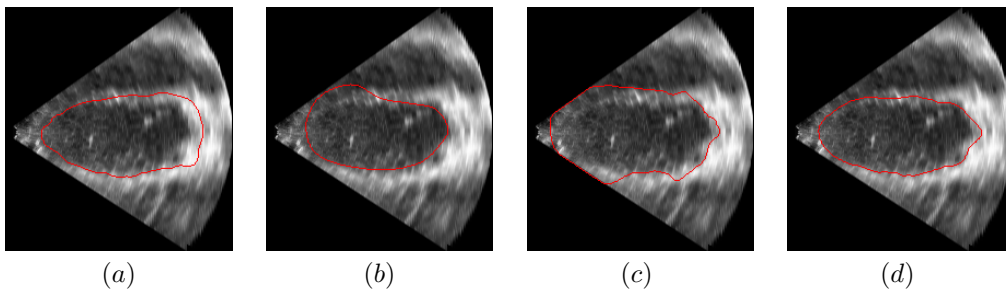


Figure 5.3: Comparison on human echocardiography (a) VFC (b) Chan-Vese snake (c) RAGS and (d) VFCCV

We have also done the experiments using another heart phantom ultrasound images. The detailed description of the heart phantom will be introduced in the next section. The comparative results are shown in Fig. 5.4. The initialization for the four cases is a contour slightly larger than the real boundary. Similar to the previous comparative results, the performance of VFC, Chan-Vese snake and RAGS are worse than that of VFCCV.

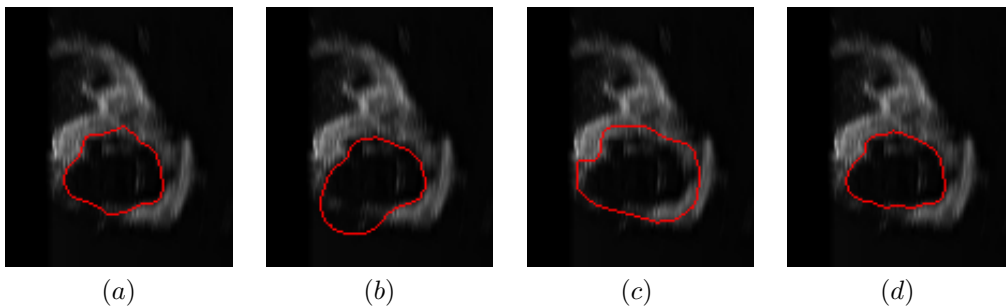


Figure 5.4: Comparison on heart phantom images on ultrasound (a) VFC (b) Chan-Vese snake (c) RAGS and (d) VFCCV

Finally, we have compared our VFCCV snake model with the competing algorithms in

an experiment on both left and right ventricles of an MRI image. The results of the various competing algorithms on one MRI image are shown in Fig. 5.5. The red and green curves are two separate independently evolved active contours but shown in the same figure to save space. The image is taken from MICCAI 2012 Right Ventricle Challenge (RVSC) [56]. The manual initializations of both the left ventricle (green contour on the right side) and the right ventricle (red contour on the left side) are slightly larger than the true boundaries. Fig. 5.5(a) is the result of using VFC where the contour is attracted incorrectly to the nearby strong edges as the VFC snake is designed to do. Fig. 5.5(b) shows the result of the Chan-Vese snake. Both contours experience leakage problems because the Chan-Vese model ignores edge information. Fig. 5.5(c) is the result of RAGS. The right ventricle on the left side is distracted by nearby strong edges. The bottom-left of this ventricle region in the region segmentation map is merged with nearby muscle tissues that influence the segmentation result. The left ventricle on the right side leaks through the weak edge which is a direct result of the initial region segmentation. Fig. 5.5(d) is the result of our VFCCV which converges to the object boundary correctly.

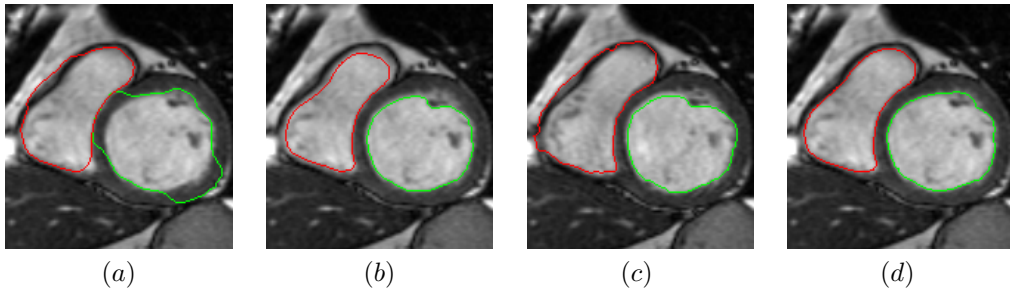


Figure 5.5: Comparison on left and right ventricles of MRI (a) VFC (b) Chan-Vese snake (c) RAGS and (d) VFCCV

Using the right ventricle dataset and the provided ground truth segmentation in RVSC [56], the quantitative comparison among different competing algorithms is made and the result is shown in Tab. 5.1 in terms of Pratt's Figure of Merit (FOM), Dice's coefficient (DC) and Hausdorff Distance (HD). In the case of FOM and DC, the larger the value, the better the performance of an algorithm is, whereas with HD, the smaller the value, the superior the algorithm. FOM is a common metric used for evaluating the accuracy of an edge detection algorithm whereas DC and HD are metrics used in the RVSC [56]. Both the averages of the three metrics, computed over a set of 21 images that have ground truth segmentation and their standard deviations are shown in Tab. 5.1. We can see that for all the three metrics, our method is superior to the competing methods.



	FOM	DC	HD
VFC	0.66±0.12	0.92±0.03	8.34±3.41
Chan-Vese Snake	0.63±0.10	0.89±0.05	9.02±3.61
RAGS	0.70±0.08	0.92±0.03	7.82±2.68
VFCCV	<b>0.79±0.06</b>	<b>0.95±0.02</b>	<b>5.11±1.31</b>

Table 5.1: Quantitative Comparison of the Competing Algorithms in Terms of Pratt’s Figure of Merit (FOM), Dice’s coefficient (DC), and Hausdorff Distance (HD). Note that the higher the FOM or DC value is, the better the performance, and that the smaller the HD value, the better the performance. Each metric value is an average computed over 21 images that have ground truth segmentation and the metric values are followed by the respective standard deviations.

## 5.2 Performance of the Propagation and Validation Algorithm

In this section, we will give the experimental results and analysis of the segmentation algorithm proposed in Chapter 4. The dataset we will mainly use in this chapter is comprised of the ultrasonic heart phantom images, which will be described first. Then, we will show the performance of VFCCV in this segmentation system when all the contours are judged as accurate by the validation step. However, the contours may be inaccurate in some cases. We will also show that the validation step can truly detect these inaccurate contours. Using the propagation and validation algorithm, we can obtain the final volumes of the 3D ventricle. The accuracy of those volumes compared with the ground truth is given. To verify the reproducibility of the segmentation system, we compute the intra-person variance and inter-person variance. The specificity and sensitivity, which are the two metrics to measure the performance of the system, are also calculated. Then, we will briefly introduce the interface we build. In Section 5.2.8, we will show the performance of the system on the human heart dataset.

### 5.2.1 Dataset

The dataset we will use in this part is the heart phantom dataset<sup>2</sup>. A heart phantom is an artificial mechanical heart that can mimic the performance of a real human heart. The shape of the heart phantom is similar to a human heart, and it also allows the assessment of ultrasound techniques [73]. The image quality of the heart phantom is higher than the quality of the human heart echocardiography, and thus it contains less noise. The reason that we choose to use the heart phantom dataset is that it comes with the ground truth regarding the volume measurement. The true volume of the phantom ventricle is measured by a physical method. The “ground truth” of human or animal ventricle can only be esti-

<sup>2</sup>The heart phantom dataset is provided by Dr. Harold Becher of Echocardiography Laboratory, Mazankowski Alberta Heart Institute

mated by physicians using clinical methods which are subjective and inaccurate. Using the heart phantom is a common practice in medical research [11].

The heart phantom dataset consists of five 4D images, as shown in Fig. 5.6. Each 4D image contains nine 3D images corresponding to the time sequence from the end-diastolic status to the end-systolic status, as shown in Fig. 5.7. Each 3D image contains a ventricle to be segmented, and each 3D image can be divided into a sequence of 2D slices. A typical sequence is shown in Fig. 5.8. Experiments are conducted on the five 4D images. The five 4D images are generated using the same heart phantom, and thus they share the same ground truth regarding the volume measurement. The training data for each 4D data is taken from the first 3D image in the time sequence. We manually check that all the contours in the training set are accurate. Most of the contours in the training set are generated automatically, and the inaccurate ones are redrawn manually.

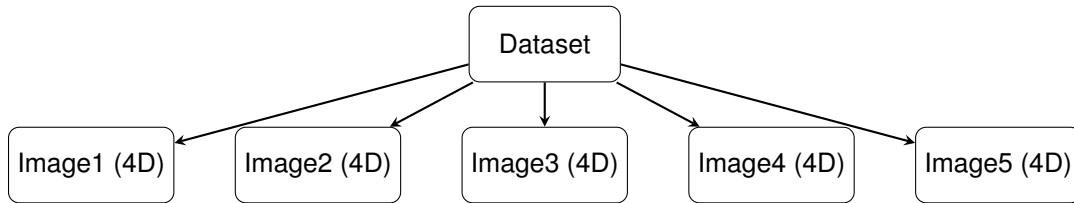


Figure 5.6: Our dataset consists of five 4D images

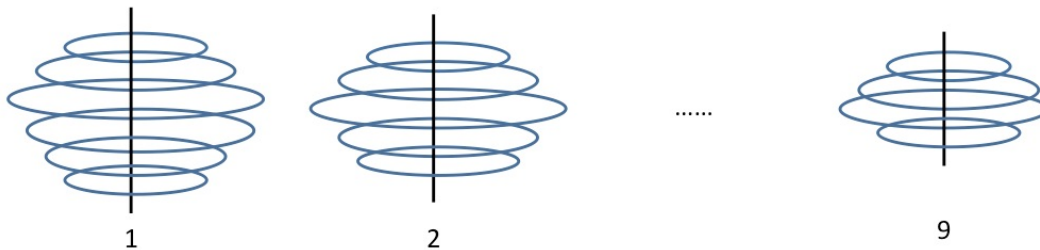


Figure 5.7: One 4D image consists of nine 3D images indexed from two to ten

### 5.2.2 Performance of VFCCV in the Segmentation System

Using the heart phantom ultrasound images, we first show in Fig. 5.8 a segmented 3D ventricle which is displayed as a sequence of 2D slices. In this sequence, all the contours are judged as accurate by the validation step. Here all the slices are from one 3D image in one heart cycle. Since all the contours are accepted by the validation, besides the initial manual initialization of the middle slice, the remaining segmentation is fully-automatic. We can see that the segmentation model can detect the boundary accurately. Also, the validation step can accept the accurate segmentation results correctly.

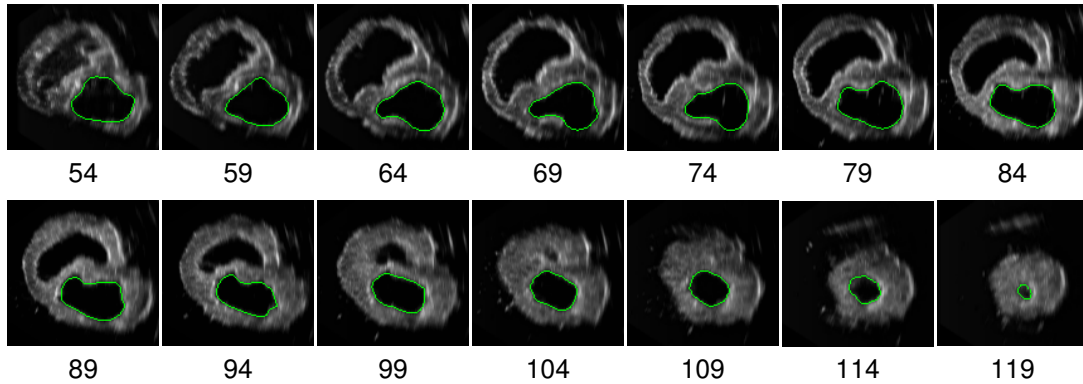


Figure 5.8: Segmentation results on the heart phantom using VFCCV when all the contours are accepted by the validation step. Every five slice from the 54th slice to the 119th slice

For most of the 2D slices, the VFCCV model can generate accurate segmentation results. However, some slices are difficult for VFCCV to handle. Some examples of wrong cases are shown in the next section, which can be detected by the validation step.

### 5.2.3 Inaccurate Contours Detected by the Validation Step

Fig. 5.9 shows several typical cases when the contours are inaccurate, and the validation is able to detect the errors. The red curves are the segments that fail the validation which are truly wrong, for example, when the curve does not fully evolve to the true boundary, or it goes across the heart tissues. A user can manually reinitialize the contour as shown in the images labeled *After*. As mentioned in Section 4.2.2, we can either reinitialize the whole contour, or just reinitialize the failed segment on the contour. Note that after reinitialization, we smooth and refine the contour by running only a few iterations of VFCCV to prevent it from returning to the same wrong boundary as before. In the experiments, the reconstruction error threshold for acceptance decision is chosen experimentally. All the slices in one 4D image use the same threshold.

Generally, the validation step can detect all the inaccurate contours. However, some accurate contours may also be judged as inaccurate. This would lead to some extra manual work from the user, e.g., to manually accept the current results. The evaluation of this part is shown in Section 5.2.6.

### 5.2.4 Accuracy of the Final Volume

With the propagation and validation algorithm, we can segment all the slices in one 3D image to obtain the final volume. The original volume in our segmentation system is in voxels. To transform it to ml, we use one of the five 4D images in turn to calculate a constant conversion scaler. Then, we use this constant to scale the other four images.

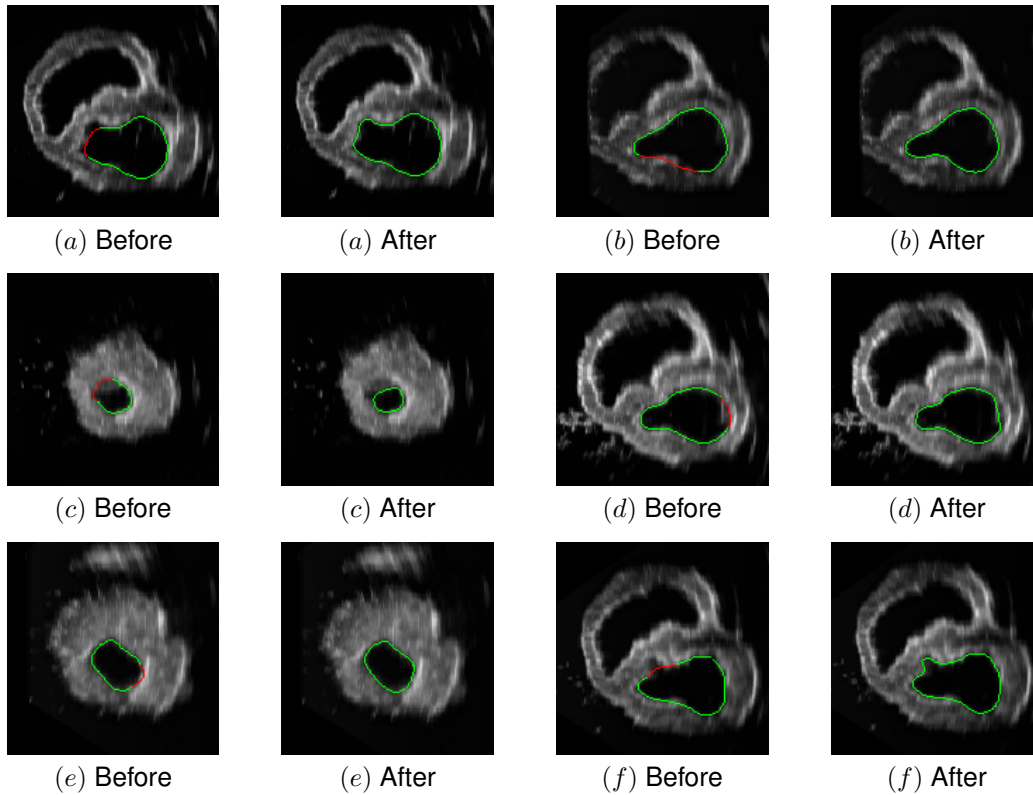


Figure 5.9: Before: the red curves show the inaccurate curves detected by the validation step. After: the contours after user correction

Based on each constant scaler, we can get the results of the five 4D images, and each one contains nine volumes.

Tab. 5.2 shows the accuracy of our method. The number in bold in each row corresponds to the 4D image that is used to calculate the constant scaler. Each column shows the accuracy of the five 4D images based on the corresponding constant scaler.  $a_{xy}$  means the accuracy of the  $x$ th 4D image using the constant scaler calculated using the  $y$ th 4D image,  $x = 1, \dots, 5$ ,  $y = 1, \dots, 5$ . Denote  $v_{ij}$  as the volume of the  $j$ th 3D image in the  $i$ th 4D image,  $i = 1, \dots, 5$ ,  $j = 1, \dots, 9$ . Each value in row  $i$ ,  $i = 1, \dots, 5$ , is the average accuracy of  $v_{ij}$ ,  $j = 1, \dots, 9$ . The second column shows the accuracy of the five 4D images based on the constant scaler calculated using the first 4D image. The highest accuracy is 98.9% which is obtained from the first 4D image, and the lowest accuracy is 97.7% which is obtained from the fourth 4D image. To remove the bias using the scaler calculated using the first 4D image, we also use the scalers calculated from the other four 4D images. The results are shown in the remaining columns. We can see that no matter which image is used for calculating the constant scaler, our method achieves a high accuracy for all the five images.

Fig. 5.10(a) shows the volume curves. The blue curve is the ground truth, and the red

Image	Accuracy based on the five constant scalers respectively				
1	$a_{11}$ = <b>98.9%</b>	$a_{12}$ =98.3%	$a_{13}$ =98.1%	$a_{14}$ =97.7%	$a_{15}$ =98.8%
2	$a_{21}$ =98.0%	$a_{22}$ = <b>98.5%</b>	$a_{23}$ =96.4%	$a_{24}$ =96.0%	$a_{25}$ =98.4%
3	$a_{31}$ =98.1%	$a_{32}$ =96.5%	$a_{33}$ = <b>99.3%</b>	$a_{34}$ =99.2%	$a_{35}$ =97.2%
4	$a_{41}$ =97.7%	$a_{42}$ =96.1%	$a_{43}$ =98.6%	$a_{44}$ = <b>98.5%</b>	$a_{45}$ =96.9%
5	$a_{51}$ =98.8%	$a_{52}$ =98.9%	$a_{53}$ =97.1%	$a_{54}$ =96.7%	$a_{55}$ = <b>99.1%</b>

Table 5.2: Average accuracy of the volume for the five 4D images based on the five constant scalers respectively

curves are the results for the five images respectively. Each curve is composed of nine points. The red curves are scaled based on the constant scaler calculated using the first image data. Note that different constant scalers only influence the shift in  $y$ -axis of the five curves instead of their shapes. Fig. 5.10(a) is the one with the minimum cumulative error among the five constant scalers. Fig. 5.10(b) is the one with the maximum cumulative error, which is based on the constant scaler calculated from the fourth image. We can see that in both Fig. 5.10(a) and Fig. 5.10(b), all the red curves have similar shapes and locations with the ground truth.

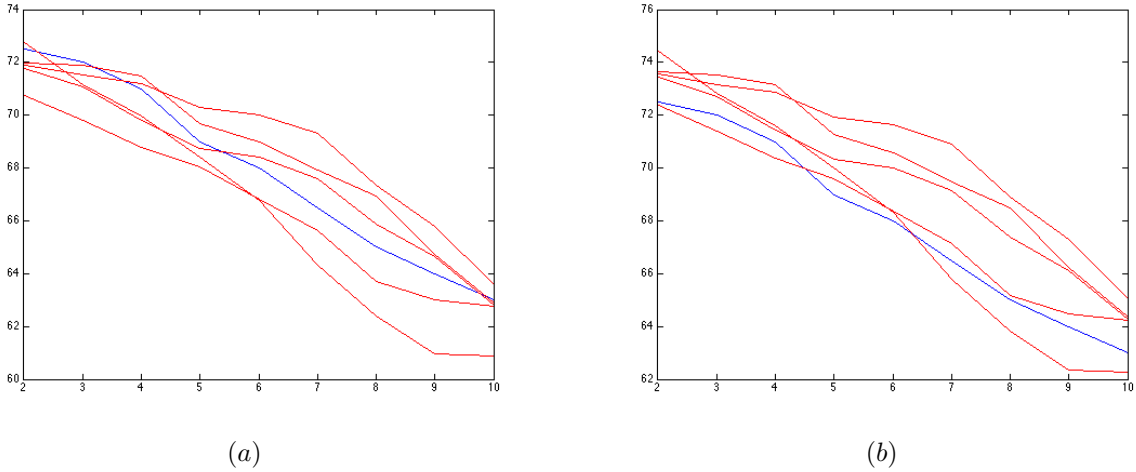


Figure 5.10: The red curves show the volumes of the five image data respectively. The blue curve is the ground truth. The x-axis is the time sequence, and the y-axis shows the corresponding volumes. (a) The volumes are scaled based on the constant calculated using the first image. It obtains the minimum cumulative error among the five constants. (b) The scaler is calculated using the fourth image, which obtains the maximum cumulative error

To obtain the volumes from the end-diastolic status to the end-systolic status, we can also only segment the two extreme 3D images, i.e., the end-systolic status and end-diastolic status, and estimate the intermediate volumes by linear interpolation. However, this may lead to a relatively low average accuracy.

### 5.2.5 Reproducibility: Intra-Person Variability and Inter-Person Variability

In this section, we will investigate the reproducibility of our segmentation system. Higher reproducibility, e.g, lower variability, means higher stability of the system. The reproducibility can be measured by the coefficient of variability (CV, defined as the ratio of the standard deviation to the mean), the Bland-Altman plot (BA, defined as the mean of the differences between two sets of data  $\pm 2 \times$  standard deviation of the differences) [44], the percentage difference of the mean (MD, defined as the difference between two means divided by the average of the two means) [45], etc. [73].

We have two users. Denote the first user's data as  $v_{ij}^1$ , the second one's as  $v_{ij}^2$ ,  $i = 1, \dots, 5$ ,  $j = 1, \dots, 9$ . Here  $i$  indicates the index of the 4D image, and  $j$  indicates the index of the 3D image in the 4D image. Since BA and MD are normally performed on two sets of data, they are better for measuring the inter-person variability than the intra-person variability in our case. So we use CV to measure the intra-person variability and the other two to measure inter-person variability.

As mentioned before, the five 4D images share the same ground truth. So the first user's data on the five 4D images can be used to measure CV. That is, we use  $v_{ij}^1$ ,  $i = 1, \dots, 5$ ,  $j = 1, \dots, 9$  to measure CV. The nine CVs, corresponding to the nine time points from the end-diastolic status to the end-systolic status, are calculated based on  $v_{i1}^1, v_{i2}^1, \dots, v_{i9}^1$  respectively,  $i = 1, \dots, 5$ . We can obtain the CV through  $CV_t = std(v_{it}^1)/mean(v_{it}^1)$ ,  $i = 1, \dots, 5$ ,  $t = 1, \dots, 9$ . The CVs of the nine volumes are shown in Tab. 5.3. The average CV is 2%. We obtain a smaller CV compared with other common clinical results [73]. Note that CV is irrelevant to the constant scaler transforming from voxel to ml that we have had to deal with in the previous subsection.

3D image	1	2	3	4	5	6	7	8	9
CV	1.0%	1.1%	1.6%	1.4%	2.1%	3.0%	3.3%	3.0%	1.6%

Table 5.3: Coefficient of variability (CV) of the nine 3D images

The BA is calculated based on the 45 pairs of data composed of five 4D images with nine volumes in each. That is, the difference  $d_{ij} = v_{ij}^1 - v_{ij}^2$ , and we have 45 differences in total. We can obtain the BA through  $BA = mean(d_{ij}) \pm 2std(d_{ij})$ . The BA of the two users' data is  $39.7 \pm 1988$  voxel. The BA plot is shown in Fig. 5.11. Transforming from voxel to ml using the constant scaler calculated using the first image of the first user's results, we can get a BA of  $0.04 \pm 2.02$  ml, which is narrower than the clinical results using the same dataset [73].

In the calculation of MD, the nine means of the first user's are calculated using  $v_{i1}^1, v_{i2}^1, \dots, v_{i9}^1$  respectively,  $i = 1, \dots, 5$ , and the nine means of the second user's are calculated us-

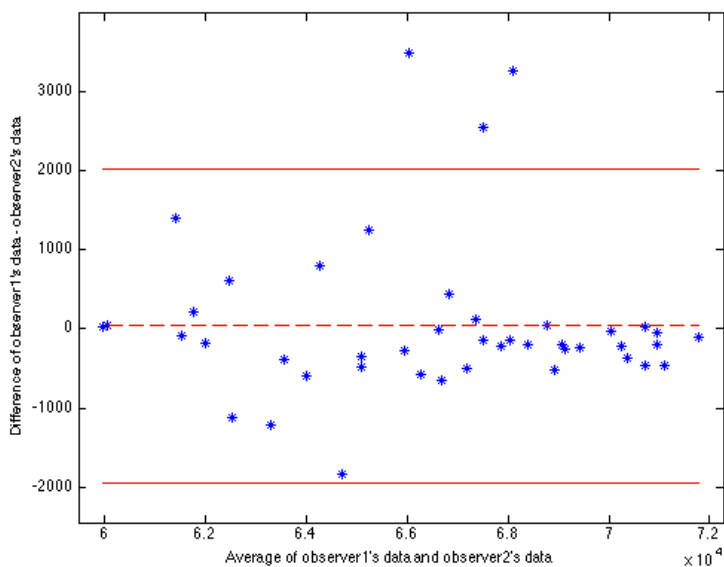


Figure 5.11: Bland-Altman Plot

ing  $v_{i1}^2, v_{i2}^2, \dots, v_{i9}^2$  respectively,  $i = 1, \dots, 5$ . We have  $MD_t = \frac{|\text{mean}(v_{it}^1) - \text{mean}(v_{it}^2)|}{(\text{mean}(v_{it}^1) + \text{mean}(v_{it}^2))/2}$ ,  $i = 1, \dots, 5$ ,  $t = 1, \dots, 9$ . The MDs between the two users' data of the nine 3D ventricles respectively are shown in Tab. 5.4. The MD ranges from zero to one, and the smaller the MD, the better the performance. We obtain a very small MD. We also obtain a smaller MD compared with other common clinical results [73]. Note that the MD is also irrelevant to the constant scaler.

3D image	1	2	3	4	5	6	7	8	9
MD	0.76%	0.47%	0.73%	0.12%	0.13%	0.09%	0.32%	0.46%	0.74%

Table 5.4: Percentage of difference of mean (MD) of the nine 3D images

The above intra-person variability and inter-person variability represent that our segmentation system has high reproducibility and high stability.

## 5.2.6 Specificity and Sensitivity

In our segmentation system, a contour either passes or fails the validation. In all the contours that pass the validation, they can be either accurate or inaccurate. In all the contours that fail the validation, they can also be truly inaccurate or accurate in fact. This is a classical binary classification problem, and the performance of our validation system can therefore be measured in terms of the common metrics of specificity and sensitivity. In this section, we will measure the performance of the system by investigating among all the inaccurate contours, the percentage of contours that are detected by the validation, i.e., the sensitivity,

and among all the accurate contours, the percentage of contours that pass the validation, i.e., the specificity.

Tab. 5.5 shows the average number of automatic acceptances, manual acceptances, manual corrections, and inaccurate contours passing the validation of the five 4D images. Here, the automatic acceptance indicates the accurate contours passing the validation. Manual acceptance indicates the contours that fail the validation, but in fact they are accurate, so we need manually accept the contours. Manual correction indicates that the contour is detected by the validation, and it is truly inaccurate, and thus we need to manually correct the result. A few slices in the five 4D images are so unclear that it is difficult to define where the true boundary is. For these few slices, we judge the segmentation as accurate if it is acceptable, and judge it as inaccurate if it is obviously wrong. So we use an approximately equal sign in Tab. 5.5 instead of an exactly equal sign. These numbers are obtained by a trained operator with all the parameters carefully tuned.

average # of auto acceptances	average # of manual acceptances	average # of manual corrections	average # of inaccurate contours passing the validation
≈450	≈15	≈48	≈0

Table 5.5: Average number of auto acceptances, manual acceptances, manual corrections, and inaccurate contours passing the validation of the five 4D images

In our case, a true positive (TP) is defined as an inaccurate segmentation that is detected. A false positive (FP) is defined as an accurate segmentation that is mistakenly detected. A true negative (TN) is defined as an accurate segmentation that pass the validation. A false negative (FN) is defined as an inaccurate segmentation that passes the validation. The sensitivity is defined as  $Sensitivity = TP / (TP + FN)$ , which indicates among all the inaccurate segmentations, how many of them are detected. The specificity is defined as  $Specificity = TN / (TN + FP)$ , which indicates among all the accurate segmentations, how many of them are accepted by the validation. For both the sensitivity and specificity, the higher they are, the better the performance. The sensitivity and specificity for the five 4D images are shown in the Tab. 5.6. As mentioned above, we can only say the average sensitivity and specificity are approximately equal to some values because the true boundary is difficult to define in a few unclear slices. A sensitivity of almost 1 indicates that the PCA validation step can detect almost all the inaccurate segmentations. A user has the chance to correct those correctly detected inaccurate segmentations to obtain an accurate final volume. The average specificity is approximately equal to 0.965. Without the PCA validation step, we need to manually accept or revise the segmentation result in every slice. However, with the validation step, we only need to accept or revise the contours that are detected to be inaccurate, that is, those that fail the PCA validation. The 0.965 specificity



indicates that among all the accurate segmentations, about 96.5% of them can be passed automatically, which represents a significantly saving compared with manual validation.

Image	Sensitivity	Specificity
1	$\approx 1$	$\approx 0.998$
2	$\approx 1$	$\approx 0.95$
3	$\approx 1$	$\approx 0.974$
4	$\approx 1$	$\approx 0.994$
5	$\approx 1$	$\approx 0.911$
Avg	$\approx 1$	$\approx 0.965$

Table 5.6: Sensitivity and specificity of the five image data and their average

As discussed in Chapter 4, we choose three slices in the training set for each new slice. If we choose only one slice, that is, the corresponding mapped slice, the number of manual acceptances will be significantly increased resulting in a low specificity. This is because the amount of the training data is too small to cover the features of the new pattern images. The top PCs are not so significant as when the training set is large and thus not representative enough. We have also conducted experiments using five or more neighbouring slices, and the results are similar with using just three slices. So the number of slices for PCA is chosen as three in our experiments.

### 5.2.7 User Interface

In this section, we will briefly introduce the user interface we built for the above system. The interface can be used in practice for the convenient operation. The interface is shown in Fig. 5.12. A user can open a 4D image and select the 3D image to be segmented. Then, a user can select to segment along either  $x$  or  $y$  or  $z$  direction of the 3D image. The top, bottom, and middle slice number indicate the range of slices to segment and the slice to initialize. These can be left empty, and a user can stop the propagation anytime by pressing key  $s$ . The scale is the calibration constant transforming a volume measurement from voxels to ml. The system also has a default scaler. After the image data is loaded, a user can zoom-in or zoom-out the image and change the position of the display.

The segmentation and propagation continue if the segmentation passes the PCA validation. If it does not pass, a user can either correct it or accept it. When a segmentation needs to be corrected, a user can either press key  $r$  to reinitialize the whole contour or press key  $d$  to just reinitialize the failed segments detected by the validation. The interface when a contour fails the validation is shown in Fig. 5.13.

The contour first propagates from the middle slice to the top slice, and then from the middle slice to the bottom slice. After one 3D image is segmented, the volume is shown in the Final Volume field. After all the 3D images are segmented, a user can store these

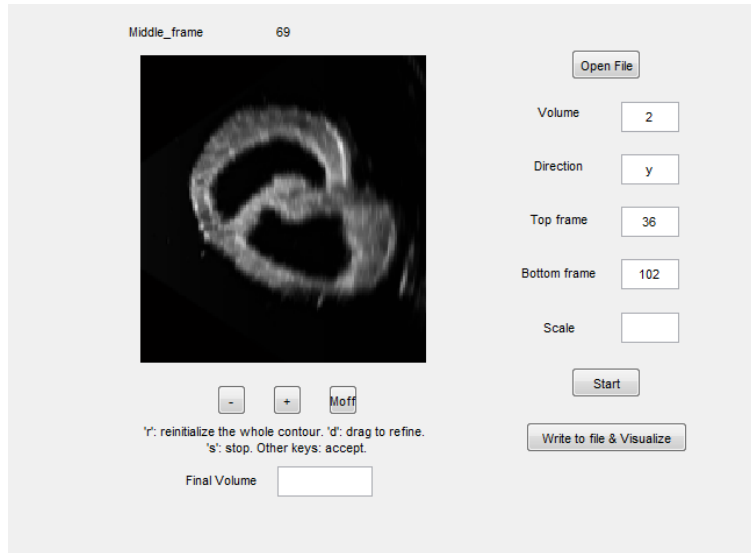


Figure 5.12: Interface

numbers to a file and visualize those 3D ventricles. A typical 3D visualization result is shown in Fig. 5.14.

### 5.2.8 Performance on Human Echocardiography

Besides the heart phantom dataset, we have also evaluated our system on a human echocardiography dataset. The human ultrasonic heart images are less clear than the ultrasonic heart phantom images. For many unclear slices, it is difficult to define where the true boundary is even for an expert. Typical segmentation results are shown in Fig. 5.15. The first row shows the segmentation results along the long-axis direction of the 3D heart, and the second row shows the results along the short-axis direction. The contour can trace the ventricle boundary in most of the cases. Also, the validation step can detect most of the inaccurate segmentations. Typical results are shown in Fig. 5.16. For example, the validation step can detect the cases when the contour does not fully evolve to the true boundary, or it goes across the tissue structure. Fig. 5.16 *Before* shows the cases detected by the validation step. Fig. 5.16 *After* shows the segmentation results after the user correction. The training data is taken from the first 3D image in the heart cycle, and the number of slices in each training set is also chosen as three. The parameters used for the human echocardiography dataset are different from the parameters used for the heart phantom dataset. We also measure the sensitivity and specificity of the system using the human echocardiography dataset. Due to the lack of clinical expertise, these numbers can only be roughly estimated. The average sensitivity is about 0.82 and the average specificity is about 0.92. Both of the two numbers are lower than the numbers obtained using the heart phantom dataset. This is because some of the images are noisy, and the gradient inside

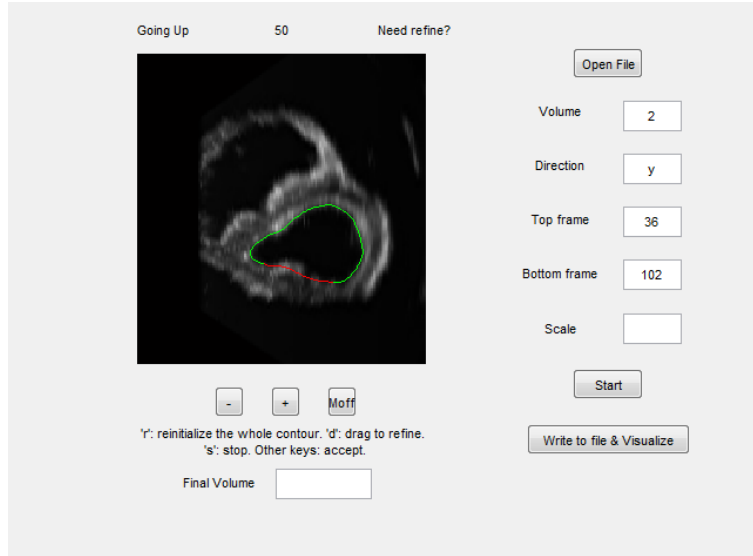


Figure 5.13: Interface when the contour fails the validation

the true boundary may be as bright as the gradient outside the contour. So the PCs may cover the feature of a non-ideal boundary, and thus the validation is not sensitive enough to detect the inaccurate contour that goes across the true boundary, which leads to a low sensitivity. On the other hand, the validation can still detect most of the contours with both bright gradient inside and outside. However, some of the bright gradient inside the contour is noise, and the contour should pass the validation, which leads to a low specificity.

### 5.3 Summary

In this chapter, we first give the performance of the VFCCV model. We have shown the experimental results on synthetic images, heart phantom ultrasound images, real heart ultrasound images and MRI images. Experimental results show that the VFCCV model not only has a large capture range and the ability to converge into concavities, but is also robust to noise, cluttered image background, and weak boundary. The results demonstrate that our method is effective and superior to VFC snake, Chan-Vese snake model, and another state-of-the-art method combining the edge and region information.

We then show the performance of the segmentation system. We use heart phantom ultrasound images and human heart ultrasound images to conduct the experiments. The segmentation system can generate correct segmentation results in most of the cases, and it can detect the errors when the contours are inaccurate. We also demonstrate that the segmentation system significantly saves manual work. For the heart phantom dataset, the final volumes have high accuracy compared with the ground truth, and our system has high reproducibility. A user interface is also built for the convenient operation in practice.

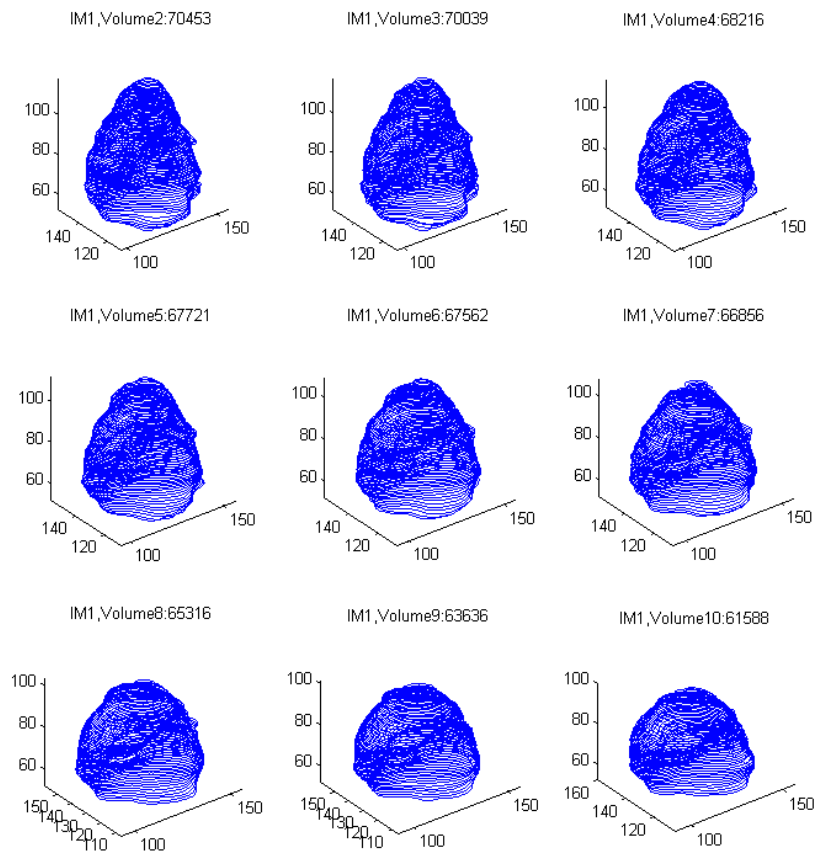


Figure 5.14: A typical 3D visualization result

The experiments on human heart echocardiography are presented in the end, which also demonstrate the effectiveness of our method.

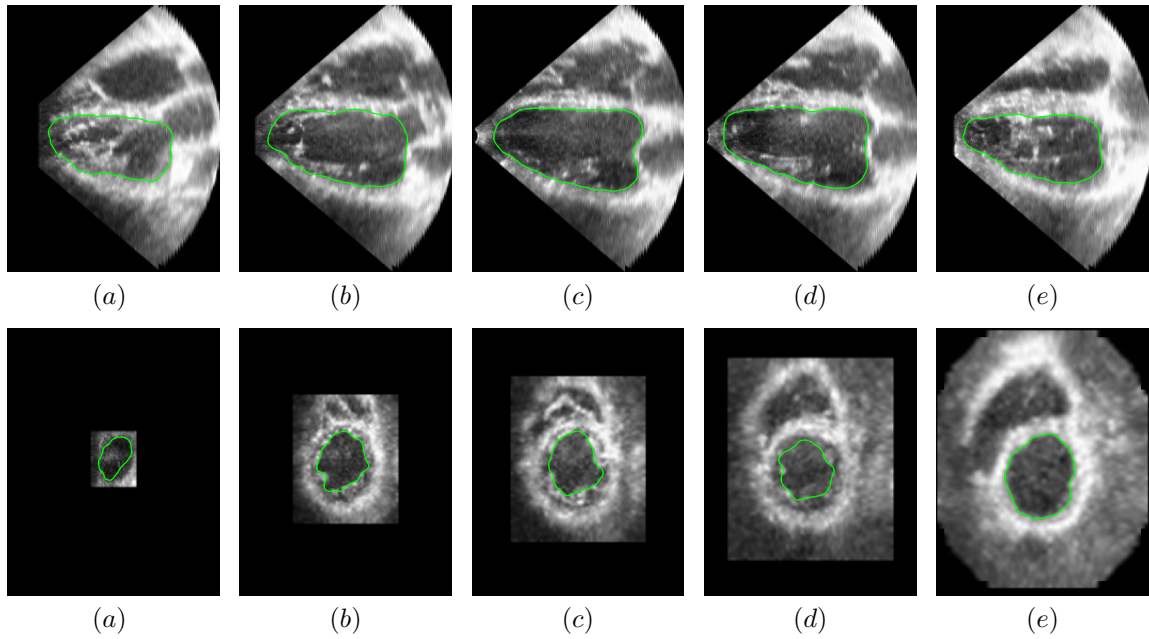


Figure 5.15: Segmentation results on human ultrasonic heart images. The first row: along long-axis direction. The second row: along short-axis direction

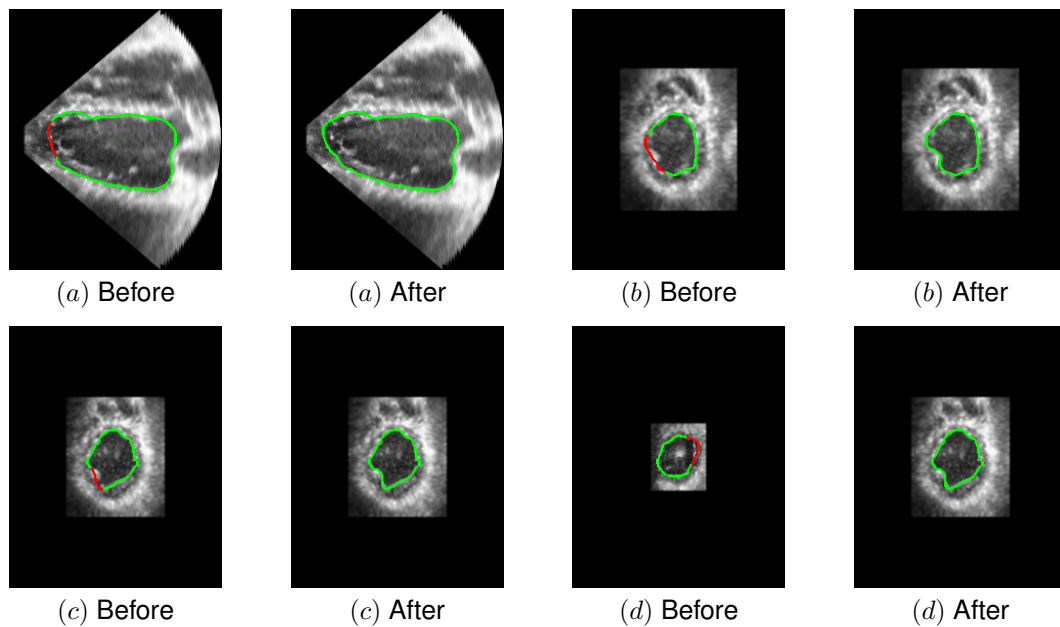


Figure 5.16: Before: the red curves show the inaccurate curves detected by the validation step. After: the contour after user correction

## Chapter 6

# Conclusion and future work

### 6.1 Conclusion

In this thesis, we first propose an active contour model called VFCCV. It considers both the edge and regional information. The VFCCV not only has large capture range and the ability to detect concavity, but is also robust to noise, cluttered image background, and weak boundary. Experiments on synthetic images, heart ultrasound and MRI images show VFCCV has better performance over either the original VFC model or the Chan-Vese model, and the RAGS model combining edge and region information.

The VFCCV is further used to segment the 2D slice sequence in the 3D ultrasound image in order to segment the 3D ventricle. We propose to embed an automatic validation step in the segmentation system. The validation step can judge whether the contour on each slice is accurate in order to make the whole process as automatic as possible while keeping a high segmentation accuracy. The validation is based on the PCA. We extend the existing method by constructing local and overlap pattern images in order to detect local errors. Also, we propose to build the training data in a location-specific manner. We mainly use the heart phantom ultrasound images as the dataset in this part for it comes with the ground truth regarding the volume measurement. The VFCCV can generate correct segmentation results in most of the cases, and the validation step can detect the errors when the contours are inaccurate. The final ventricle volumes have high accuracy. Experiments also show that the segmentation system has high reproducibility, and it is significantly faster than manual validation. A user interface is built for the convenient operation in practice. Experiments are also done on human echocardiography, which also demonstrates the effectiveness of our method.

## 6.2 Future work

In the propagation and validation section, now we segment the ventricle along either x or y or z direction of the 3D image. Future work includes to segment the ventricle along the longest axis of the heart. This will lead to a more accurate ventricle volume. It is because we can get the maximum number of slices if we divide the ventricle along its longest axis, and thus the error caused by selecting the range of the slices to segment would be minimized. Also, normally the apex few slices become unclear. Dividing along the longest direction reduces the number of unclear slices to the smallest, and thus the influence of the unclear slices is minimized.

Future work also includes the improvement of the validation on human ultrasonic heart images, which would be more useful for the clinical application. Since the quality of the human ultrasonic heart image is worse than the quality of the phantom ultrasonic heart image, the sensitivity and specificity on the human heart dataset we get are relatively low. To improve the performance may need the validation step to be more intelligent, for example, take the prior information given by the physicians and scientists into consideration.

# Bibliography

- [1] Georges B Aboutanos and Benoit M Dawant. Automatic brain segmentation and validation: image-based versus atlas-based deformable models. In *Medical Imaging 1997*, pages 299–310. International Society for Optics and Photonics, 1997.
- [2] Luis Alvarez, Frédéric Guichard, Pierre-Louis Lions, and Jean-Michel Morel. Axioms and fundamental equations of image processing. *Archive for rational mechanics and analysis*, 123(3):199–257, 1993.
- [3] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 112(9):855–867, 1990.
- [4] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, December 1993.
- [5] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997.
- [6] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.
- [7] Laurent D. Cohen. On active contour models and balloons. *CVGIP: Image Underst.*, 53(2):211–218, March 1991.
- [8] Laurent D. Cohen and Isaac Cohen. Finite element methods for active contour models and balloons for 2d and 3d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1131–1147, 1991.
- [9] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1197–1203. IEEE, 1999.
- [10] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti. Left ventricular volume estimation for real-time three-dimensional echocardiography. *IEEE Transactions on Medical Imaging*, 21:1202 – 1208, 2002.
- [11] Cristiana Corsi, Giuseppe Saracino, Alessandro Sarti, and Claudio Lamberti. Left ventricular volume estimation for real-time three-dimensional echocardiography. *Medical Imaging, IEEE Transactions on*, 21(9):1202–1208, 2002.
- [12] R Couant and D Hilbert. Methods of mathematical physics. *Interscience, New York*, 1, 1953.



- [13] Jephtha P Curtis, Seth I Sokol, Yongfei Wang, Saif S Rathore, Dennis T Ko, Farid Jadbabaie, Edward L Portnay, Stephen J Marshalko, Martha J Radford, and Harlan M Krumholz. The association of left ventricular ejection fraction, mortality, and cause of death in stable outpatients with heart failure. *Journal of the American College of Cardiology*, 42(4):736–742, 2003.
- [14] Chris A Davatzikos and Jerry L Prince. An active contour model for mapping the cortex. *Medical Imaging, IEEE Transactions on*, 14(1):65–80, 1995.
- [15] Hervé Delingette. Simplex meshes: a general representation for 3D shape reconstruction. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 856–859. IEEE, 1994.
- [16] Yining Deng and BS Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):800–810, 2001.
- [17] Edsger. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [18] Gang Dong, Nilanjan Ray, and Scott T Acton. Intravital leukocyte detection using the gradient inverse coefficient of variation. *Medical Imaging, IEEE Transactions on*, 24(7):910–924, 2005.
- [19] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
- [20] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.
- [21] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.
- [22] Matthew A Grayson. Shortening embedded curves. *Annals of Mathematics*, pages 71–111, 1989.
- [23] W. Eric L. Grimson, GJ Ettinger, T Kapur, Michael E. Leventon, William M Wells III, and Ron Kikinis. Utilizing segmented mri data in image-guided surgery. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(08):1367–1397, 1997.
- [24] Lawrence O Hall, Amine M Bensaid, Laurence P Clarke, Robert P Velthuizen, Martin S Silbiger, and James C Bezdek. A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *Neural Networks, IEEE Transactions on*, 3(5):672–682, 1992.
- [25] Larry Hatcher. *A step-by-step approach to using the SAS system for factor analysis and structural equation modeling*. Sas Institute, 1994.
- [26] Jason Hsu. Principal components analysis. *For use of UCLA Anderson MFE Program*.
- [27] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [28] Vincent S Khoo, David P Dearnaley, David J Finnigan, Anwar Padhani, Steven F Tanner, and Martin O Leach. Magnetic resonance imaging (mri): considerations and applications in radiotherapy treatment planning. *Radiotherapy and Oncology*, 42(1):1–15, 1997.

- [29] Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum, and Anthony Yezzi Jr. Conformal curvature flows: from phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134(3):275–301, 1996.
- [30] Benjamin B. Kimia, Allen R. Tannenbaum, and Steven W. Zucker. Shapes, shocks, and deformations i: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1994.
- [31] Ron Kimmel, Arnon Amir, and Alfred M. Bruckstein. Finding shortest paths on surfaces using level sets propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(6):635–640, 1995.
- [32] Igor Klem, Dipan J Shah, Richard D White, Dudley J Pennell, Albert C van Rossum, Matthias Regenfus, Udo Sechtem, Paulo R Schwartzman, Peter Hunold, Pierre Croisille, et al. Prognostic value of routine cardiac magnetic resonance assessment of left ventricular ejection fraction and myocardial damage an international, multicenter study. *Circulation: Cardiovascular Imaging*, 4(6):610–619, 2011.
- [33] RM Lang, M Bierig, RB Devereux, FA Flachskampf, E Foster, PA Pellikka, MH Picard, MJ Roman, J Seward, JS Shanewise, et al. Recommendations for chamber quantification. *J Am Soc Echocardiogr*, 18(12):1440–1463, 2005.
- [34] Roberto M Lang, Michelle Bierig, Richard B Devereux, Frank A Flachskampf, Elyse Foster, Patricia A Pellikka, Michael H Picard, Mary J Roman, James Seward, Jack Shanewise, et al. Recommendations for chamber quantification. *European Journal of Echocardiography*, 7(2):79–108, 2006.
- [35] Stephen M Lawrie and Suheib S Abukmeil. Brain abnormality in schizophrenia. a systematic and quantitative review of volumetric magnetic resonance imaging studies. *The British Journal of Psychiatry*, 172(2):110–120, 1998.
- [36] Bing Li and Scott T Acton. Active contour external force using vector field convolution for image segmentation. *Image Processing, IEEE Transactions on*, 16(8):2096–2106, 2007.
- [37] Chunming Li, Chiu-Yen Kao, John C Gore, and Zhaohua Ding. Implicit active contours driven by local binary fitting energy. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.
- [38] Chunming Li, Chiu-Yen Kao, John C Gore, and Zhaohua Ding. Minimization of region-scalable fitting energy for image segmentation. *Image Processing, IEEE Transactions on*, 17(10):1940–1949, 2008.
- [39] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Distance regularized level set evolution and its application to image segmentation. *Image Processing, IEEE Transactions on*, 19(12):3243–3254, 2010.
- [40] P Lipson, Alan L Yuille, D O’Keeffe, J Cavanaugh, J Taaffe, and D Rosenthal. Deformable templates for feature extraction from medical images. In *Computer Vision, ECCV 90*, pages 413–417. Springer, 1990.
- [41] David MacDonald, David Avis, and Alan C Evans. Multiple surface identification and matching in magnetic resonance images. In *Visualization in Biomedical Computing 1994*, pages 160–169. International Society for Optics and Photonics, 1994.
- [42] Sara Mahdavi, Orcun Goksel, and Septimiu E. Salcudean. 3D Prostate Segmentation in Ultrasound Images Based on Tapered and Deformed Ellipsoids. In *Medical Image Computation and Computer Assisted Intervention (MICCAI)*, pages 960–967, 2009.

- [43] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:158–175, 1995.
- [44] J Martin Bland and DouglasG Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The lancet*, 327(8476):307–310, 1986.
- [45] mathsisfun. Percentage difference, 2013. <http://www.mathsisfun.com/percentage-difference.html>.
- [46] Tim Mcinerney and Demetri Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1:91–108, 1996.
- [47] Max Mignotte, Jean Meunier, and Jean-Claude Tardif. Endocardial boundary estimation and tracking in echocardiographic images using deformable template and markov random fields. *Pattern Analysis & Applications*, 4(4):256–271, 2001.
- [48] Steven C. Mitchell, Johan G. Bosch, Boudewijn P. F. Lelieveldt, Rob J. van der Geest, Johan H. C. Reiber, and Milan Sonka. Active appearance models: segmentation of cardiac MR and Ultrasound images. *IEEE Transactions on Medical Imaging*, 21:1167–1178, 2002.
- [49] Steven C Mitchell, Boudewijn PF Lelieveldt, Rob J van der Geest, Hans G Bosch, JHC Reiver, and Milan Sonka. Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac mr images. *Medical Imaging, IEEE Transactions on*, 20(5):415–423, 2001.
- [50] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 191–198, New York, NY, USA, 1995. ACM.
- [51] Nils J Nilsson. *Principles of artificial intelligence*. Springer, 1982.
- [52] Eric N.Mortensen and William A.Barrett. Interactive segmentation with intelligent scissors. *Thesis, Brigham Young University*.
- [53] J. Alison Noble and Djamel Boukerroui. Ultrasound image segmentation: a survey. *IEEE Trans. Med. Imaging*, 25(8):987–1010, 2006.
- [54] Richard Nock and Frank Nielsen. Statistical region merging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1452–1458, 2004.
- [55] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [56] Caroline Petitjean. RV Segmentation Challenge in Cardiac MRI, 2012. <http://www.litislabs.eu/rvsc>.
- [57] Dzung L. Pham, Chenyang Xu, and Jerry L. Prince. Current methods in medical image segmentation1. *Annual Review of Biomedical Engineering*, 2(1):315–337, 2000.
- [58] Nilanjan Ray and Scott T. Acton. Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE Trans. Med. Imaging*, 23(12):1466–1478, 2004.
- [59] Jadwiga Rogowska. Overview and fundamentals of medical image segmentation. *Handbook of Medical Imaging, Processing and Analysis*, pages 69–85, 2000.

- [60] Rmi Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.
- [61] Baidya Nath Saha, Nilanjan Ray, and Hong Zhang. Snake Validation: A PCA-Based Outlier Detection Method. *IEEE Signal Process. Lett.*, 16(6):549–552, 2009.
- [62] Guillermo Sapiro and Allen Tannenbaum. Affine invariant scale-space. *International journal of computer vision*, 11(1):25–44, 1993.
- [63] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science ... on Applied and Computational Mathematics*. Cambridge University Press, 2 edition, June 1999.
- [64] Kaleem Siddiqi, Yves Bérubé Lauziere, Allen Tannenbaum, and Steven W Zucker. Area and length minimizing flows for shape segmentation. *Image Processing, IEEE Transactions on*, 7(3):433–443, 1998.
- [65] Richard Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301, 1990.
- [66] P Taylor. Computer aids for decision-making in diagnostic radiologya literature review. *Computer*, 68(813), 1995.
- [67] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, August 1988.
- [68] Yun Tian, Fuqing Duan, Mingquan Zhou, and Zhongke Wu. Active contour model combining region and edge information. *Mach. Vis. Appl.*, 24(1):47–61, 2013.
- [69] Bram Van Ginneken, Alejandro F Frangi, Joes J Staal, Bart M ter Haar Romeny, and Max A Viergever. Active shape model segmentation with optimal features. *Medical Imaging, IEEE Transactions on*, 21(8):924–933, 2002.
- [70] Wikipedia. Divergence, November 2013. <http://en.wikipedia.org/wiki/Divergence>.
- [71] Wikipedia. Normal in geometry, November 2013. [http://en.wikipedia.org/wiki/Normal\\_\(geometry\)](http://en.wikipedia.org/wiki/Normal_(geometry)).
- [72] Wikipedia. Partial derivative, March 2014. [http://en.wikipedia.org/wiki/Partial\\_derivative](http://en.wikipedia.org/wiki/Partial_derivative).
- [73] Peter W. Wood. Left ventricular ejection fraction and volume calculations using three-dimensional echocardiography. Master's thesis, Department of Medicine, University of Alberta, Spring 2014.
- [74] Andrew J Worth, Nikos Makris, Verne S Caviness Jr, and David N Kennedy. Neuroanatomical segmentation in mri: technological objectives. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(08):1161–1187, 1997.
- [75] Xianghua Xie and Majid Mirmehdi. RAGS: Region-aided geometric snake. *IEEE Transactions on Image Processing*, 13(5):640–652, May 2004.
- [76] Chenyang Xu, Dzung L Pham, and Jerry L Prince. Image segmentation using deformable models. *Handbook of medical imaging*, 2:129–174, 2000.

- [77] Chenyang Xu and Jerry L. Prince. Gradient vector flow: A new external force for snakes. In *CVPR*, pages 66–71. IEEE Computer Society, 1997.
- [78] Chenyang Xu and Jerry L Prince. Generalized gradient vector flow external forces for active contours. *Signal processing*, 71(2):131–139, 1998.
- [79] Chenyang Xu, Anthony Yezzi, Jr., and Jerry L. Prince. On the relationship between parametric and geometric active contours. In *Proc. of 34th Asilomar Conference on Signals, Systems, and Computers*, pages 483–489, 2000.
- [80] Anthony Yezzi Jr, Satyanad Kichenassamy, Arun Kumar, Peter Olver, and Allen Tannenbaum. A geometric snake model for segmentation of medical imagery. *Medical Imaging, IEEE Transactions on*, 16(2):199–209, 1997.
- [81] Alan L Yuille, Peter W Hallinan, and David S Cohen. Feature extraction from faces using deformable templates. *International journal of computer vision*, 8(2):99–111, 1992.
- [82] Kaihua Zhang, Huihui Song, and Lei Zhang. Active contours driven by local image fitting energy. *Pattern recognition*, 43(4):1199–1206, 2010.
- [83] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9):884–900, September 1996.
- [84] Alex P Zijdenbos and Benoit M Dawant. Brain segmentation and white matter lesion detection in mr images. *Crit Rev Biomed Eng*, 22(5-6):401–465, 1994.

# Appendix A

The below shows the minimization process of Eq. 2.1, and we can finally get Eq. 2.7. Denote a functional as

$$I[f] = \int_{x_0}^{x_1} L(x, f, f', f'', \dots, f^{(n)}) dx; f' = \frac{df}{dx}, f'' = \frac{d^2f}{dx^2}, f^{(n)} = \frac{d^n f}{dx^n} \quad (\text{A.1})$$

Based on the Euler-Lagrange equation, the solution that minimizes the above functional should satisfy the following equation

$$\frac{\partial L}{\partial f} - \frac{d}{dx} \left( \frac{\partial L}{\partial f'} \right) + \frac{d^2}{dx^2} \left( \frac{\partial L}{\partial f''} \right) - \dots + (-1)^n \frac{d^n}{dx^n} \left( \frac{\partial L}{\partial f^{(n)}} \right) = 0 \quad (\text{A.2})$$

Based on Eq. 2.2, the integral of the internal energy can be written as

$$E(\mathbf{v}) = \frac{1}{2} \int_0^1 (\alpha |\mathbf{v}'|^2 + \beta |\mathbf{v}''|^2) ds \quad (\text{A.3})$$

To be consistent with the Euler-Lagrange equation mentioned above, we rewrite the variables as  $E = I, \mathbf{v}(s) = f(x), s = x$ . Therefore, we have

$$I(f) = \frac{1}{2} \int_0^1 \alpha |f'|^2 + \beta |f''|^2 dx \quad (\text{A.4})$$

$L$  can be expressed as

$$L = \alpha |f'|^2 + \beta |f''|^2 + 0f \quad (\text{A.5})$$

Then, we can get

$$\begin{aligned} \frac{\partial L}{\partial f} &= 0 \\ -\frac{d}{dx} \left( \frac{\partial L}{\partial f'} \right) &= -\frac{\partial}{\partial x} \left( \frac{\partial L}{\partial f'} \right) = -\frac{\partial}{\partial x} (2\alpha f') \\ \frac{d}{dx^2} \left( \frac{\partial L}{\partial f''} \right) &= \frac{\partial}{\partial x^2} \left( \frac{\partial L}{\partial f''} \right) = \frac{\partial}{\partial x^2} (2\beta f'') \end{aligned} \quad (\text{A.6})$$

Therefore, the solution should satisfy

$$0 - \frac{\partial}{\partial x} (2\alpha f') + \frac{\partial}{\partial x^2} (2\beta f'') = 0 \quad (\text{A.7})$$

which is equal to the following function after we transform the variables back

$$\frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{v}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{v}}{\partial s^2} \right) = 0 \quad (\text{A.8})$$

## Appendix B

The below shows the detailed derivation regarding the internal force of Eq. 3.18. Based on Eq. 3.17, the internal force can be written as

$$F_{int} = \frac{\partial}{\partial s}(\alpha \frac{\partial \mathbf{v}}{\partial s}) - \frac{\partial^2}{\partial s^2}(\beta \frac{\partial^2 \mathbf{v}}{\partial s^2}) \quad (\text{B.1})$$

We have

$$\frac{\partial \mathbf{v}}{\partial s} = \frac{\mathbf{v}_{i+1} - \mathbf{v}_i}{h} \quad (\text{B.2})$$

and

$$\begin{aligned} \frac{\partial^2 \mathbf{v}}{\partial s^2} &= \frac{\partial(\frac{\partial \mathbf{v}}{\partial s})}{\partial s} = \frac{\partial(\frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{h})}{\partial s} = \frac{1}{h}(\frac{\partial \mathbf{v}_i}{\partial s} - \frac{\partial \mathbf{v}_{i-1}}{\partial s}) = \frac{1}{h}(\frac{\mathbf{v}_{i+1} - \mathbf{v}_i}{h} - \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{h}) \\ &= \frac{\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}}{h^2} \end{aligned} \quad (\text{B.3})$$

Based on the above two equations, we can get

$$\begin{aligned} \frac{\partial}{\partial s}(\alpha \frac{\partial \mathbf{v}}{\partial s}) &= \frac{\partial}{\partial s}(\alpha \frac{\mathbf{v}_{i+1} - \mathbf{v}_i}{h}) = \frac{\alpha}{h}(\frac{\partial \mathbf{v}_{i+1}}{\partial s} - \frac{\partial \mathbf{v}_i}{\partial s}) = \frac{\alpha}{h}(\frac{\mathbf{v}_{i+2} - \mathbf{v}_{i+1}}{h} - \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{h}) \\ &= \frac{\alpha}{h^2}[(\mathbf{v}_{i+2} - \mathbf{v}_{i+1}) - (\mathbf{v}_i - \mathbf{v}_{i-1})] \end{aligned} \quad (\text{B.4})$$

$$\begin{aligned} \frac{\partial^2}{\partial s^2}(\beta \frac{\partial^2 \mathbf{v}}{\partial s^2}) &= \frac{\partial^2}{\partial s^2}(\beta \frac{\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}}{h^2}) = \frac{\beta}{h^2}[\frac{\partial^2}{\partial s^2}(\mathbf{v}_{i+1}) - \frac{\partial^2}{\partial s^2}(2\mathbf{v}_i) + \frac{\partial^2}{\partial s^2}(\mathbf{v}_{i-1})] \\ &= \frac{\beta}{h^2}[(\frac{\mathbf{v}_{i+2} - 2\mathbf{v}_{i+1} + \mathbf{v}_i}{h^2}) - 2(\frac{\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}}{h^2}) + \frac{\mathbf{v}_i - 2\mathbf{v}_{i-1} + \mathbf{v}_{i-2}}{h^2}] \\ &= \frac{\beta}{h^4}[(\mathbf{v}_{i+2} - 2\mathbf{v}_{i+1} + \mathbf{v}_i) - 2(\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}) + (\mathbf{v}_i - 2\mathbf{v}_{i-1} + \mathbf{v}_{i-2})] \end{aligned} \quad (\text{B.5})$$

So the numerical implementation of the Eq. B.1 can be represented as

$$\begin{aligned} F_{int} &= \frac{\partial}{\partial s}(\alpha \frac{\partial \mathbf{v}}{\partial s}) - \frac{\partial^2}{\partial s^2}(\beta \frac{\partial^2 \mathbf{v}}{\partial s^2}) \\ &= \frac{\alpha}{h^2}[(\mathbf{v}_{i+2} - \mathbf{v}_{i+1}) - (\mathbf{v}_i - \mathbf{v}_{i-1})] \\ &\quad - \frac{\beta}{h^4}[(\mathbf{v}_{i+2} - 2\mathbf{v}_{i+1} + \mathbf{v}_i) - 2(\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}) + (\mathbf{v}_i - 2\mathbf{v}_{i-1} + \mathbf{v}_{i-2})] \end{aligned} \quad (\text{B.6})$$