

# ROBUST SEGMENTATION AND OBJECT CLASSIFICATION IN NATURAL AND MEDICAL IMAGES

BY LIN YANG

A dissertation submitted to the  
Graduate School—New Brunswick  
Rutgers, The State University of New Jersey  
in partial fulfillment of the requirements  
for the degree of  
Doctor of Philosophy  
Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Peter Meer

and approved by

---

---

---

---

---

New Brunswick, New Jersey

May, 2009

© 2009

Lin Yang

ALL RIGHTS RESERVED

## ABSTRACT OF THE DISSERTATION

# Robust Segmentation and Object Classification in Natural and Medical Images

by Lin Yang

Dissertation Director: Prof. Peter Meer

Image segmentation and object classification are two fundamental tasks in computer vision. In this thesis, a novel segmentation algorithm based on deformable model and robust estimation is introduced to produce reliable segmentation results. The algorithm is extended to handle touching object and partially occluded image segmentation. A multiple class segmentation algorithm is described to achieve multi-class "object cut". The accurate results are achieved using the appearance and bag of keypoints models integrated over mean-shift patches. An affine invariant descriptor is proposed to model the spatial configuration of the keypoints. Besides working with 2D image segmentation problem, a robust, fast and accurate segmentation algorithm is illustrated for processing 4D volumetric data. One-step forward prediction is applied to generate the motion prior based on motion modes learning. Two collaborative trackers are introduced to achieve both temporal consistency and failure recovery. Multi-class classification algorithms using a gentle boosting is used to classify three types of breast cancer. The algorithm is Grid-enabled and launched on the IBM World Community Grid. We will introduce a fast and robust image registration algorithm for both 2D and 3D images. The algorithm starts from an automatic detection of the landmarks followed by a coarse to fine estimation of the nonlinear mapping. The parallelization of the algorithm on the IBM Cell Broadband Engine (IBM Cell/B.E.) will also be explained in details.

## Acknowledgements

I would like to express my sincere appreciation to my advisors. Dr. Peter Meer in the Electrical and Computer Engineering, Rutgers and Dr. David J. Foran in the Center of Biomedical Imaging and Informatics, UMDNJ and Cancer Institute of New Jersey. Both of them provide me numerous help and guidance over these years. I learned from them on not only research but also life. I would also like to acknowledge my appreciation to Ioana Meer and JoAnn Foran for their hospitality and nice food.

I appreciate all the time and patience of the other committee members: Dr. Kristin J. Dana, Dr. Yanyong Zhang and Dr. Manish Parashar.

I would like to acknowledge my advisors in Siemens Corporate Research and IBM T. J. Watson: Dr. Bogdan Georgescu, Dr. Dorin Comaniciu and Dr. Leiguang Gong, where I finished part of my thesis work. I would especially thank my colleagues in Center of Biomedical Imaging and Informatics Laboratory and Robust Image Understanding Laboratory: Wenjin Chen, Oncel Tuzel, Hong Zhang, Raghav Subbarao, Haifeng Chen, Xin Qi, Jun Hu, Will Cukierski, Oula Al-Dakkak and Vicky Chu.

Special thanks to my family for their tolerance and love.



## Dedication

To my parents, MIN and RENEE.

# Table of Contents

<b>Abstract</b> . . . . .	ii
<b>Acknowledgements</b> . . . . .	iii
<b>Dedication</b> . . . . .	iv
<b>List of Tables</b> . . . . .	viii
<b>List of Figures</b> . . . . .	ix
<b>1. Introduction</b> . . . . .	1
1.1. Organization of the Thesis . . . . .	3
<b>2. Image Segmentation Using Robust Estimation and Color Active Con-</b>	
<b>tour Models</b> . . . . .	5
2.1. A Quick Look at Active Contour Models . . . . .	7
2.2. The Robust Color GVF Snake . . . . .	9
2.2.1. Color Gradient in $Luv$ Color Space . . . . .	9
2.2.2. $L_2E$ Robust Estimation . . . . .	9
2.2.3. Robust Color GVF Snake . . . . .	12
2.3. Touching Objects Segmentation . . . . .	14
2.3.1. Boundary Contour Extraction . . . . .	14
2.3.2. Concave Points and Inner Edges Detection . . . . .	15
2.3.3. Touching Cells Segmentation . . . . .	16
2.3.4. Experimental Results . . . . .	19
2.4. Occluded Objects Segmentation . . . . .	21
2.4.1. Features and Shape Clustering . . . . .	23
2.4.2. Semi-density Approximation . . . . .	24

2.4.3.	Shape Alignment . . . . .	26
2.4.4.	The Variational Framework. . . . .	27
2.4.5.	Experimental Results . . . . .	28
<b>3.</b>	<b>Robust Multiple Class Segmentation Using Mean-shift Patches . . .</b>	<b>30</b>
3.1.	Unified Feature Representation Model . . . . .	32
3.1.1.	Mean-Shift Texton Histogram . . . . .	32
3.1.2.	Spatial Keyton Histogram . . . . .	34
3.1.3.	Global Shape Model . . . . .	36
3.2.	Training Procedure . . . . .	38
3.3.	Segmentation Algorithm and Testing . . . . .	39
3.4.	Experiments . . . . .	41
3.5.	Discussion . . . . .	44
<b>4.</b>	<b>Robust 4D Segmentation Using Prediction and Collaborative Models</b>	<b>45</b>
4.1.	Bayesian Tracking Framework . . . . .	46
4.2.	Motion Learning and Classifiers Training . . . . .	48
4.2.1.	Learning the Motion Modes . . . . .	48
4.2.2.	Learning the ED Detector . . . . .	50
4.2.3.	Learning the Boundary Classifiers . . . . .	51
4.3.	Tracking Procedure . . . . .	52
4.3.1.	Initialization of Tracking . . . . .	53
4.3.2.	One-Step Forward Prediction . . . . .	54
4.3.3.	Collaborative Trackers . . . . .	55
4.3.4.	Data Fusion . . . . .	56
4.3.5.	Postprocessing and Projection . . . . .	57
4.4.	Experimental Results . . . . .	57
<b>5.</b>	<b>Multiple Class Object Recognition in Medical Images . . . . .</b>	<b>61</b>
5.1.	IBM World Community Grid . . . . .	62

5.2. Data Generation and System Framework . . . . .	64
5.3. Data Analysis . . . . .	65
5.3.1. Features . . . . .	65
5.3.2. Dimension Reduction . . . . .	67
5.3.3. Classification . . . . .	68
5.4. Experiments . . . . .	72
 <b>6. A Fast Nonlinear Image Registration Using Landmarks and Robust Estimation</b> . . . . .	 76
6.1. Introduction . . . . .	76
6.2. Image Registration . . . . .	78
6.2.1. Landmark Detection and Matching . . . . .	79
6.2.2. Robust Estimation and Nonlinear Image Registration . . . . .	81
6.3. Current Parallelization Method For Image Registration . . . . .	83
6.4. Parallelization on the IBM Cell/B.E. . . . .	84
6.4.1. Data Partitioning and Parallelization . . . . .	85
6.4.2. Landmarks assembling and transformation estimation . . . . .	88
6.5. Experimental Results . . . . .	88
6.6. Conclusion . . . . .	91
 <b>7. Conclusions</b> . . . . .	 92
7.1. Direction for Future Research . . . . .	93
7.1.1. Online Learning for Object Recognition and Tracking . . . . .	93
7.1.2. Registration Based Heart Torsion Modeling . . . . .	95
<b>References</b> . . . . .	96
<b>Vita</b> . . . . .	107

## List of Tables

2.1. Segmentation accuracy(%) using the concave vertex graph. The $accuracy_c$ and $accuracy_n$ represent the segmentation accuracy for cytoplasm and nuclei respectively. . . . .	20
2.2. The segmentation accuracy(%) using the watershed algorithm and the concave vertex graph. . . . .	21
3.1. The segmentation accuracy for MHMS 11 database. . . . .	41
4.1. The <i>point-to-mesh</i> (PTM) errors measured in millimeters using three tracking algorithms. . . . .	58
5.1. The confusion matrix of three classes with column labels as inferred class and row labels as ground-truth classes . . . . .	73
6.1. The break points for the three algorithms . . . . .	88
6.2. The statistics of the running speed for 10 trials using the sequential and parallel multicore platform . . . . .	90

## List of Figures

2.1. The estimation results using least square ( $LS$ ), total least square ( $TLS$ ) and $L_2E$ for a linear regression problem of a dataset containing outliers.	11
2.2. Some representative morphologies of touching lymphocytes. In the first row, from left to right: CLL, MCL and FCC. In the second row, from left to right: ALL, AML and benign. Note the variations among the staining is because the specimens were prepared at different hospitals and institutions.	13
2.3. The segmentation result of robust color GVF snake. (a) The ROI contains only one cell. (b) The ROI contains the touching cells.	14
2.4. Construction of the concave vertex graph. (a) The original image with the yellow boundary contour. (b) High curvature points detection. (c) Concave points detection. (d) Inner edges detection. (e) The outer boundary $C$ , concave vertices $V$ and inner edges $E$ , superimposed on the original image. (f) The constructed concave vertex graph $G$ . The filling edges are shown with dotted lines.	15
2.5. The algorithm to separate touching cells using concave vertex graph.	17
2.6. The image segmentation results using robust color GVF snake: (a), (c) are the original images. (b), (d) are the corresponding segmentation results.	19
2.7. The segmentation results using the concave vertex graph.	20
2.8. The filtering response using our revised $MR$ filter bank.	23
2.9. The standard KDE and our proposed semi-parametric density approximation results upon a six Gaussian mixture density.	26
2.10. The occluded image segmentation results.	28

3.1.	The graphic model of the segmentation framework. . . . .	32
3.2.	A mean-shift patch and a $31 * 31$ square patch and their corresponding texton histograms. See text. . . . .	34
3.3.	An illustration of spatial keyton histogram, where we only show $r = 1, 2$ , and 3 in relative distance and $\theta = 0, \frac{\pi}{2}, \pi$ and $\frac{3\pi}{2}$ . The yellow region is the $10th(2 * 4 + 2, r = 3$ and $\theta = \frac{\pi}{2})$ bin $(10, \diamond)$ of the coordinate system with the position of the center $\diamond$ as the origin. . . . .	34
3.4.	Two spatial keyton histograms $h(i, j)$ with three keytons $\square, \Delta$ and $\diamond$ de- noting the 1st, 2nd and 3rd keyton. Affine transformations are performed on the keypoints belong to each keyton separately in the upper-left to produce the upper-right configuration. Nine $h(i, j)$ are shown for each of the two configurations. . . . .	35
3.5.	Examples of the Elliptical Fourier Descriptors (EFD). The contour is superimposed on original image. . . . .	37
3.6.	Segmentation algorithm and testing procedure. . . . .	38
3.7.	A testing example. (a) The original test image. (b) The mean-shift segmentation results. (c) The object labeling using appearance only. Different colors corresponding to different objects. (d)-(g) Four hypoth- esis for car, building, chair and cow. The brighter intensity means higher probability. (h) The object labeling which maximizes the sum of the log likelihood. (i) The refined segmentation result using the global shape of the car. (j) The hand-draw segmentation by a human. . . . .	41
3.8.	The confusion matrix for the MSRC dataset with row labels as inferred class and column as ground true class. . . . .	42
3.9.	Some segmentation results using our algorithm. The first row is the original image. The second row is the mean-shift segmented patches. The third row is the Harris corner detector results with red circles marking the keypoints. The fourth row is the labels provided by the algorithm. The fifth row is the hand-draw label by human. . . . .	43

4.1.	Manifold embedding for LV motion patterns. (a) Two LV boundary mesh sequences. (b) The 11 sequences embedded in a 2D subspace. Note: The end diastolic (ED) phase has larger volumes and represented as stars in (b), while the end systolic (ES) phase has smaller volumes and represented as squares in (b).	49
4.2.	The positive (a), (c) and negative samples (b), (d) used for training. (a) and (b) Training samples for the detector. (c) and (d) Training samples for the the boundary classifiers.	51
4.3.	The four canonical view and 3D representations of the segmentation result (automatic tracking initialization) of LV.	53
4.4.	The LV boundaries in 3D world coordinates of a motion mode. The results are calculated using TPS reverse mapping and superimposed in the two-dimensional reduced subspace.	55
4.5.	The prior $p(r1)$ for detection tracker (red solid line) and $p(r2)$ for template tracker (blue dotted line). The ED phase has frame index zero and ES phase is around frame six.	56
4.6.	Two volume-time curves demonstrate the whole cardiac cycle, which includes the systole stage and the diastole stage.	59
4.7.	Comparative tracking results on a testing sequence with 12 frames. (a) Tracking results using the proposed method. (b) Tracking by detection. (c) Tracking using 3D optical flow. The rows correspond to frame index 1, 6, 8 and 10.	60
5.1.	A screenshot of Help Defeat Cancer (HDC) clinet running on IBM world community grid (WCG).	62
5.2.	The work-flow and logical units of the Grid-enabled tissue microarray.	64



5.3.	Two breast tissue images on the left, benign (on the top) cancer (on the bottom). In order to keep the figure readable, only the texton maps of red rectangle regions were shown in the middle with different colors representing different textons. In practice the texton histograms of full images, shown on the right, were used for classification. . . . .	66
5.4.	The five different dimension reduction algorithms shown as five-fold cross-validation errors for different reduced dimensions. . . . .	68
5.5.	The binary gentle AdaBoost using an eight nodes classification and regression tree (CART) as the weak learner. . . . .	69
5.6.	Soft Margin SVM using the Mercer kernel based on $\chi^2$ distance. . . . .	71
5.7.	The multi-class gentle AdaBoost using an eight nodes classification and regression tree (CART) as the weak learner. . . . .	72
5.8.	The classification results. (a) The accuracy as the function of the size of training set using Gentle Boosting, Soft Margin SVM, $KNN$ with $K = 3$ or 5 and naive Bayesian. (b) The false positive and false negative errors using 20% images as training set. (c) Some misclassified samples. The upper row is false positive and lower row is false negative. . . . .	74
5.9.	The multi-class classification results using the gentle AdaBoost. The left three columns are correct classified samples and the right fourth column shows the failed cases. The first row is the benign samples. The second and third rows are the cancer samples. . . . .	75
6.1.	The image registration examples. (a) Three images which have different view points. (b) Three side view images which were taken for three different persons. . . . .	77
6.2.	The landmark detection using Harris corner detector. (a) The original images. (b) The images with the landmarks overlaid. . . . .	79
6.3.	The procedure for 3D image registration. (a) The procedure for the region based image registration. (b) The flowchart for the whole coarse to fine image registration algorithm. . . . .	80

6.4.	An example of a three layer Gaussian image pyramid and their corresponding 3D volumetric data. . . . .	81
6.5.	The RANSAC algorithm . . . . .	82
6.6.	Apply the robust estimation to find the robust landmark correspondence. (a) The original fixed and moving image. (b) The original pairs of matching points. (c) The robust matching points after rejecting the outliers. .	83
6.7.	The time profile for the entire nonlinear registration algorithm . . . . .	84
6.8.	The multi-core architecture of the IBM Cell/B.E. . . . .	85
6.9.	The procedure of the parallel image registration algorithm. (a) The flow chart of the parallelization. (b) The data partitioning using K-means clustering. . . . .	86
6.10.	The landmark distribution on each SPE unit of the IBM Cell/B.E. . . .	87
6.11.	Comparative registration results on the human abdomen CT image pair. (a) The fixed image. (b) The moving image. (c) The registration algorithm using the MedINRIA [95]. (d) The registration algorithm using ITK [66]. (e) The registration results using our algorithm. . . . .	89
7.1.	The online bagging algorithm. . . . .	93
7.2.	The online boosting algorithm. . . . .	94

# Chapter 1

## Introduction

Image segmentation and object recognition in natural and medical images is one of the most important topics in computer vision. The application can be extended to human computer intervention, computer aided diagnosis and automatic computer processing system, etc.

Image segmentation is the process of delineating an image into several "homogeneous" regions based on the similarity of pixel attributes. If the segmentation is performed independently by the computer, the processing is referred to as *unsupervised*. This is in contrast to *supervised* image segmentation which requires human input and intervention. Depending on whether or not prior knowledge has been used in the image segmentation, it can be classified as either "high-level" or "low-level" segmentation. Low-level image segmentation relies upon the pixel attributes of the image without consideration of prior-knowledge. For many practical applications, however, it is often necessary to incorporate additional "high-level" information into the analysis. There exists many difficulties within image segmentation area. This includes occluded image segmentation, multiple class segmentation and touching object segmentation, to mention only a few.

While it may be obvious that people are able to recognize objects under many variations in conditions, it is well known that object recognition and classification are extremely difficult for computers. Computer might be able to analyze simple features of objects and apply those features to recognize objects. However, it might have difficulties to recognize an object that varied slightly in form or was seen from a novel viewpoint because the features would be altered. Moreover, it might not be able to discriminate between two objects that contained the same features, but with a different organization.

In this thesis, we will present several novel algorithms which can handle robust segmentation and object classification in natural and medical images. We have described a color gradient vector flow (GVF) deformable model based on robust estimation by exploiting prior-knowledge of the specific applications. The algorithm starts from an initialization using  $L_2$  norm error ( $L_2E$ ) robust estimation. By merging  $L_2E$  robust estimation with the GVF snake and the  $Luv$  color space gradient, we will describe a quick and robust approach which is shown to produce reliable results for detecting and delineating the boundaries of imaged medical objects. The algorithm is able to provide satisfactory performance even when confronted with images exhibiting weak contrast and subtle edges. The algorithm is further extended to handle partially occluded image segmentation and touching objects segmentation.

A simple but effective segmentation framework will be described for multiple class object-based segmentation. The algorithm contains a novel but simple model to represent the spatial configurations of key points called spatial keyton histogram. The elliptical Fourier descriptor (EFD) is applied to refine the final segmentation results for certain type of objects. We demonstrate that our method provides good results for multiple class segmentation using real datasets.

A robust, fast and accurate 4D left ventricle (LV) segmentation algorithm is developed for the 3D echocardiography. According to our knowledge, this is the first study reporting fast and reliable 4D ultrasound segmentation of the left ventricle on a very large dataset, which contains 1143 3D volumetric data. From our research we report that collaborative trackers increased the tracking accuracy dramatically. The final accurate results are achieved by applying the motion priors using one-step forward prediction on the manifold. The robustness to complex background and weak edges come from the learned discriminative detectors and boundary classifiers, while the temporal consistence is preserved by template tracker. Instead of building specific models for heart, all the major steps in our algorithm are based on learning. Our proposed algorithm is therefore general enough to be extended to other 4D medical image segmentation problems.

We will describe a Grid-enabled framework which utilized texton histograms to

perform high throughput analysis of digitized breast cancer specimens. Experimental results will show that a gentle AdaBoost classifier using an eight node classification and regression tree (CART) decision tree as the weak learner provided the best results. We illustrate the classification results of separating benign from cancer and also two sub-classes of breast cancer.

Finally, we will present a new method for fast and robust image registration combining landmark and region based techniques. The algorithm is completely unsupervised and computationally efficient. Due to a relatively small number of landmarks, the method runs faster than many existing nonlinear registration algorithms reported in the literature, such as B-Spline based image registration and the Demon's algorithm. The method can also handle large transformation and deformation while still providing good registration results. We will also explain how to implement the algorithm in a multi-core platform, the IBM Cell Broadband Engine.

## 1.1 Organization of the Thesis

The thesis is organized as follows.

- In Chapter 2, we investigate the design, development, and implementation of a robust color gradient vector flow (GVF) active contour model for performing segmentation. The algorithms developed for this research operate in *Luv* color space, and introduce the color gradient and robust estimation into the traditional gradient vector flow (GVF) snake. The algorithm is shown to provide good results and successfully applied on segmenting blood cells. It is further extended to touching object and occluded image segmentation.
- In Chapter 3, the multiple class object-based segmentation is achieved using the appearance and bag of keypoints models integrated over mean-shift patches. We also propose a novel affine invariant descriptor to model the spatial relationship of keypoints and apply the Elliptical Fourier Descriptor (EFD) to describe the global shapes. The algorithm is computationally efficient and has been tested for three real datasets which contain a large number of natural objects. Our algorithm

provides better results than other studies reported in the literature.

- In Chapter 4, we present a robust, fast and accurate 4D left ventricle (LV) segmentation algorithm. We propose a novel one-step forward prediction to generate the motion prior using motion modes learning, and introduce two collaborative trackers to achieve both temporal consistency and failure recovery. Compared with tracking by detection and 3D optical flow, our algorithm provides the best results and subvoxel accuracy. The new algorithm is completely automatic and computationally efficient. It requires less than 1.5 seconds to process a 3D volume which contains 4,925,440 voxels.
- In Chapter 5, we introduce a Grid-enabled decision support system for performing automatic analysis of imaged breast tissue microarrays. Texture based features were extracted from the digitized images and isometric feature mapping (ISOMAP) is applied to achieve nonlinear dimension reduction. Iterative prototyping and testing are performed to classify several major subtypes of breast cancer. Overall the most reliable approach was gentle AdaBoost using an eight node classification and regression tree (CART) as the weak learner.
- In Chapter 6, we will explain a 2D/3D image registration algorithm. The method combines the landmark based and region based method together. Using a coarse-to-fine searching schema, the algorithm is accurate and computationally efficient. Using a C++ implementation, the algorithm can register a pair of 3D volumes which contains 24,641,536 voxels in less than 2 minutes. The parallelization of the algorithm on an IBM Cell Broadband Engine (IBM Cell/B.E.) will also be explained in details.

## Chapter 2

### Image Segmentation Using Robust Estimation and Color Active Contour Models

New segmentation methods have emerged which guide the partitioning process by utilizing cues based on shape, appearance and/or contextual models. In early low-level segmentation pixels are clustered based on their similarity in spatial and feature space and the resulting sub-regions are then assigned object labels. In contrast, high-level approaches attempt to detect and extract objects from images using prior knowledge and establish models which allow the segmentation method to adapt to the object of interest. Deformable models belong to this family of high-level image segmentation approaches.

There are two general types of deformable models described in the literature: parametric deformable models and geometric or level set based deformable models [70], [91], [12] (geometric deformable models and level set based deformable models are mathematically proved to be the same). Parametric deformable models have gained significant attention throughout the image processing community since its first introduction by Kass, Witkin and Terzopoulos [70]. Snakes are curves which are defined within the image's domain and move under the influence of internal forces within the curve and external forces derived from the image data. All snake properties and behaviors are specified through a function called energy function by analogy with physical systems. A partial differential equation controlling the snake causes it to evolve so as to reduce its energy and the local minima of this energy then correspond to desired image properties. Parametric deformable models have been used in a range of applications, including edge detection [70], object recognition [85], [36], shape modeling [36], [131] and motion tracking [85], [132], to mention only a few. In an almost parallel effort, a variety of

snakes based on utilizing the image gradients as external forces have been proposed. Examples include the traditional snake [4], [145], the balloon snake [21], the pressure forces model [22] and the more recently reported gradient vector flow (GVF) model [149]. The GVF snake model often outperforms other gradient-based models because it is insensitive to initialization values and can move into boundary concavities. It also has a much larger capture region than earlier approaches.

However, as the GVF snake was designed for binary or gray-level images, it is not straightforward to adapt this approach to color images. The original snake is also not robust approach which can be affected by noise and also sensitive to the initial positions. In this chapter we propose a robust color GVF snake [158], [152], [123] based on  $Luv$  color gradients and  $L_2E$  robust estimation. The proposed algorithm was shown to be effective in segmenting color pathology images [160]. We also extend the robust GVF snake into touching objects segmentation using concave vertex graph [161].

Geometric deformable models, or level set based deformable models, were almost simultaneously proposed by Caselles et al. [12] and by Malladi et. al. [91] to address the fact that parametric active contour models could not resolve topological changes. Geodesic deformable models are based on the theory of curve evolution and are numerically implemented using level set methods. They can automatically handle topology changes in an image and allow for multiple simultaneous boundary estimations. Furthermore, they are not sensitive to initial starting positions as parametric deformable models are. Therefore, these group of deformable models continue to gain increasing interests throughout the research community [67], [38], [14]. In this chapter, we also propose a level set based deformable model using semi-density approximation and global shape prior [153] to handle the occluded image segmentation problem.

In Section 2.1 we introduce the general active contour models. In Section 2.2 we describe the proposed robust color gradient vector flow snake. Section 2.3 provides an algorithm to segment touching objects. The occlude image segmentation algorithm is illustrated in Section 2.4.



## 2.1 A Quick Look at Active Contour Models

Defined within an image domain, a traditional deformable model [70] is parametrically defined as  $\mathbf{x}(s) = (x(s), y(s))$ , where  $x(s)$  and  $y(s)$  are  $x$  and  $y$  coordinates along the contour and  $s$  represents the arc-length with value in  $[0, 1]$ , to minimize an energy function as follows:

$$E_{snake} = \int_0^1 (E_{int}(\mathbf{x}(s)) + E_{ext}(\mathbf{x}(s))) ds \quad (2.1)$$

where the first term represents the internal energy of snake, and the second term represents the external forces pushing the snake toward the desired objects' edges. The internal energy is defined as:

$$E_{int}(\mathbf{x}(s)) = (\alpha |\mathbf{x}_s(s)|^2 + \beta |\mathbf{x}_{ss}(s)|^2)/2 \quad (2.2)$$

where  $\mathbf{x}_s(s)$  is the first derivative of  $\mathbf{x}(s)$  and  $\mathbf{x}_{ss}(s)$  is the second derivative of  $\mathbf{x}(s)$  with respect to  $s$ . The external energy is defined as the image energy which is derived from the image data over the position that the snake lies. This energy function attracts snakes to salient features in images such as lines and edges. The edges of the image is actually the boundary between color differences. Therefore, the external energy is usually defined as follows:

$$E_{ext}(\mathbf{x}(s)) = -|\nabla \{G_{\sigma(x,y)} * I(x,y)\}| \quad (2.3)$$

where  $I(x, y)$  is the image intensity at point  $(x, y)$ ,  $G_{\sigma(x,y)}$  is the two dimensional Gaussian kernel with  $\sigma$  as standard deviation. Therefore, the external energy in the traditional snake is defined as the negative value of the gradient of the Gaussian smoothing image.

The goal of the active contour models is to find the local minima of  $E_{snake}$  defined in equation (2.1), Based on the Euler-Lagrange principle. Equation (2.1) has a minimum only if the Euler-Lagrange differential equation is satisfied as:

$$\alpha * \mathbf{x}_{ss}(s) - \beta * \mathbf{x}_{ssss}(s) - \gamma * \nabla E_{ext}(\mathbf{x}(s)) = 0 \quad (2.4)$$

where  $\mathbf{x}_{ss}(s)$  and  $\mathbf{x}_{ssss}(s)$  are the second and fourth derivatives of the curve with respect to the parameter  $s$ .

In order to find the solution for equation (2.4), the snake is made dynamic by defining  $\mathbf{x}$  as the function of time  $t$  and  $s$  as follows:

$$\mathbf{x}_t(s, t) = \alpha * \mathbf{x}_{ss}(s, t) - \beta * \mathbf{x}_{ssss}(s, t) - \gamma * \nabla E_{ext}(\mathbf{x}(s, t)) \quad (2.5)$$

Then the partial derivative of  $\mathbf{x}$  with respect to  $t$  is the same as the left side in equation (2.4). When  $\mathbf{x}$  gets a stable value, its partial derivative with respect to  $t$  will be 0 and we get the solution for equation (2.4). The solution to equation (2.5) can be achieved by solving the discrete equations iteratively.

To effectively lead the snakes into concave region, Xu and Prince [149] proposed a new external force. According to Helmholtz theorem [53], rewriting (2.5) and replacing the  $-\nabla E_{ext}(\mathbf{x}(s))$  with  $\Theta$

$$\mathbf{x}_t(s, t) = \alpha \mathbf{x}_{ss}(s, t) - \beta \mathbf{x}_{ssss}(s, t) + \Theta \quad (2.6)$$

where  $\Theta$  is the gradient vector flow defined as  $\Theta(x, y) = [u(x, y), v(x, y)]$  that minimizes the energy functional

$$\begin{aligned} \Psi = & \int \int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \\ & + (|\nabla \{G_{\sigma(x,y)} * f(x, y)\}|^2 \cdot \\ & |\Theta - \nabla \{G_{\sigma(x,y)} * f(x, y)\}|^2) dx dy \end{aligned} \quad (2.7)$$

where  $\nabla \{G_{\sigma(x,y)} * f(x, y)\}$  is the gradient of the input image  $f(x, y)$  after Gaussian smoothing with variance  $\sigma$  and mean  $\mathbf{0}$ . Equation (2.7) is dominated by  $u_x^2 + u_y^2 + v_x^2 + v_y^2$  when  $|\nabla \{G_{\sigma(x,y)} * f(x, y)\}|$  is small. When  $|\nabla \{G_{\sigma(x,y)} * f(x, y)\}|$  is large, the second term dominates the integrand and is minimized when  $\Theta = \nabla \{G_{\sigma(x,y)} * f(x, y)\}$ . This result keeps  $\Theta$  nearly equal to the gradient of the edge when the snake is near the object, while enabling the snake to move towards the edges when it is far away from the object.

## 2.2 The Robust Color GVF Snake

In this section we will introduce the color GVF snake model using robust estimation to provide the initializations.

### 2.2.1 Color Gradient in $Luv$ Color Space

In gray-level images, the gradient is defined as the first derivative of the image luminance. It has a high value in those regions exhibiting high luminance contrast. However, this strategy is not suitable for color images. Simply transforming color images into gray-level image by taking the average of three channels and applying the gray-level image gradient operator do not provide satisfactory results in our applications.

We adopt the definition of gradients for color images [163], [119], [49]. In contrast to previous approaches, we define the color gradient in  $Luv$  color space rather than  $RGB$  color space because Euclidean metrics and distances are perceptually uniform in  $Luv$  color space, which is not the case in  $RGB$  color space [119].

Let  $\Gamma(x, y) : \mathbb{R}^3$  be a color image, based on classical Riemannian geometry results [73], the  $L_2$  norm can be written in matrix form

$$d\Gamma^2 = \begin{bmatrix} dx \\ dy \end{bmatrix}^T \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \quad (2.8)$$

where  $g_{11} = [\frac{\partial \Gamma}{\partial x}]^2$ ,  $g_{12} = g_{21} = \frac{\partial \Gamma}{\partial x} \frac{\partial \Gamma}{\partial y}$ ,  $g_{22} = [\frac{\partial \Gamma}{\partial y}]^2$ .

The quadratic form (2.8) achieves its extreme changing rates in the directions of the eigenvectors of matrix  $[g_{i,j}]$ ,  $i = 1, 2, j = 1, 2$  and the changing magnitude is decided by its eigenvalues  $\lambda_+$  and  $\lambda_-$ . In our approach, we select  $\sqrt{\lambda_+ - \lambda_-}$  [119], [50] to define the color gradient.

### 2.2.2 $L_2E$ Robust Estimation

In the previous section, we defined the color gradient in  $Luv$  color space in order to replace the gray level gradient of the original GVF snake. Although the capture range of original GVF snake is large, it still may fail to find the edges of the object when

given unsuitable initial positions. For example, to capture the boundary of a circular object the initial snake must include the center of the circle [112]. Another potential difficulty relates to computational complexity. Given an initial location far away from the object, the original GVF snake may take a long time to converge to the edges of the object. Because of these challenges, it was necessary to introduce a robust estimation method to provide accurate initial positions.

One of the common approaches used to resolve this sort of classification problem is the least square (*LS*) approach. Define the image classification model as

$$\mathbf{g} = \mathbf{F}\mathbf{w} + \mathbf{e} \quad (2.9)$$

where  $\mathbf{F}$  is the sample image, a  $m \times 3$  matrix with  $m = w \cdot h$  as the total pixels in the image,  $w$  is the width of the image and  $h$  is the height of the image. Applying the ordinary least square principle, we obtain the least square solution

$$\mathbf{w} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{g}. \quad (2.10)$$

By choosing  $\mathbf{g}$  as 1 and -1 to denote the background and the object, respectively, *LS* provides a least square solution to the classification problem.

A total least square (*TLS*) performs better by minimizing the error between both  $\mathbf{g}$ ,  $\hat{\mathbf{g}}$  and  $\mathbf{F}$ ,  $\hat{\mathbf{F}}$ . The weights  $\mathbf{w}$  are given by the right singular vector corresponding to the smallest eigenvalue of the singular value decomposition (*SVD*) of  $\begin{bmatrix} \mathbf{F} & \mathbf{g} \end{bmatrix}$ , which is the stacked data of input image  $\mathbf{F}$  and output image  $\mathbf{g}$ . It can be shown [97] that the solution is

$$\mathbf{w} = -\frac{\mathbf{v}_{n+1}(1:n)}{\mathbf{v}_{n+1}(n+1)} \quad (2.11)$$

where  $\mathbf{v}_{n+1}$  corresponding to the right singular vector of the smallest singular value.

Because  $L_2$  norm is sensitive to outliers, neither *LS* nor *TLS* are robust estimators. In order to address the outliers and variabilities of the images,  $L_2E$  robust estimation and training have been proposed. By merging training and  $L_2E$  robust estimation with

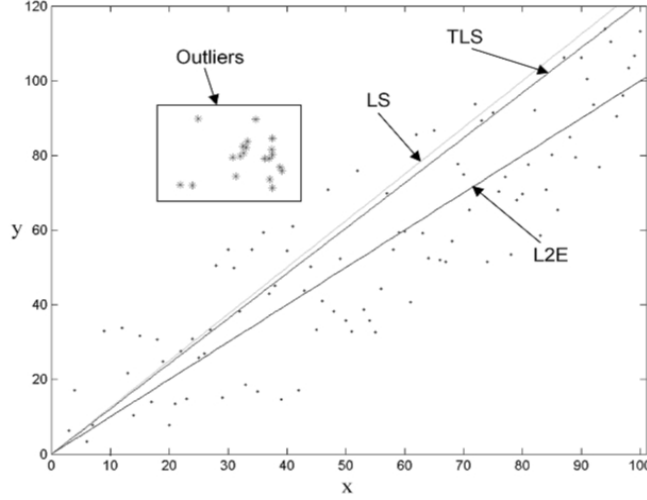


Figure 2.1: The estimation results using least square ( $LS$ ), total least square ( $TLS$ ) and  $L_2E$  for a linear regression problem of a dataset containing outliers.

color GVF snake, the segmentation accuracy improved significantly, while simultaneously saving computation time. In order to obtain a robust estimation of the initial locations of the nucleus and cytoplasm of cells without being affected by outliers, we adopt a statistically robust matching criteria based on the minimization of the integral squared error ( $ISE$ ) also known as  $L_2E$  between a Gaussian model of the residual and the true density function of the residual. It was shown in [121], [120] that minimum distance estimators, including the  $L_2E$ , are inherently robust without requiring the specification of tuning parameters.

Figure 2.1 is a linear regression example showing the capacity of  $L_2E$  robust estimation to reject outliers. One can easily see that the  $LS$  and  $TLS$  are perturbed by the outliers on the left upper side of Figure 2.1, marked with a rectangle, while the  $L_2E$  based fit has successfully rejected the outliers.

In (2.9), the residual  $\mathbf{e}$  is assumed to be composed of independent, identically distributed random variables, whose distribution will be modeled by a Gaussian with mean  $\mathbf{0}$  and variance  $\sigma$ . Our goal is to minimize the integrated square error ( $ISE$ ) or  $L_2E$  error measurement given by

$$ISE(\hat{\theta}) = E_{\hat{\theta}} \int [g(\mathbf{e}|\hat{\theta}) - h(\mathbf{e}|\theta)]^2 d\mathbf{e} \quad (2.12)$$

where  $g(\cdot)$  is the Gaussian function modeling the density of the residual error,  $\hat{\theta}$  and

$\theta$  are the estimation parameter and the true parameter respectively, and  $h$  is the true unknown density of the residual error term.  $E_{\hat{\theta}}(\cdot)$  denotes the estimate of the integral. Considering minimizing an estimate of  $ISE$  with respect to  $\theta$ :

$$\begin{aligned}\hat{\theta} &= \arg \min_{\hat{\theta}} E_{\hat{\theta}} \int [g(\mathbf{e}|\hat{\theta}) - h(\mathbf{e}|\theta)]^2 d\mathbf{e} \\ &= \arg \min_{\hat{\theta}} [E_{\hat{\theta}} \int g^2(\mathbf{e}|\hat{\theta}) d\mathbf{e} - 2E_{\hat{\theta}} \int g(\mathbf{e}|\hat{\theta}) h(\mathbf{e}|\theta) d\mathbf{e} \\ &\quad + E_{\hat{\theta}} \int h^2(\mathbf{e}|\theta) d\mathbf{e}].\end{aligned}\tag{2.13}$$

The third term  $h^2(\cdot)$  is independent of  $\hat{\theta}$  and can be omitted from the minimization. The first term in the expansion is  $E_{\hat{\theta}} \int g^2(\cdot) d\mathbf{e}$  and can be evaluated in closed form. The second term is  $-2E_{\hat{\theta}} \int g(\mathbf{e}|\hat{\theta}) h(\mathbf{e}|\theta) d\mathbf{e}$  with  $E_{\hat{\theta}}[g(\cdot)]$  being the expectation of  $g(\cdot)$  with respect to  $\hat{\theta}$ . The following Rudemo's unbiased estimator can be used for the second term [118]

$$-\frac{2}{m} \sum_{i=1}^m g(e_i|\theta) \quad \text{for } i = 1, \dots, m\tag{2.14}$$

where  $m = w \cdot h$  denotes the total number of pixels. Thus, the minimization using the  $L_2E$  criterion for normal density is given by

$$\hat{\theta}_{L_2E} = \arg \min_{\theta=[\sigma, \mathbf{w}]} \left[ \frac{1}{2\sqrt{\pi}\sigma} - \frac{2}{m} \sum_{i=1}^m \frac{1}{2\sqrt{\pi}\sigma} \exp \left( -\frac{\|\mathbf{g} - \mathbf{F}\mathbf{w}\|^2}{2\sigma^2} \right) \right]\tag{2.15}$$

The numerical solution is calculated by applying the gradient descant algorithm to equation (2.15). For readers who are interested in robust estimators, we should point out that  $L_2E$  robust estimation has many useful applications, such as image registration [88], computer latency and workload prediction [157], [156], [54], applied statistics [45], [118] and data mining [72] .

### 2.2.3 Robust Color GVF Snake

Now we describe the design and development of the robust color GVF snake. Rewriting equation (2.6) and replacing the original GVF vector field  $\Theta$  with the new color GVF vector field  $\Theta$  in  $Luv$  color space, we have

$$E_{snake} = \alpha v_{ss}(s) - \beta v_{sss}(s) + \Theta\tag{2.16}$$

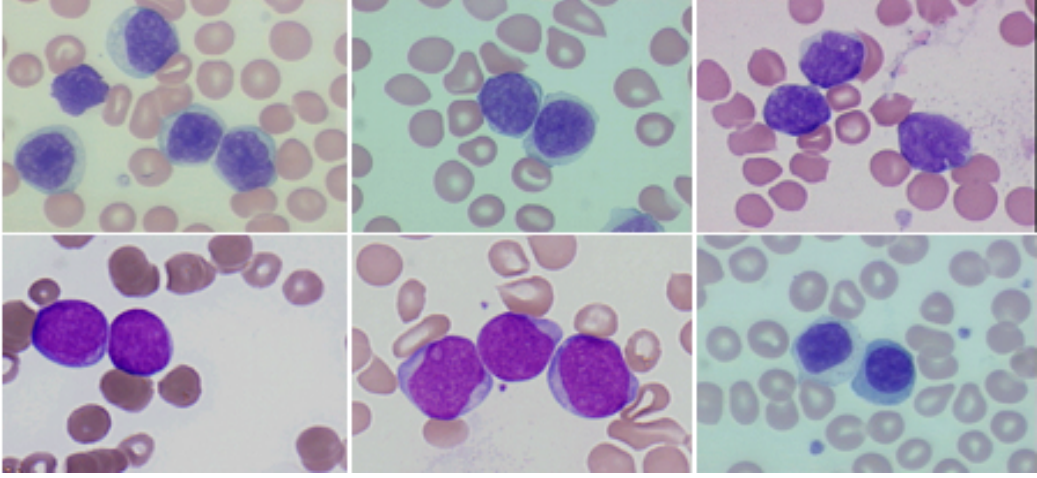


Figure 2.2: Some representative morphologies of touching lymphocytes. In the first row, from left to right: CLL, MCL and FCC. In the second row, from left to right: ALL, AML and benign. Note the variations among the staining is because the specimens were prepared at different hospitals and institutions.

where  $\Theta$  is the color gradient vector flow defined as  $\Theta(x, y) = [u(x, y), v(x, y)]$  that minimizes the energy function defined in equation (2.7).

In the robust color GVF snake,  $\nabla$  in equation (2.7) is the *Luv* color gradient. The two initial snake contour locations are obtained from  $L_2E$  robust estimation and correspond to the boundaries of the nucleus and the cytoplasm respectively. By applying calculus of variations, it can be shown that minimizing the integral in equation (2.7) is equal to solve the following equation

$$\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \quad (2.17)$$

$$\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0. \quad (2.18)$$

The numerical solution to (2.17) and (2.18) can be solved by treating  $u$  and  $v$  as functions of time and solving

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - (u(x, y, t) - f_x(x, y)) * (f_x^2(x, y) + f_y^2(x, y)) \quad (2.19)$$

$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - (v(x, y, t) - f_y(x, y)) * (f_x^2(x, y) + f_y^2(x, y)). \quad (2.20)$$

In order to set up an iterative solution, let the indices  $i, j$  and  $n$  correspond to  $x, y$ , and  $t$  will be derived from the iterative algorithm for (2.19) and (2.20).

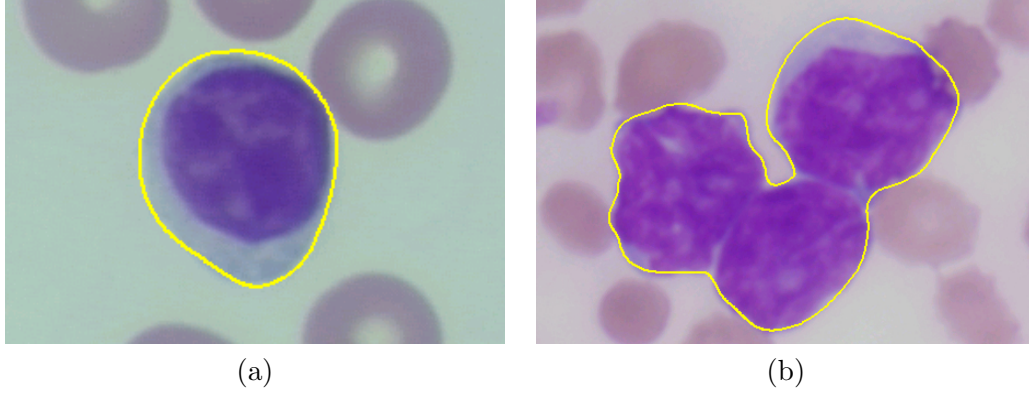


Figure 2.3: The segmentation result of robust color GVF snake. (a) The ROI contains only one cell. (b) The ROI contains the touching cells.

### 2.3 Touching Objects Segmentation

In real applications, there always exist touching objects which are challenging to be accurately segmented using traditional methods. This problem is especially prominent in medical images. In Figure 2.3 we show several malignant touching cells. In this section, we describe a novel algorithm which can reliably handle touching cells segmentation. The algorithm is based on the previously proposed color GVF snake and the concave vertex graph.

#### 2.3.1 Boundary Contour Extraction

The initial step of the algorithm is to extract the boundary contour of the touching cells. We first apply a robust color GVF snake proposed by [158] in each region of interest (ROI). This algorithm has two steps. The first step is a  $L_2E$  robust estimation [121], which provides a rough estimate of the starting position of the deformable model. The second step is the gradient vector flow (GVF) snake [149] using  $Luv$  [148, Sec. 8.4] color gradients as the external force. The touching cases are shown in Figure 2.3b, where the output contour represents the outer boundary of the touching cells.



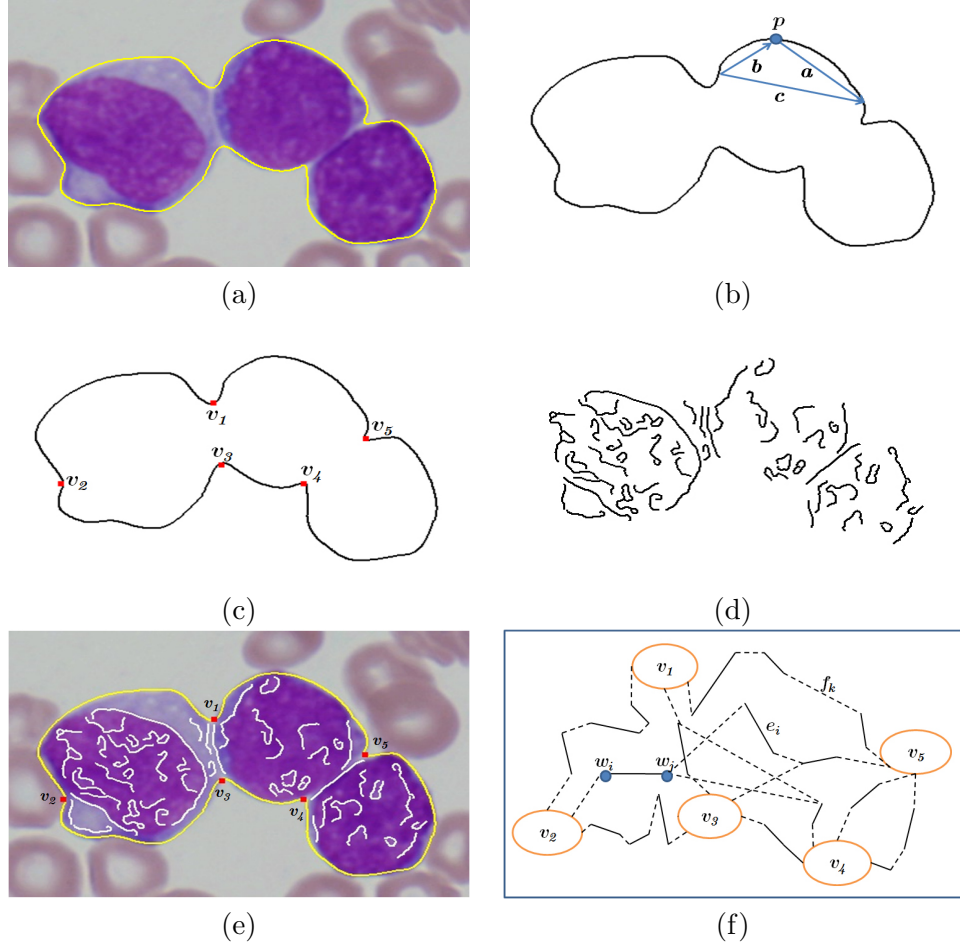


Figure 2.4: Construction of the concave vertex graph. (a) The original image with the yellow boundary contour. (b) High curvature points detection. (c) Concave points detection. (d) Inner edges detection. (e) The outer boundary  $C$ , concave vertices  $V$  and inner edges  $E$ , superimposed on the original image. (f) The constructed concave vertex graph  $G$ . The filling edges are shown with dotted lines.

### 2.3.2 Concave Points and Inner Edges Detection

In Figure 2.4, we show the construction of the concave vertex graph. The boundary of the touching cells is illustrated as the yellow contour in Figure 2.4a. We then apply the algorithm in [18] to detect the high curvature points on the contour (Figure 2.4b). At each point  $p$  a set of triangles are constructed which satisfies

$$d_{\min} \leq |\mathbf{a}| \leq d_{\max} \quad d_{\min} \leq |\mathbf{b}| \leq d_{\max} \quad \alpha \leq \alpha_{\max} \quad (2.21)$$

where  $\alpha = \arccos \frac{|\mathbf{a}|^2 + |\mathbf{b}|^2 - |\mathbf{c}|^2}{2|\mathbf{a}||\mathbf{b}|}$ ,  $d_{\min}, d_{\max} = 7, 9$  pixels and  $\alpha_{\max} = 150^\circ$  are kept. The candidates are further processed to suppress the local nonmaxima points. The final

high curvature points correspond to both concave and convex points. We keep only the concave points, shown as red rectangles in Figure 2.4c. This can be calculated from the sign of the cross product  $\mathbf{a} \otimes \mathbf{b}$ , which has to be negative for concave points.

Canny edge detector is applied inside the cell region and straight line fitting is used to model the edges (Figure 2.4d). The separating curve combines a pair of convex vertices on the boundary and is enforced to pass through the inner edges.

### 2.3.3 Touching Cells Segmentation

The outer boundary of the touching cells is defined as  $C$ , and the region enclosed by  $C$  is  $R(C)$ . The concave points are the set  $V$ , e.g.  $v_1 - v_5$  which are shown in Figure 2.4e. The inner edges are the set  $E$ , e.g. shown as white solid lines in Figure 2.4e and also illustrated by  $e_i$  in Figure 2.4f.

**Concave Vertex Graph.** In Figure 2.4f we construct the concave vertex graph  $G$ . Let  $W$  be the vertex set consisting of the end points of inner edges  $E$ , e.g.  $w_i$  and  $w_j$  in Figure 2.4f. The vertices of graph  $G$  are then equal to  $V \cup W$ . Let  $F$  be the *filling edges* which connect the gaps among all these vertices in  $G$ , e.g.  $f_k$  in Figure 2.4f.

In graph  $G$ , if the edges are real inner edges  $E$ , we set their length as a small positive number  $\epsilon$  ( $10^{-16}$ ). If the edges are gap filling edges  $F$ , we set their length as the real distance. The length of the edges is defined in this way to restrict the segmentation to follow real inner edges, because most of the time the trivial solution of directly connecting two concave vertices using only filling edges in  $G$ , will provide longer distance than a path using real inner edges. A path  $p_{ij}$  is defined as the shortest path between the concave vertices  $v_i, v_j \in V$ . The Dijkstra's algorithm, which is a greedy algorithm but guarantees to provide the optimal solution, is used to find the  $p_{ij}$  between  $v_i$  and  $v_j$ . The length of the  $p_{ij}$ ,  $\|p_{ij}\|$ , is defined as the total length of the *filling edges*  $f_k$  in the shortest path  $p_{ij}$

$$\|p_{ij}\| = \sum_{f_k \in p_{ij}} \text{length}(f_k). \quad (2.22)$$

In Figure 2.4f, as an example, we can see  $\|p_{12}\| > \|p_{13}\|$ , because  $p_{12}$  traverse much longer filling edges than  $p_{13}$ .

**Input:** Given the region of interest (ROI) containing touching cells.

- Extract the boundary contour  $C$ , detect the concave points  $V$ , the inner edges  $E$  in  $R(C)$ , construct the concave vertex graph  $G$ .
- for each vertex  $v(i) \in V$ 
  - Find the path  $p_{ij}$  and calculate the length  $\|p_{ij}\|$  using (2.22).
- Initialize  $mincost = +\infty$  and  $Q = \emptyset$ .
- while ( $V$  is not empty)
  - for each vertex  $v(i) \in V$ 
    - \* for each vertex  $v(j \neq i) \in V$ 
      - Apply the path  $p_{ij}$  to separate the graph  $G$  in to  $L$  and  $R$ .
      - Calculate the cost  $c$  using (2.26) and save in  $Q$ .
  - Sort  $Q$  and pick up the path  $p_{ij}$  with the lowest cost  $c$ .
  - if ( $c < 1.5 * mincost$ )
    - \* Record path  $p_{ij}$  and the region  $R(C, p_{ij})$  with cost  $c$  in the *result*.
    - \* The edges and vertices in the  $R(C, p_{ij})$  are removed from  $G$ .
    - \* Set  $mincost = c$  and  $Q = \emptyset$ .
  - else return *result*.

Figure 2.5: The algorithm to separate touching cells using concave vertex graph.

After the Dijkstra’s algorithm was applied, we found all  $p_{ij}$ -s, which were valid candidates following the real inner edges to separate touching cells. In our algorithm, we treat the touching cells segmentation as recursively searching for the best path  $p_{ij}$  in  $G$ , which minimizes a cost function specifically designed to prefer cell-like object-cut.

**Cost Function.** We are looking for perceptually ”good” segmentation of touching cells. For this purpose, we design the cost function to represent the clues that clinical pathologists use for judgement.

- The cells should be objects which are perceptually salient, because humans tend to separate such objects in an image. A good definition of saliency is proposed in [127] based on the Gestalt laws [37]. We apply the minimum of two saliency costs

$$c_s = \min \left( \frac{\|p_{ij}\|}{\sqrt{area_L(C, p_{ij})}}, \frac{\|p_{ij}\|}{\sqrt{area_R(C, p_{ij})}} \right) \quad (2.23)$$

where  $\|p_{ij}\|$  is the length defined in (2.22), each path  $p_{ij}$  in  $G$  divides  $R(C)$  into two regions  $L$  and  $R$ , and the *min* function in (2.23) selects the region with the

smallest cost. The  $area(C, p_{ij})$  denotes the area enclosed by  $C$  and path  $p_{ij}$ .

- The cells are objects which are close to elliptical shape and can be modeled by ellipse fitting using points on  $C$  and  $p_{ij}$ . The ratio between the long and short axes is recorded as  $tg$ . The segmented objects are expected to provide a ratio  $tg$  in the range  $[tg_1, tg_2]$ , in which case the  $dist(tg, [tg_1, tg_2]) = 0$ . Otherwise, we define  $dist(tg, [tg_1, tg_2]) = \min(|tg - tg_2|, |tg - tg_1|)$ .

$$c_g = \min \left( \frac{1}{1 + \exp(-dist(tg_L, [tg_1, tg_2]))}, \frac{1}{1 + \exp(-dist(tg_R, [tg_1, tg_2]))} \right) \quad (2.24)$$

where the  $L$  and  $R$  have the same definition as (2.23). The  $tg_1 = 0.5$  and  $tg_2 = 1.5$  are selected by fitting an Gaussian distribution on the ratio with 90% confidence region.

- The cells are objects which have biologically reasonable areas. Following the definition above, we use  $[ta_1 = \pi * 80^2, ta_2 = \pi * 120^2]$  for the  $60\times$  magnification in our tests, which are selected by fitting an Gaussian distribution on the area.

$$c_a = \min \left( \frac{1}{1 + \exp(-dist(ta_L, [ta_1, ta_2]))}, \frac{1}{1 + \exp(-dist(ta_R, [ta_1, ta_2]))} \right). \quad (2.25)$$

- The final cost  $c$  is the weighted sum

$$c = \lambda_1 c_s + \lambda_2 c_g + \lambda_3 c_a \quad \sum_{i=1}^3 \lambda_i = 1. \quad (2.26)$$

The optimal values of coefficients are selected as  $\lambda_1 = 0.5$ ,  $\lambda_2 = 0.3$  and  $\lambda_3 = 0.2$ , which are learned in an offline process using a training set and held constant throughout the experiments.

**Algorithm.** Using the concave vertex graph  $G$  and the cost function  $c$ , the method is described in Algorithm 2.5. It is recursively applied to separate touching cells until all the vertices  $v_i \in V$  is empty or the cost  $c < 1.5 * mincost$ , which represents oversegmentation. The algorithm only separates the cytoplasm of touching cells. The nuclei in each cell is further separated from its cytoplasm using [158]. In order to provide

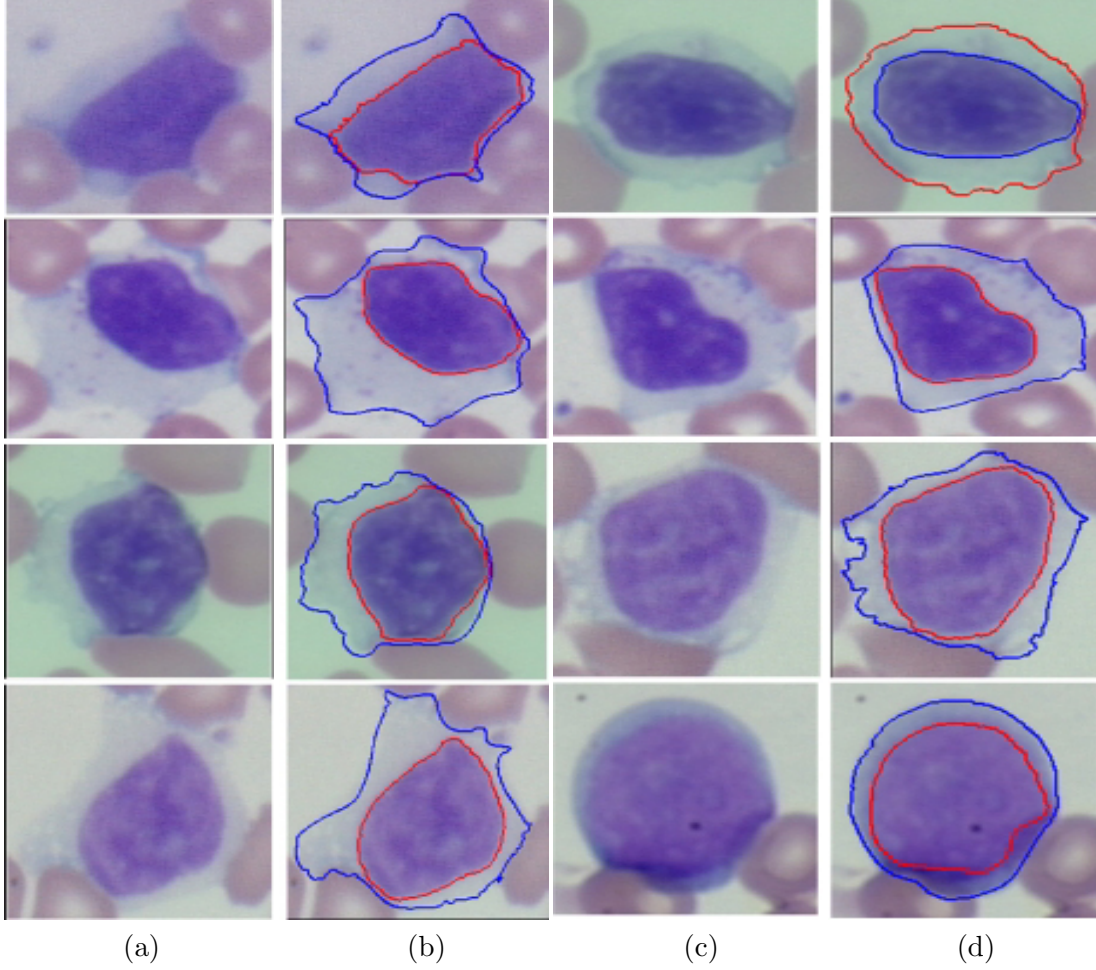


Figure 2.6: The image segmentation results using robust color GVF snake: (a), (c) are the original images. (b), (d) are the corresponding segmentation results.

smooth boundaries, we apply the quadratic splines to postprocess the boundaries of each segmented cell.

### 2.3.4 Experimental Results

The cell database consists of a mixed set of 86 hematopathology cases: 18 Mantle Cell Lymphoma (MCL), 20 Chronic Lymphocytic Leukemia (CLL), 9 Follicular Center Cell Lymphoma (FCC), 18 Acute Lymphocytic Leukemia (ALL), 19 Acute Myelocytic Leukemia (AML), and 19 benign cases. For each case, there are varying number of cell images from 10 to 90. In total there exists 3898 cell images in our complete database.

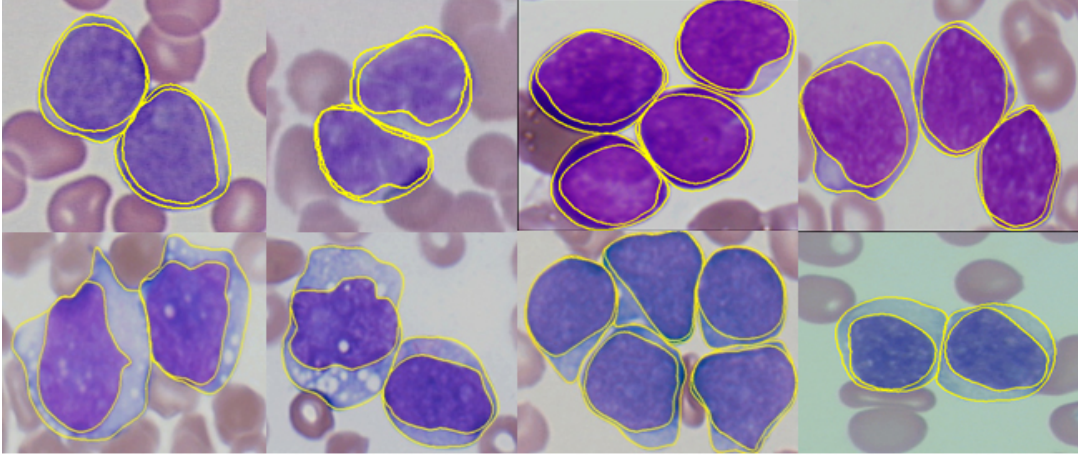


Figure 2.7: The segmentation results using the concave vertex graph.

Table 2.1: Segmentation accuracy(%) using the concave vertex graph. The  $accuracy_c$  and  $accuracy_n$  represent the segmentation accuracy for cytoplasm and nuclei respectively.

	Benign	CLL	MCL	FCC	AML	ALL
$accuracy_c$ (%) of touching cells	90.1	90.8	86.4	86.9	86.3	85.2
$accuracy_n$ (%) of touching cells	92.3	91.2	88.1	88.7	87.5	87.9
$accuracy_c$ (%) of all cells	92.5	91.7	87.2	89.1	88.5	87.6
$accuracy_n$ (%) of all cells	95.8	92.8	90.1	91.0	88.9	89.2

All cases originated from the archives of City of Hope Hospital in California, University of Pennsylvania of School of Medicine, Spectrum Health System, Grand Rapids, MI and Robert Wood Johnson Medical School, University of Medicine & Dentistry of New Jersey.

The test platform for the experiments consisted of an Intel-based workstation interfaced with a high-resolution Olympus DP70 camera equipped with 12-bit color depth on each color channel and 1.45 million pixel effective resolution. The system also includes a single 2/3 inch CCD digital camera, an Olympus AX70 microscope equipped with a Prior 6-way robotic stage, motorized objective turret and a magnification changer. In order to validate the proposed algorithm, two sets of experiments are performed.

We compare the segmentation results with manual segmentation. Two sets of experiments are performed.

- The 217 touching cases of the hematologic cell image dataset.

Table 2.2: The segmentation accuracy(%) using the watershed algorithm and the concave vertex graph.

	Mean	Median	Min	Max	80%
Watershed	74.3	75.1	65.4	82.7	72.9
Concave Vertex Graph	88.9	90.2	75.2	95.5	87.1

- *The complete database which contains 3898 hematologic cell images.*

Figure 2.6 shows some results of difficult single cell image segmentation using robust color GVF snake, where (a), (c) are original images and (b), (d) exhibit corresponding segmentation results. Some images exhibit very weak differences between cytoplasm and background. Some images have complex texture or even several different color regions within the cytoplasm. In Figure 3.9, we show the touching cells segmentation results. In Table 2.1 we present the segmentation accuracies for the six different classes of lymphocytes in two set of experiments. We obtained an average accuracy 88.9% on the touching cells dataset and 90.1% on the complete database.

Only a limited number of recent literature reports work on the touching cells segmentation in pathology images which are prepared with hematoxylin staining and imaged at high resolution (60 $\times$ ). As the watershed algorithm is widely accepted for touching object segmentation and successfully used in segmenting pathology images [1], we compared our method with watershed using the 217 touching cell image dataset and listed the results in Table 2.2. The 80% column in Table 2.2 represents the sorted 80% highest accuracy of all the results, and is commonly used by doctors to evaluate the usability of the system. It is obvious that the proposed algorithm performed much better than the watershed.

## 2.4 Occluded Objects Segmentation

There exist a lot of effort in recent literature to combine top down information to improve the segmentation results. Kumar et al. [75] presents the OBJ CUT algorithm which applied the pictorial structure (PS) model and Markov Random Field (MRF) to combine the top-down and bottom-up cues for object-based image segmentation. A

parts-based image segmentation approach, the Layout Consistent Random Field (LayoutCRF), is proposed in [147] to handle partial occluded object based segmentation. A holistic-level representation of top-down constraints are integrated with the low-level features, such as color and texture, in [134] and numerically solved using simulated annealing. Borenstein et al [9], [8] construct a Bayesian model to integrate top-down and bottom-up information and the shape prior is obtained using multiple scale segmentation. The image fragments, or parts, are grouped together to produce a global approximation of the object’s shape. Ren et al. [113], [114] combine different levels of cues into the conditional random field (CRF) for object/background labeling. Levin et al. [84] also apply bottom-up and top-down cues into CRF and the training of the bottom-up and top-down cues is performed jointly. For the effort of including shape prior in the active contour models, Leventon et. al [82] applied principle component analysis (PCA) to obtain the training shape modes and presented them as signed functions in the geodesic active contour. Similarly, Dambreville et. al [29] proposed to apply Kernel PCA to boost the performance of shape prior in geodesic active contour model. Cremers et. al [27] proposed a more nature way to include the shape prior into the level-set scheme, where the transformation and rotation parameters will depend on the level-set function  $\phi$ . Shape symmetry information is included into level-set framework in [111], which can be used to segment objects with symmetric structure. Hong et. al [61] presented an integral kernel based representation of the shape prior and the corresponding non-rigid template-based segmentation is implemented using level-set.

Compared with previous work, the advantages of our algorithm can be summarized as follows: For the top-down information integration, we separate the stage into shape learning and non-rigid transformation estimation, which replace the eigenspace expansion of shapes with the shape modes. Similar as replacing Bayesian estimation with MLE, we decrease the complexity of shape learning in PCA related method, such as pre-alignment of the training shapes. Meanwhile, the in-class variations captured by the PCA method are addressed by fast non-rigid transformation estimation, which can be implemented efficiently. Furthermore, multiple objects and occlusion can be handled



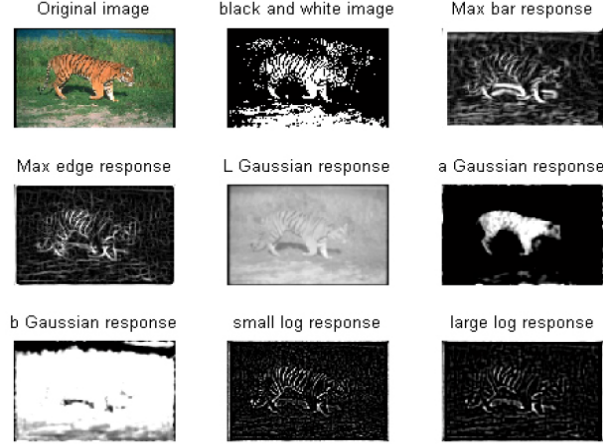


Figure 2.8: The filtering response using our revised *MR* filter bank.

by minimizing the proposed cost functions in our approach. For the bottom-up information integration, we are seeking a trade-off between accuracy and storage usage. A semi-parametric density approximation using adaptive mean-shift and  $L_2E$  robust estimation is proposed to represent the likelihood. The whole cues from both models and attributes are coupled into a variational framework and numerically implemented with level-set for seamless integration of prior contours. Our algorithm is easy to implement and computationally efficient with modest storage usage.

#### 2.4.1 Features and Shape Clustering

The *Luv* colors and textures are used to describe the bottom-up information which are extracted using a set of linear filters - filter banks. We revised the *MR* filter bank [140] to compute the filter response. The feature vector is composed with two *LoG* filter response on the *L* channel with  $\sigma = 1, 2$ , six Gaussian filtering response on the *L*, *u* and *v* channels with  $\sigma = 1, 2$  and the *MR* maximum bar and edge response on six different directions,  $\theta = 0, \pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6$ , with  $\sigma = 1, 2$  and 4. In total, each image pixel is represented by a 10 dimensional feature vector. Figure 2.8 shows the filtering results after applying the proposed filter bank to one tiger image.

For each human-delineated contours, we apply the elliptic Fourier descriptor (EFD) [74] (will be described in Section 3.1.3) to calculate the Fourier coefficients and use the

first 8 normalized harmonies. Each harmony contains four coefficients. An agglomerative clustering was applied on the set of the coefficients to find the clustering centers, which represent the shape modes. The advantages of using EFD are: 1) EFD is a good measure of global properties of shapes, which is preferred in our algorithm because small in-class variance is handled using non-rigid transformation estimation. This is explained in details in section 3.3; 2) The normalized EFD descriptions are invariant to rotation, translation and scaling.

### 2.4.2 Semi-density Approximation

Define  $\mathbf{f} \in \mathbf{F}$  in  $R^d$  as the random variable describing the feature vector, where  $\mathbf{F}$  is the set of all features and  $d$  is the dimension of the feature space. In order to build the connection between the image partition and contours, we define a partitioning operator  $\vartheta$  with  $\vartheta(I)$  describing the partition of image  $I$ . According to Bayesian rule, the segmentation can be modeled as the maximum a posterior (*MAP*) estimation

$$p(\vartheta(I)|f) \propto p(f|\vartheta(I))p(\vartheta(I)) \quad (2.27)$$

where  $p(f|\vartheta(I))$  is the likelihood of bottom-up features  $f$  given  $\vartheta(I)$ . The  $p(\vartheta(I))$  is the prior probability of the image partition which is used to model the shape constraints. The final  $\vartheta(I)$  can be solved by maximizing posterior probability  $p(\vartheta(I)|f)$ .

In order to calculate the posterior probability  $p(\vartheta(I)|f)$ , we need to compute the likelihood  $p(f|\vartheta(I))$ . Given the likelihood  $p(f|\vartheta(I))$  to be an arbitrary density function, kernel density estimation (KDE) is often used. The flexibility to represent any complicated probability density is KDE's major advantage. On the other hand, the high memory and computational complexity inhibit its practical usage.

We propose to model the likelihood  $p(f|\vartheta(I))$  using a semi-nonparametric method which is the linear combination of  $M$  Gaussian distributions. We show that density approximation using adaptive mean-shift and  $L_2E$  robust estimation performs almost as well as KDE using much less parameters. The semi-nonparametric Gaussian mixture model (GMM) is distinguished from the normal GMM because the mean, variance, weights and number of Gaussian distributions are not known in advance. The likelihood

is

$$p(\mathbf{f}|\vartheta(I)) = \sum_{i=1}^N \frac{\alpha_i \exp\left(-\frac{1}{2}(\mathbf{f} - \mathbf{f}_i)\mathbf{\Sigma}_i^{-1}(\mathbf{f} - \mathbf{f}_i)'\right)}{(2\pi)^{d/2} \left|\widehat{\mathbf{\Sigma}}_i\right|^{1/2}} \quad (2.28)$$

with  $i = 0, \dots, N$  denotes the total number of points in the image. The  $\alpha_i$  is the weight of the  $i$ th Gaussian,  $Normal(\mathbf{f}_i, \mathbf{\Sigma}_i)$ , and  $\sum_{i=1}^N \alpha_i = 1$ . Assume that after applying adaptive mean-shift based mode detection, there is totally  $M$  unique modes with weights  $\widehat{\alpha}_i$  as the proportion of the number of points in the  $i$ th cluster, where  $\widehat{\mathbf{f}}_i$  is the stationary mode point. The  $L_2E$  robust estimation can be used to estimate the variance  $\widehat{\mathbf{\Sigma}}_i$

$$L_2E(\widehat{\theta}_i) = \arg \min_{\widehat{\theta}_i = [\widehat{\mathbf{f}}_i, \widehat{\mathbf{\Sigma}}_i]} \left[ \varphi(\widehat{\theta}_i) \right] \quad (2.29)$$

where  $\varphi(\widehat{\theta}_i)$  is

$$\frac{m-2^{d+1}\pi^{d/2} \left|\widehat{\mathbf{\Sigma}}_i\right|^{1/2} \sum_{j=1}^m \phi(\mathbf{f}_j|\widehat{\mathbf{f}}_i, \widehat{\mathbf{\Sigma}}_i)}{2^d m \pi^{d/2} \left|\widehat{\mathbf{\Sigma}}_i\right|^{1/2}} \quad (2.30)$$

where  $j = 1 \dots m$  is the points in the  $i$ th cluster and  $\phi(\mathbf{f}_j|\widehat{\mathbf{f}}_i, \widehat{\mathbf{\Sigma}}_i)$  is defined as

$$\phi(\mathbf{f}_j|\widehat{\mathbf{f}}_i, \widehat{\mathbf{\Sigma}}_i) = \frac{\exp\left(-\frac{1}{2}(\mathbf{f} - \widehat{\mathbf{f}}_i)\widehat{\mathbf{\Sigma}}_i^{-1}(\mathbf{f} - \widehat{\mathbf{f}}_i)'\right)}{(2\pi)^{d/2} \left|\widehat{\mathbf{\Sigma}}_i\right|^{1/2}}. \quad (2.31)$$

The final density approximation can be written as

$$p(\mathbf{f}|\vartheta(I)) = \sum_{i=1}^M \frac{\widehat{\alpha}_i \exp\left(-\frac{1}{2}(\mathbf{f} - \widehat{\mathbf{f}}_i)\widehat{\mathbf{\Sigma}}_i^{-1}(\mathbf{f} - \widehat{\mathbf{f}}_i)'\right)}{(2\pi)^{d/2} \left|\widehat{\mathbf{\Sigma}}_i\right|^{1/2}} \quad (2.32)$$

where  $\sum_{i=1}^M \widehat{\alpha}_i = 1$ .

Instead of keeping all the  $N$  sets of parameters in (2.28), the likelihood  $p(\mathbf{f}|\vartheta(I))$  is modeled using  $M$  sets of parameters and  $M \ll N$ . We tested our density approximation method on many densities and it provides similar *pdf* to the KDE result using much less storage space. Figure 2.9 shows an example. The original six mixture Gaussian distributions and the modes detected by the adaptive mean-shift algorithm [48] are shown in Figure 2.9a and Figure 2.9b. Figure 2.9c is the standard kernel density estimation (KDE) using Epanetchnikov kernel. The density approximation using (2.29) and (2.32) is shown in Figure 2.9d.

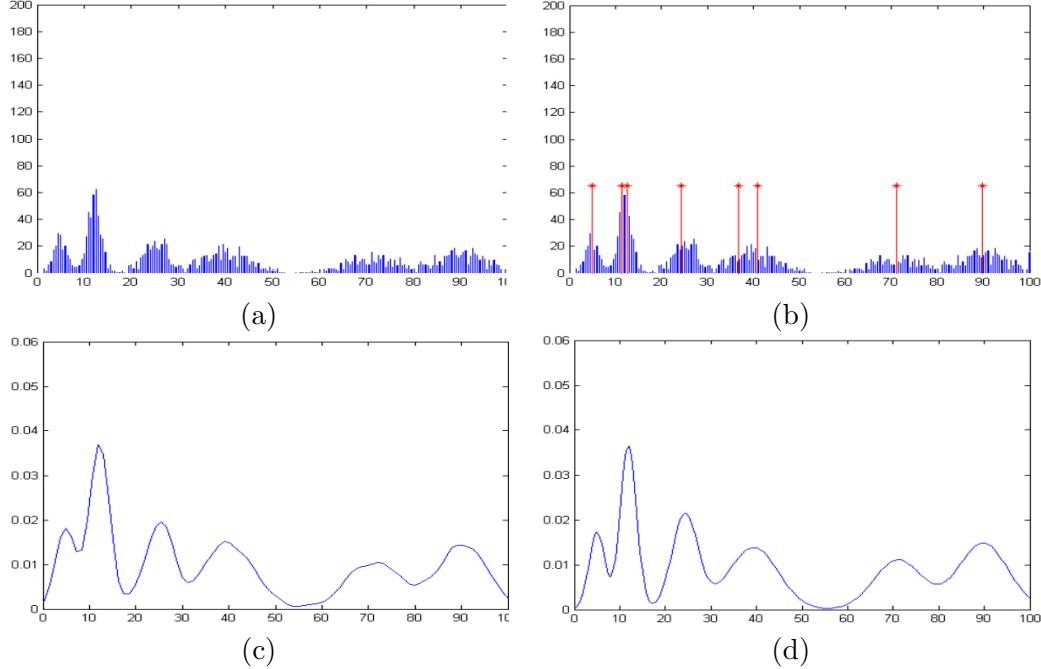


Figure 2.9: The standard KDE and our proposed semi-parametric density approximation results upon a six Gaussian mixture density.

### 2.4.3 Shape Alignment

Based on EFD clustering, we can obtain the shape modes for each object class. However, the shape modes can not be used directly in the level set framework without alignment. Therefore, a mapping  $T : \text{shape mode} \rightarrow \text{testing contour}$  is required to be calculated.

In order to find the transformation  $T$ , gradient decent is used in each iteration of the level set function in [17] and on the nested Euler-Lagrange equations in [27]. One common problem of these methods is the affine or even rigid transformation assumption of the mapping  $T$ . Instead, we generalize  $T$  to be the thin plate spline (TPS) transformation [20]. Affine transformation has proven to be a special case of TPS. Given two point sets, the TPS is estimated by minimizing

$$E_{TPS} = \sum_i \|T(w_i) - v_i\|^2 + \lambda I_f \quad (2.33)$$

where

$$I_f = \iint \left[ \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right] dx dy \quad (2.34)$$

with  $w_i$  denote the points on the training contours and  $v_i$  denote the points on the contours of the calculated likelihood map.

Because  $T$  is a nonrigid mapping which has infinite number of solutions for first term in (2.33), the smoothness term  $\lambda I_f$  is used to regularize the ambiguity. Each shape mode is matched to the contour of the likelihood map of the testing image by minimizing (2.33). The final mapping  $T$  is defined by the shape mode with minimal matching errors.

#### 2.4.4 The Variational Framework.

Define  $C$  as the contour to separate the object from the background, the  $p(C)$  describes the shape prior of the object. We calculate the non-rigid transforms  $T$  by minimizing (2.33) and use  $T(C)$  to denote the aligned shape prior. Define  $\vartheta(I_{x,y})$  as the labeling  $\{-1, 1\}$  of the  $x$ th column and  $y$ th row of the original image and  $\Omega_1/\Omega_2$  as the region in which  $I_{x,y}$  locates inside/outside the contour  $T(C)$ . Denote label  $\cdot^o$  as the object and  $\cdot^b$  as background and  $pos_{x,y}$  as the coordinates. Because the pixels inside contour  $T(C)$  has more chances to be labeled as object, the  $p(\vartheta(I))$  in equation (2.27) can be linked with  $p(C)$  using the softmax function and signed Chamfer distance

$$\begin{aligned} p(\vartheta^o(I_{x,y})) &= \frac{1}{1 + \exp(-dist(x, y, T(C)))}, \\ p(\vartheta^b(I_{x,y})) &= 1 - p(\vartheta^o(I_{x,y})) \end{aligned} \quad (2.35)$$

where

$$\begin{aligned} dist(x, y, T(C)) &= \min \|pos_{x,y} - T(C)\| \text{ on } \Omega_1, \\ dist(x, y, T(C)) &= -\min \|pos_{x,y} - T(C)\| \text{ on } \Omega_2 \end{aligned} \quad (2.36)$$

The maximization of equation (2.27) is equal to minimize its negative logarithm. Rewrite (2.27) in the energy integral equation and extending the approach in [14]

$$\begin{aligned} E(o, b, \phi) &= \gamma \int_{\Omega} |\nabla u(\phi(x, y))| dx dy \\ &\quad - \lambda_1 \int_{\Omega} u(\phi(x, y)) \log p(f|\vartheta^o(I)) dx dy - \lambda_2 \int_{\Omega} (1 - u(\phi(x, y))) \log p(f|\vartheta^b(I)) dx dy \\ &\quad - \lambda_3 \int_{\Omega} u(\phi(x, y)) \log p(\vartheta^o(I)) dx dy - \lambda_4 \int_{\Omega} (1 - u(\phi(x, y))) \log p(\vartheta^b(I)) dx dy \end{aligned} \quad (2.37)$$

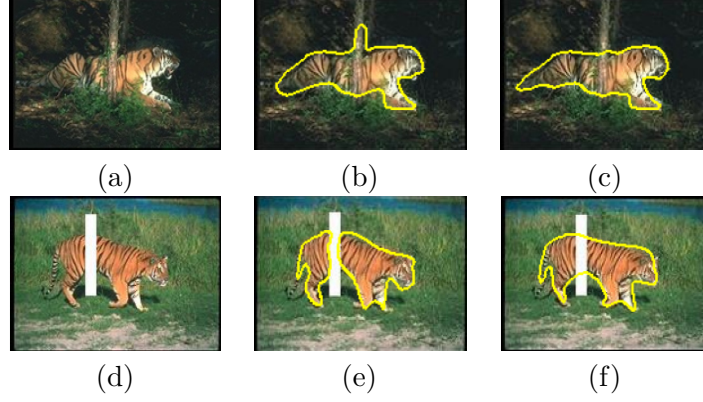


Figure 2.10: The occluded image segmentation results.

where the  $u(\phi(x, y))$  is the Heaveside step function. The  $\gamma \geq 0$  is used to adjust the effect of the length of contour  $C$ . The  $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$  are used to adjust the weights of the bottom-up information and the top-down shape constraints. The minimization is performed using the Euler-Largrange equation with the gradient of level set function  $\phi$  defined as

$$\begin{aligned} \frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} = & \delta(\phi) \left( \gamma \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \right) \\ & + \delta(\phi) (\lambda_2 \log p(f|\vartheta^b(I)) - \lambda_1 \log p(f|\vartheta^o(I))) \\ & + \delta(\phi) (\lambda_4 \log p(\vartheta^o(I)) - \lambda_3 \log p(\vartheta^b(I))) \end{aligned} \quad (2.38)$$

where  $\delta(\phi) = u'(\phi)$  and  $u(t) = \frac{1}{2} \left( 1 + \frac{2}{\pi} \arctan \left( \frac{t}{\varepsilon} \right) \right)$ . For all the experiments we are using  $\gamma = 0.01$  and all the  $\lambda$  are set to be 1.

#### 2.4.5 Experimental Results

We tested the performance of the algorithm on a set of tiger images taken from COREL database [11] and GOOGLE search. Two types of occlusions are shown here. In Figure 2.10a, the tree which blocked part of the tiger is mislabeled as object; the man-made white strip which divides the whole tiger into two parts is shown in Figure 2.10d. The segmentation results without shape priors are shown on Figure 2.10b and Figure 2.10e. Applying the shape priors calculated using EFD shape clustering and non-rigid transformation estimation, more accurate results are obtained in Figure 2.10c and Figure 2.10f for both cases. In total, we used 15 tiger images for likelihood density

approximation/shape clustering in the training stage and 20 partially occluded images for testing. The overall pixel-wise segmentation accuracy we obtained is 91.27%. Note that without applying shape prior the accuracy dropped to 85.01%.

## Chapter 3

### Robust Multiple Class Segmentation Using Mean-shift Patches

Region based segmentation, such as  $K$ -means, mean-shift [24], graph cut [124] and normalized cut [124], has been successfully applied in many applications. These methods treat image segmentation as a clustering or optimal grouping problem based on the low-level features. However, they only utilize the bottom-up information which makes it difficult to guarantee meaningful segmentation results. Recently, top-down prior knowledge has been combined with bottom-up features to improve the object-based segmentation results.

The pictorial structures model was proposed in [40] for visual object recognition. Kumar et al. [75] presented the OBJ CUT algorithm which applied the pictorial structure (PS) model and Markov Random Field (MRF) to segment objects from background. Borenstein et al [8] constructed a Bayesian model to integrate top-down and bottom-up information with the shape priors obtained from multiple scale segmentation. Orbanz et al [105] applied a nonparametric Bayesian model for image segmentation and used MRF as smoothing constraints. Levin et al. [84] integrated bottom-up and top-down cues into conditional random field (CRF) and the training was performed jointly.

As the number of classes increases, these specific models become complex both for training and parameter tuning. Winn [146] obtained good segmentation results with a simpler model using boosting on image appearance (texton histograms). Spatial weighting [93] and spatial pyramid [77] were used to improve the accuracy of the bag of keypoints model [126]. Rabinovich [110] proposed to treat the segmentation as optimal grouping problem and develop a model order selection schema to find the most stable



segmentation from a number of possible segmentations. All these methods suggest that a general framework can be suitable to perform multiple class object-based segmentation too.

In this chapter, using an idea similar to [60], [143], where the patches are used for outdoor scene labeling and video-cut, the segmentation problem is treated as an optimal grouping of *patches* in the image. A small group of pixels (patches) are labeled together to increase the robustness and decrease the running time. The traditional mean-shift algorithm [24] is used to combine image appearance [146] with the bag of keypoints model [126] for segmentation. We also propose a novel affine invariant representation of spatial co-occurrence of keypoints. The global shapes of objects are modeled using Elliptical Fourier Descriptor (EFD). All these features are combined in a unified framework to segment objects from a large number of classes. The contributions are:

- the appearance model and bag of keypoints are simple but surprisingly successful methods for generic object recognition. We demonstrate that for segmentation these two methods can be linked together over mean-shift patches to provide successful segmentation results too;
- the spatial relationship among the keypoints, which is disregarded in the traditional bag of keypoints model, are modeled using a novel and simple affine invariant descriptor;
- the algorithm we propose is much faster for both training and testing;
- the experimental results using three real datasets demonstrate that our algorithm provides satisfactory results.

In Section 3.1 we introduce the unified feature representation model. In Section 3.2 and Section 3.3, we describe our training method and the segmentation algorithm. Section 3.4 provides the experimental results and Section 3.5 concludes this chapter.

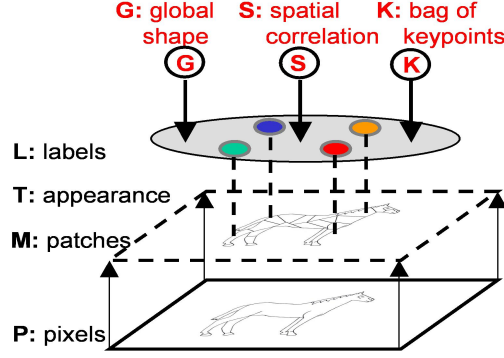


Figure 3.1: The graphic model of the segmentation framework.

### 3.1 Unified Feature Representation Model

In Figure 3.1, we represent the general model of our algorithm. The pixels **P** in the image are segmented using mean-shift algorithm to generate patches **M**. For each patch, the image appearance is represented with texton histogram **T**. Multiple hypothesis are generated based on appearance and refined by top-down information: through the bag of keypoints histogram **K**, the spatial correlation **S** (spatial keyton histogram) and the global shape **G** (EFD). The final label **L** is assigned to each patch considering all the cues.

#### 3.1.1 Mean-Shift Texton Histogram

In order to generate the mean-shift texton histogram, we first apply the five-dimensional mean-shift segmentation using two dimension for  $x, y$  coordinates and three dimensions for  $Luv$  color [24]. The parameters for all the datasets have spatial radius  $2h_s + 1 = 7$  and color radius  $2h_c = 7$ . The kernel we used is Epanechnikov kernel.

Texture and color are computed from the image through a set of linear filters using a modified MR filter bank [140]. The feature vector is composed of two *LoG* filter responses on the  $L$  channel ( $\sigma = 2, 4$ ), six one-dimensional Gaussian filter responses on the  $L, u$  and  $v$  channel ( $\sigma = 2, 4$ ) and the maximum bar and edge responses on six different directions between  $0$  and  $5\pi/6$  and three different variances ( $\sigma = 1, 2, 4$ ). In total, each image pixel is represented by a 10 dimensional feature vector. All the filter responses obtained from the training set are put together and clustered using  $K$ -means

to build the texton library.

For the traditional histogram-based segmentation, the windowed texton histogram is used to model the image appearance of the training set. Instead, we computed the texton histograms for each mean-shift patch separately

$$h(i) = \sum_{j \in M} \text{count}(T(j) = i) \quad (3.1)$$

where  $M$  denotes the mean-shift patch,  $i$  is the  $i$ th element of the texton histogram,  $T(j)$  returns the texton assigned to pixel  $j$ . The advantages of applying the texton histogram over mean-shift patches are:

- mean-shift patches take the edges into the consideration;
- the number of mean-shift patches is much smaller, which decrease the complexity for training and classification;
- visually similar and spatially close pixels are grouped together and given the same label, which is a more natural approach than arbitrary square windows;
- mean-shift patches provide a natural link between the appearance and the bag of keypoints model.

Although this method may still mislabel a whole mean-shift patch based on appearance, we will show that the top-down information in our framework helps to correct these types of errors.

In Figure 3.2, an example is shown where mean-shift patches provide more distinctive information than an arbitrary square window. The original image (top-left) is processed with mean-shift (top-right). The single mean-shift patch (middle-left) is represented by texton histogram (bottom-left), where a  $31 \times 31$  square window (middle-right) has a completely different texton histogram (bottom-right). The classification based on these two texton histograms labeled the mean-shift patch as airplane, but the square window patch as sky.

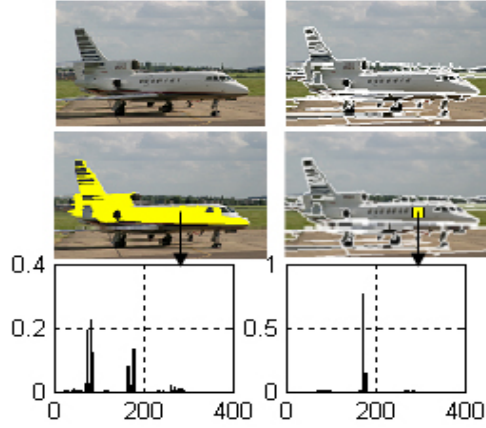


Figure 3.2: A mean-shift patch and a  $31 \times 31$  square patch and their corresponding texton histograms. See text.

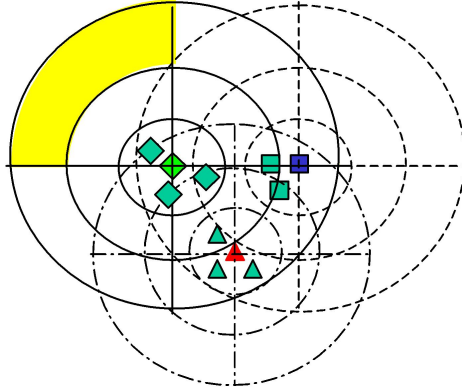


Figure 3.3: An illustration of spatial keyton histogram, where we only show  $r = 1, 2$ , and  $3$  in relative distance and  $\theta = 0, \frac{\pi}{2}, \pi$  and  $\frac{3\pi}{2}$ . The yellow region is the  $10^{th}(2 \times 4 + 2, r = 3 \text{ and } \theta = \frac{\pi}{2})$  bin  $(10, \diamond)$  of the coordinate system with the position of the center  $\diamond$  as the origin.

### 3.1.2 Spatial Keyton Histogram

Given a training image, the Harris corner detector is applied on the *gray-level* image to detect the interesting points. Affine invariant features are extracted from the neighborhood of the detected points. We choose the 20 dimensional moment invariants [51] as keypoint descriptors because of their low dimensionality and satisfactory performance. Although SIFT features [89] are shown to be superior to other local descriptors for recognition [96], it did not provide better results in our segmentation experiments. This observation is also shown in [104, p.81] for Fergus et al. dataset [41].

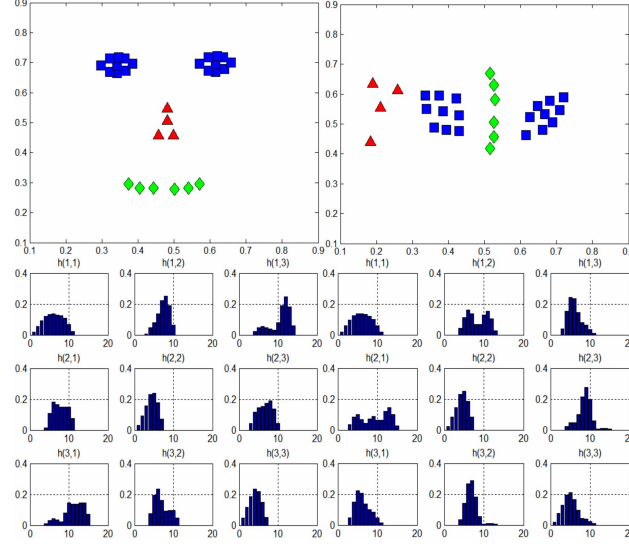


Figure 3.4: Two spatial keyton histograms  $h(i, j)$  with three keytons  $\square$ ,  $\triangle$  and  $\diamond$  denoting the 1st, 2nd and 3rd keyton. Affine transformations are performed on the keypoints belong to each keyton separately in the upper-left to produce the upper-right configuration. Nine  $h(i, j)$  are shown for each of the two configurations.

The descriptors of all the keypoints in the training set are put together and  $K$ -means clustering is used to build the dictionary of the cluster centers. Similar to the definition of textons, we call the cluster centers of keypoints as "keytons". Each object in the training image is represented by a histogram,  $h(i)$ , of the keytons calculated from the keyton dictionary. Each keypoint is assigned to its closest keyton based on the Euclidean distance

$$h(i) = \sum_{j \in O} \text{count}(O(j) = i) \quad (3.2)$$

where  $O$  denotes the set of keypoints of a given object in the image. The  $O(j)$  returns the keyton assigned to the  $i$ th keypoint.

Although bag of keypoints model, which we call the keyton histogram, has been successfully used for object classification in many applications [93], [126], it has been shown that recognition accuracy is increased by considering the spatial correlation of keypoints [93], [78]. We propose a novel spatial keyton histogram to model the spatial configuration of keypoints.

The Figure 3.3 shows a set of keypoints which belong to three different keytons, noted as  $\square$ ,  $\triangle$  and  $\diamond$ . For each keypoint in the image, we separate the image plane into

co-centered circles using this keypoint position as the origin. In general, the range of  $r$  is from 1 to 5 in relative distance, and  $\theta$  has four values. Assume the  $i$ th keyton has  $N_i$  keypoints, then the spatial keyton histogram  $h(i, j)$  is defined as the average of the  $j$ th keytons spatial distribution relative to all the keypoints of the  $i$ th keyton. It can be calculated as

$$h(i, j)^k = \frac{1}{N_i} \sum_{m=1}^{N_i} \sum_{j \in \text{bin}(k, m)} \text{count}(j) \quad (3.3)$$

where  $k$  denotes the  $k$ th bin of  $h(i, j)$  and it is defined from  $r = 1$  and  $\theta = 0$  to  $r = 5$  and  $\theta = 3\pi/2$ , a total of 20 values. The  $\text{bin}(k, m)$  is the  $k$ th bin of the coordinate system with  $m$ th keypoint as the origin (refer to Figure 3.3). A four-tap Gaussian smoothing filter is used to postprocess the histograms. The proposed spatial keyton histogram is quasi affine invariant.

In Figure 3.4, we show two sets of data where each keyton has the same number of keypoints but different spatial configurations (upper part of Figure 3.4). They can not be separated by the bag of keypoints model, but have different spatial keyton histograms (lower part of Figure 3.4). For each configuration, note that the affine invariant is shown in  $h(1, 1)$ ,  $h(2, 2)$ ,  $h(3, 3)$  and the different spatial relationships between  $\square$ ,  $\Delta$  and  $\square$ ,  $\diamond$  are captured in  $h(1, 2)$ ,  $h(2, 1)$  and  $h(1, 3)$ ,  $h(3, 1)$  in bottom-left and bottom-right of Figure 3.4.

### 3.1.3 Global Shape Model

The global shape model is the top-down constraint used to group the image patches into real objects. The pictorial structure model and PCA was used to encode the appearance and represent the shape in a joint density for object recognition [41]. Instead of using complex shape model, the Elliptic Fourier Descriptor (EFD), which was shown to be successful in [23], is chosen to model the global shape of the objects. There are several reasons to use EFD:

- the EFD has a simple histogram-like representation. In our algorithm we use the first 32 ( $4 * 8$ ) coefficients;
- the normalized EFD is invariant to rotation, translation and scaling;

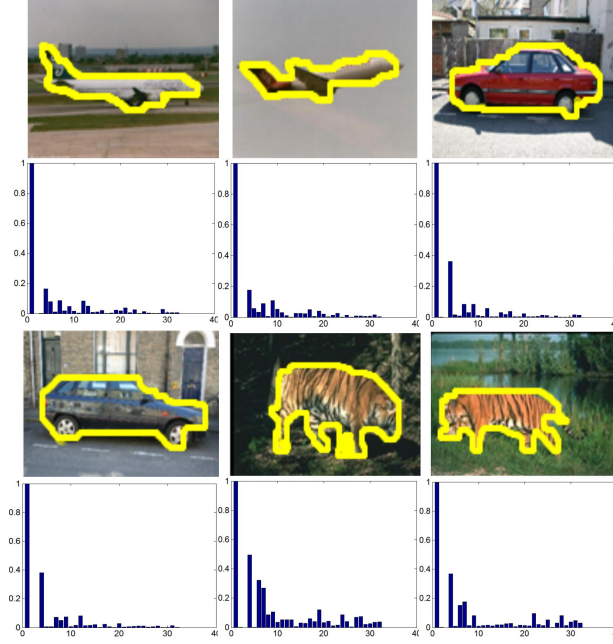


Figure 3.5: Examples of the Elliptical Fourier Descriptors (EFD). The contour is superimposed on original image.

- the close contour reconstructed from EFD is always closed.

EFD is the Fourier expansion of the chain coding. Assume we have  $M$  points on the close contour. Following the approach of Kuhl and Giardina [74], the EFD coefficients of the  $n$ th harmonic are:

$$\begin{aligned}
 a_n &= \frac{S}{2n^2\pi^2} \sum_{i=1}^M \frac{\Delta x_i}{\Delta s_i} \left[ \cos \frac{2n\pi s_i}{S} - \cos \frac{2n\pi s_{i-1}}{S} \right] \\
 b_n &= \frac{S}{2n^2\pi^2} \sum_{i=1}^M \frac{\Delta x_i}{\Delta s_i} \left[ \sin \frac{2n\pi s_i}{S} - \sin \frac{2n\pi s_{i-1}}{S} \right] \\
 c_n &= \frac{S}{2n^2\pi^2} \sum_{i=1}^M \frac{\Delta y_i}{\Delta s_i} \left[ \cos \frac{2n\pi s_i}{S} - \cos \frac{2n\pi s_{i-1}}{S} \right] \\
 d_n &= \frac{S}{2n^2\pi^2} \sum_{i=1}^M \frac{\Delta y_i}{\Delta s_i} \left[ \sin \frac{2n\pi s_i}{S} - \sin \frac{2n\pi s_{i-1}}{S} \right]
 \end{aligned} \tag{3.4}$$

where  $s_i$ ,  $S = \sum_{j=1}^{i(M)} \Delta s_j$ ,  $\Delta s_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}$ ,  $\Delta x_i = (x_i - x_{i-1})$ ,  $\Delta y_i = (y_i - y_{i-1})$ . The  $\Delta x_i$  and  $\Delta y_i$  are the changes in the  $x$  and  $y$  projection of the chain code at the  $i$ th contour point.

Figure 3.5 shows the 32 EFD coefficients for six images. It was found that the first  $4 * 8$  EFD coefficients already contain enough information for separating the objects

**Input:** Given a test image  $x$  and the training histogram dictionaries  $\mathbf{T}$ ,  $\mathbf{K}$ ,  $\mathbf{S}$  and possible  $\mathbf{G}$ . The number of classes is  $L$ ,  $l = 1, \dots, L$ .

- Apply Harris corner detector and extract the moment invariant descriptors from the gray level image.
- Calculate the mean-shift patches and represent the likelihoods of each patch as  $p^t(j, l)$ ,  $p^k(j, l)$  and  $p^s(j, l)$  where  $j = 1 \dots J$ , the number of patches in image  $x$  is  $J$ .
- For  $j = 1 \dots J$ 
  - Build texton histogram  $\mathcal{X}^t(j)$ .
  - For  $l = 1 \dots L$ : calculate  $svm_l^t(\mathcal{X}^t(j))$  using (5.3.3) and the negative outputs of  $svm$  are forced to be 0.
  - For  $l = 1 \dots L$ : if  $svm_l^t(\mathcal{X}^t(j)) > 0$ , use (3.8) to calculate appearance likelihood  $p^t(j, l)$ .
- For  $l = 1 \dots L$ 
  - Record the patches satisfied  $p^t(j, l) > 0$  as set  $J'$ , generate hypothesis  $H(l) = \bigcup_{j=1}^{J'} M(j)$ .
  - Collect all the keypoints inside  $H(l)$  and compute the keyton histogram and compact spatial keyton histogram  $\mathcal{X}^k(l)$ ,  $\mathcal{X}^s(l)$ .
  - Using (5.3.3), calculate  $svm_l^k(\mathcal{X}^k(l))$  and  $svm_l^s(\mathcal{X}^s(l))$ .
  - For all patches  $j \in J'$ , compute  $p^k(j, l)$  and  $p^s(j, l)$  using (3.8).
- For  $j = 1 \dots J$

$$label(j) = \arg \max_l \log (p^t(j, l) * p^k(j, l) * p^s(j, l)) \quad (3.5)$$

Figure 3.6: Segmentation algorithm and testing procedure.

into different classes.

### 3.2 Training Procedure

The training set contains images which were *manually* segmented into different objects. All the features of the training images are put together and  $K$ -means are used to generate the texton library. The exact value of  $K$  will be given in the experimental section. The mean-shift patches are generated from each training image. The texton histograms are calculated for each patch and saved in the training texton histogram dictionary. In our model each histogram corresponds to one mean-shift patch.

The training images are transformed into gray-level and the Harris corner detector is applied to detect the keypoints. The moment invariants are extracted and a keyton



library is constructed using  $K$ -means. For each training image, the keyton histogram is computed to record the frequency of occurrence of each keyton. In order to save space and computation time, we are using a more compact way to record the spatial keyton histogram.

Assume we have  $n = 1 \dots N$  training images and each image contains keypoints from  $k = 1 \dots M_n$  keytons. The spatial keyton histogram  $h(i, j)$  of each training image contains  $M_n * M_n$  histograms. For all training images, we get  $\sum_{n=1}^N M_n * M_n$  histograms. All the histograms are clustered using  $K$ -means with  $K$  equal to 50 and the clustering centers are recorded. For each training image, we assign each histogram of  $h(i, j)$  into its closest clustering center. The number of the histograms for the  $n$ th training image is decreased from  $M_n * M_n$  to 50. This compact spatial keyton histogram represents the patterns of spatial arrangement among keypoints in the training image.

The last step for training is the shape modeling using EFD. For each training image, we extract the contours of the object from the masks and represent each contour using the first 32 EFD coefficients. Note that the shape training based on EFD is optional because only certain type of objects have discriminative contour information. Cars, airplanes, and tigers, etc. have distinct contour shape but this is not the case for sky, water, etc. After the EFD coefficients of all the exemplars contours are calculated, a simple agglomerative clustering is applied over EFD descriptors to find the clustering centers for each class of objects.

### 3.3 Segmentation Algorithm and Testing

In this section we will explain the segmentation algorithm and testing procedure shown in Algorithm 3.6. The **T**, **K**, **S**, **G** represent the four training dictionaries: texton, keyton, compact spatial keyton, and EFD shape descriptors (refer to Figure 3.1). Given a test image, each detected keypoint is assigned to a mean-shift patch based on its spatial coordinates. Because some keypoints may locate on the borders of the patches, we inflate each patch with four pixels. For each mean-shift patch  $j$ , the  $label(j)$  is the final label of this patch. The  $p^t(label(j) = l|l)$ ,  $p^k(label(j) = l|l)$  and  $p^s(label(j) =$

$l|l)$  are used to describe the likelihoods given label  $l$  based on texton, keyton and compact spatial keyton histogram similarities with the three dictionaries  $\mathbf{T}$ ,  $\mathbf{K}$ ,  $\mathbf{S}$ . For abbreviation we write them as  $p^t(j, l)$ ,  $p^k(j, l)$ ,  $p^s(j, l)$  and define a set  $fv = \{t, k, s\}$ .

For classification we are using the nonlinear support vector machine (SVM) with a Mercer kernel [43]. Using the training dictionaries  $\mathbf{T}$ ,  $\mathbf{K}$  and  $\mathbf{S}$ , we train a  $svm_l^{fv}$  for each class  $l$ . The SVM decision function in kernel formulation is

$$svm_l^{fv}(x) = \sum_i y_i \alpha_i \kappa(x, x_i) + b \quad (3.6)$$

where  $\kappa$  is the Mercer kernel defined as  $\kappa(h_1, h_2) = \exp(-\frac{1}{\tau} \chi^2(h_1, h_2))$ . The  $x_i \in \{\mathbf{T}, \mathbf{K}, \mathbf{S}\}$ ,  $y_i \in \{-1, +1\}$  are the training samples and their labels. The  $\alpha_i$  and  $b$  are the learned weights and learned threshold. The  $\tau$  is the mean value of  $\chi^2$  distances between 100 pairs randomly selected training histograms. The  $\chi^2$  distance is

$$\chi^2(h_1, h_2) = \frac{1}{2} \sum_{t=1}^P \frac{(h_1(t) - h_2(t))^2}{h_1(t) + h_2(t)} \quad (3.7)$$

where  $h_1$  is the histogram of test patch and  $h_2$  is a member of  $\{\mathbf{T}, \mathbf{K}, \mathbf{S}\}$  with  $P$  denoting the dimension. Multiple object labels are assigned to the  $j$ th test patch with probabilities calculated using the positive raw output of (5.3.3)

$$p^{fv}(j, l) = \frac{svm_l^{fv}(\mathcal{X}^{fv}(j))}{\sum_{l=1}^L svm_l^{fv}(\mathcal{X}^{fv}(j))} \quad (3.8)$$

where  $\mathcal{X}^{fv}(j)$  is the histogram of test patch  $j$ .

Based on texton histogram, the appearance likelihood  $p^t(j, l)$  for each patch  $j$  is calculated first. Then multiple hypotheses are generated using the keypoints located in set  $J'$  (please refer to Algorithm 3.6 for definition). The likelihoods  $p^k(j, l)$  and  $p^s(j, l)$  are computed using (3.8). This procedure form a loop until all the hypotheses are scored. The final  $label(j)$  for each mean-shift patch  $j$  is decided by maximizing the sum of the log likelihoods.

Some segmented objects can be refined using EFD shape descriptors. This step is completely unsupervised. Only if the proposed objects contain distinctive contours, which are known in the training stage, the EFD descriptors are applied for shape matching.

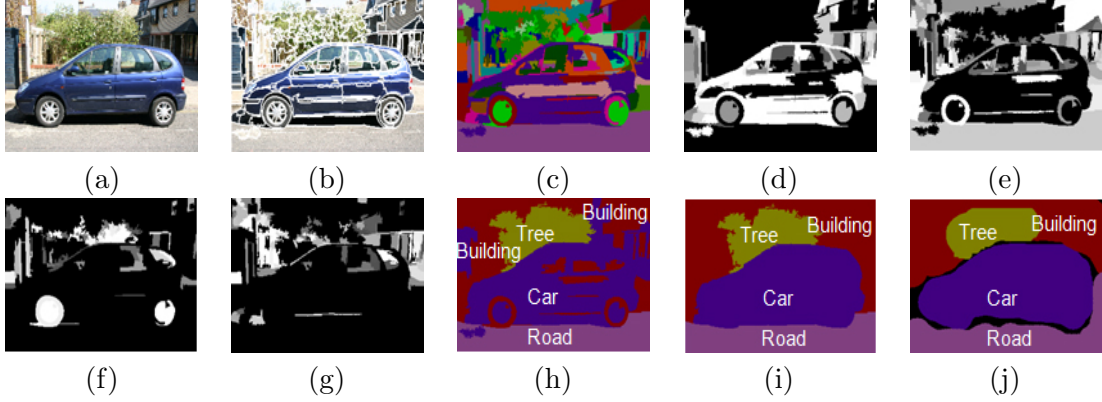


Figure 3.7: A testing example. (a) The original test image. (b) The mean-shift segmentation results. (c) The object labeling using appearance only. Different colors corresponding to different objects. (d)-(g) Four hypothesis for car, building, chair and cow. The brighter intensity means higher probability. (h) The object labeling which maximizes the sum of the log likelihood. (i) The refined segmentation result using the global shape of the car. (j) The hand-draw segmentation by a human.

Table 3.1: The segmentation accuracy for MHMS 11 database.

Plane	Car	Tiger	Zebra	Grass	Road
73.2	73.9	90.1	88.6	94.5	93.4
Water	Sky	Forest	Rock	Building	<b>Overall</b>
72.7	88.1	89.3	78.2	71.9	<b>88.9</b>

Figure 3.7 shows one complete procedure of segmentation. We generate several hypotheses and marked each patch by maximizing the likelihoods obtains from appearance and keypoints. The final result is refined using global shape information. We also show in Figure 3.7j the hand-drawn segmentation result rendered by a human.

### 3.4 Experiments

We used real images to test our algorithm on three different datasets:

- our MHMS 11 which is composed of images taken from Caltech101 [39], COREL [11] and Google search;
- the Sowerby 7 dataset [57], [125];
- the MSRC 21 dataset [125].

	Building	Grass	Tree	Cow	Sheep	Sky	Plane	Water	Face	Car	Bike	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Building	63.1	2.5	3.1			3.8	9.5	0.5		7.3	3.8		1.7				1.1				3.6
Grass	0.2	97.9	1.6				0.2												0.1		
Tree	3.5	4.4	89.5	0.2	0.3	0.4				0.8	0.5	0.3									
Cow		13.5	1.3	65.7	5.3						2.4			5.1				3.5	3.1		
Sheep		10.9	3.4	7.9	54.1					3.4				4.6		5.4		3.7	6.6		
Sky	5.8	0.3	1.2			86.2	2.4				2.1						1.7				0.3
Plane	11.3	2.8				3.5	62.7	2.1		5.1	3.7		3.5	2.1							3.1
Water	3.2					3.4	5.3	70.9		3.6			2.9	3.6			4.1				2.9
Face	2.7	0.1	0.1					0.2	83.2	2.1			1.1		3.8					6.7	
Car	7.5		2.9			3				70.5			1.8	7.2			3.1				3.9
Bike	7.5		2.1			2.4					79.6		3.2		0.9		4.3				
Flower		5.1		10.1								71.3		6.7				6.8			
Sign	25.6				6.9					11.3	3.6		37.9		1.1	5.1					8.5
Bird	3.1	15.6	6.5	4.9	8.2	7.8	9.1	8.9						23.2				7.1	5.4		
Book	5.9			2.5					0.1	0.4			3.1		87.9						
Chair	4.8	30.4		7.2							5.6		7.9	4.1	5.4	23.2	3.1	3.1	2.6		2.5
Road	2.4					1.1		1.9		3.8	2.4					0.1	88.2				
Cat	3.5	3.4	2.9	11.3	12.9									0.4			3.4	33.1	29.1		
Dog	1.2	3.9	5.3	7.5	10									2.9			10.5	21.4	34.1	3.2	
Body	7.1	3.5	2.1				0.5		22.9			0.9		5.7			3.4	4.5	6.1	43.2	
Boat	17.4		3.5			3.6	11.5	19.8		3.1	2.5		5.1	1.1							32.4

Figure 3.8: The confusion matrix for the MSRC dataset with row labels as inferred class and column as ground true class.

All the datasets contain multiple objects with different viewpoints, illuminations and scales. For all the experiments we chose only **40%** of the images for training and the remaining **60%** for testing. For the Sowerby image database, in order to get enough sampling from the objects that only shown in part of the images, such as car and road marking, we manually selected the training images that do contain these objects.

**MHMS 11:** This dataset contains 100 color images of  $192 \times 128$  and *eleven* different classes. We use mean-shift segmentation with minimum region size of 100. The dimension of the texton library and keyton library is  $11 * 50$  and  $11 * 100$ . The segmentation accuracy of each class is shown in Table 3.1. The overall pixelwise segmentation accuracy is 86.0%. For class plane, car, tiger and zebra in this dataset, global shape prior is applied and found to be useful. It increases the segmentation accuracy by 2.7%.

**Sowerby 7:** This dataset [57] contains 104 color images of  $96 \times 64$  and *seven* different classes. We use mean-shift segmentation with minimum region size of 40 because of the smaller image size. The dimension of the texton library and keyton library is  $7 * 50$  and  $7 * 100$ . The overall pixelwise segmentation accuracy we obtained is 88.9%. Compared with the results reported in [57] 89.5%, and [125] 88.6%, where the highest percentage is obtained using about half for training and context depended information. We have only used 40% images for training and did not consider context

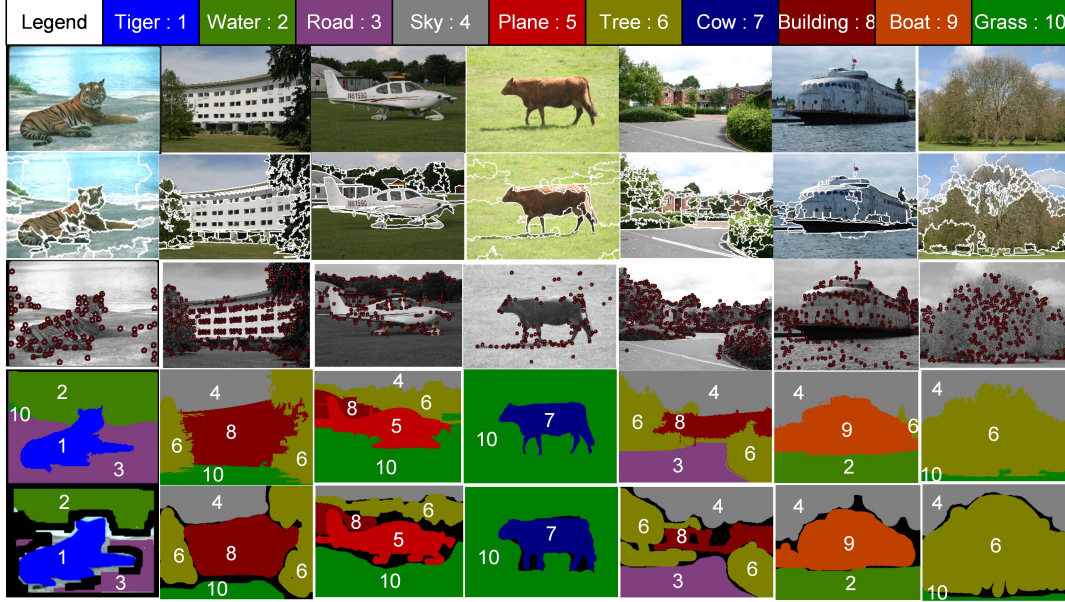


Figure 3.9: Some segmentation results using our algorithm. The first row is the original image. The second row is the mean-shift segmented patches. The third row is the Harris corner detector results with red circles marking the keypoints. The fourth row is the labels provided by the algorithm. The fifth row is the hand-draw label by human.

information.

**MSRC 21:** This is one of the most complete multiple object database for segmentation. This dataset contains 592 color images of  $320 \times 213$  and *twenty-one* objects. We use mean-shift segmentation with minimum region size of 150. The dimension of the texton library and keyton library is  $21 * 50$  and  $21 * 100$ . The overall pixelwise segmentation accuracy we obtained is 75.1%, which is higher than the accuracy reported in the literature [125] 72.7%. Our algorithm also provides higher segmentation accuracy for 18 classes out of 21 classes than [125], which are marked in grey in Table 3.8. When the segmented objects contain car, airplane, tree, face or sign, EFD shape descriptors are used to refine the labeling. However, because of the inter-class variability of this database, especially in some cases there exist multiple objects which belong to the same class in one image, global shape prior doesn't provide big improvement. Figure 3.9 provides some segmentation results.

One of the major advantages of our method is the speed. As each mean-shift patch is labeled once, we save the time. A  $320 \times 213$  image can be processed less than 1 minute

with a P3 1.5G Hz processor with 1G RAM using the MATLAB implementation. It can be much faster using C++.

### 3.5 Discussion

It is worth analyzing why this simple framework works well for multiple class object segmentation. From our research we conclude that interleaved recognition and segmentation might increase the accuracy for both tasks. Mean-shift patches provide a natural link between recognition and segmentation with the reduction of the computational time as a valuable side benefit. The keyton histogram, coupled with the spatial keyton histogram, gained benefits from the bottom-up appearance information. However, mean-shift segmentation itself can provide errors. It is also possible that the proposed method make mistakes for visually similar objects. The ambiguity of sharing features between different classes makes the generic object segmentation a very difficult problem.

## Chapter 4

### Robust 4D Segmentation Using Prediction and Collaborative Models

The 3D Echocardiography is one of the emerging diagnostic tools among modern imaging modalities for visualizing cardiac structure and diagnosing cardiovascular diseases. Ultrasound imaging is real-time, noninvasive and less expensive than CT and MR. However, ultrasound normally produces noisy images with poor object boundaries.

Recently, the problem of unsupervised detection, segmentation and tracking of heart chambers have received considerable attention [30], [52], [62], [68], [166]. Among these applications, segmentation and tracking of the left ventricle (LV) have attracted particular interests. It provides clinical significance for radiologists to evaluate disease, such as acute myocardial infarction. Several difficulties exist compared with traditional 2D tracking algorithms:

- The computation demand is much higher for 3D volumetric data.
- Feature point based tracking has difficulty for ultrasound images because they lack reliable landmarks.
- In order to provide a practical diagnosis tool for radiologists, a tracking algorithm should be able to recover from failures.

The widely used 2D tracking algorithms [162] won't provide good results if directly applied on 3D ultrasound tracking applications. In order to achieve robust tracking in 3D ultrasound which is characterized by low image quality, learning based detector and boundary classifiers are used to track the LV boundary in each frame. This tracking by detection strategy can avoid accumulating errors and is proven to be quite effective in recent literature [5], [86], [165]. However, it still has several problems:

- The boundary classifiers are sensitive to initial positions [25] and good initializations have to be provided because we can not exhaustively search all the possible configurations in the whole 3D volume. Considering the speed, the current search range is constrained on the normal directions within  $\pm 12$  mm.
- Tracking by detection applies an universal description of the objects without considering any temporal relationship, which leads to temporal inconsistency between adjacent frames.

In this chapter, we propose a fast and novel unsupervised 4D ultrasound segmentation algorithm which address these difficulties [155], [154]. The contributions are:

- We propose a novel one-step forward prediction using motion manifold learning, which respects the geodesic distances of both the shape and motion of the heart. The motion priors can provide good initial positions for the boundary classifiers.
- A collaborative 3D template tracker is applied to erase the temporal inconsistency introduced by 3D detection tracker.
- A rectangle filter is used to reject outliers and achieve robust data fusion. Smooth boundary tracking is obtained by projecting the tracking points in each frame to the constrained shape boundary.
- The algorithm can process a 3D volume, e.g.,  $160 \times 144 \times 208$  voxels, in less than 1.5 seconds and we obtain subvoxel tracking accuracy.

In Section 4.1 we introduce the Bayesian tracking framework. In Section 4.2 and Section 4.3, we describe the learning methods and tracking algorithm. Section 4.4 provides the experimental results.

## 4.1 Bayesian Tracking Framework

Define  $\mathbf{x}_t$  as the true position of each 3D boundary point of the left ventricle (LV) at time  $t$  and let  $\mathbf{z}_t$  to represent the measurement. The tracking problem can be formulated



as an estimation of A posterior probability  $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ , where  $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$  represents the past  $t$  measurements. Sequential Bayesian tracking based on Markovian assumption is performed recursively in a *prediction*

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1} \quad (4.1)$$

and *updating* step

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}). \quad (4.2)$$

Bayesian tracking assumes that the following densities are known. The  $p(\mathbf{x}_0)$  denotes the distribution of the 3D LV surface points in the first frame. In our algorithm  $p(\mathbf{x}_0)$  is automatically calculated using the trained detector and boundary classifiers. The  $p(\mathbf{x}_t|\mathbf{x}_{t-1})$  represents the motion prior (state model) and is predicted for the next frame. The  $p(\mathbf{z}_t|\mathbf{x}_t)$  represents the measurement density.

If all the densities in (4.1) and (4.2) are Gaussian, Bayesian tracking can be formulated as the Kalman filter, otherwise a particle filter can be applied. The CONDENSATION algorithm [65] can be used to sample the posterior probability for single object, and Markov Chain Monte Carlo (MCMC) [71] sampling can be used for multiple objects tracking. Both Kalman filtering and particle filter assume a Markov model, which only considers the previous state  $\mathbf{x}_{t-1}$  to estimate the density of current state  $\mathbf{x}_t$ . The  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  is therefore equal to  $p(\mathbf{x}_t|\mathbf{x}_{t-1})$  in (4.1) and is often modeled using a predetermined distribution.

In real tracking problems, the motion prior (state model)  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  may not follow a Markovian assumption and it could be of any form. In our algorithm, we model the  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  to be dependent on both  $\mathbf{x}_{t-1}$  and  $\mathbf{z}_{1:t-1}$ . One-step forward prediction using motion manifold learning is applied to estimate this motion prior.

For the measurement densities  $p(\mathbf{z}_t|\mathbf{x}_t)$ , we select two collaborative trackers: the detection tracker and the template tracker, which can mutually benefit each other. The detection tracker can discriminate the 3D target from background in low image quality and noisy environment. The template tracker respects the local image information and preserves the temporal consistence between adjacent frames. The trackers are modeled

as  $r_k$ ,  $k = 1$  for detection tracker and  $k = 2$  for template tracker, then

$$p(\mathbf{z}_t|\mathbf{x}_t) = p(\mathbf{z}_t|\mathbf{x}_t, r_1)p(r_1) + p(\mathbf{z}_t|\mathbf{x}_t, r_2)p(r_2). \quad (4.3)$$

Substituting (4.3) in (4.2) and replacing  $p(\mathbf{x}_t|\mathbf{x}_{t-1})$  with  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  in (4.1), the final posterior probability  $p(\mathbf{x}_t|\mathbf{z}_{1:t})$  is obtained from the robust data fusion and the  $\mathbf{x}_t = \arg \max_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{z}_{1:t})$ .

## 4.2 Motion Learning and Classifiers Training

Each training sequence contains a heart motion cycle which starts from the end-diastolic (ED) phase, passes through the end-systolic (ES) phase and comes back to ED. The training procedure is in a batch mode where all the annotated sequences are used at the same time. It contains three steps. First the motion modes are learned using manifold learning and hierarchical K-means. Next, an ED detector is trained to locate the position of the object in the first frame. Finally, two boundary classifiers (one for ED and one for ES) are trained using the annotated sequences to delineate the boundary.

### 4.2.1 Learning the Motion Modes

**Motion Alignment Using 4D Generalized Procrustes Analysis.** Our training sequences contain 11-25 time frames and 16 frames are chosen to resample each motion sequence. In this way we generate 4D motion vectors containing the same dimensionality  $d$ , where  $d = N_f \times 3 \times 16$ ,  $N_f = 289$  is the number of boundary points and three represents  $x$ ,  $y$  and  $z$  dimensions.

Generalized procrustes analysis (GPA) is used to align all resampled motion vectors to remove the translation, rotation and scaling [32, ch. 5]. However, the shape differences and motion patterns are still preserved. After the 4D GPA, these aligned motion vectors are decomposed into separated 3D shapes. All the following learning steps are performed on the aligned 3D shape vectors.

**Motion Manifold Learning.** Because the actual number of constraints that

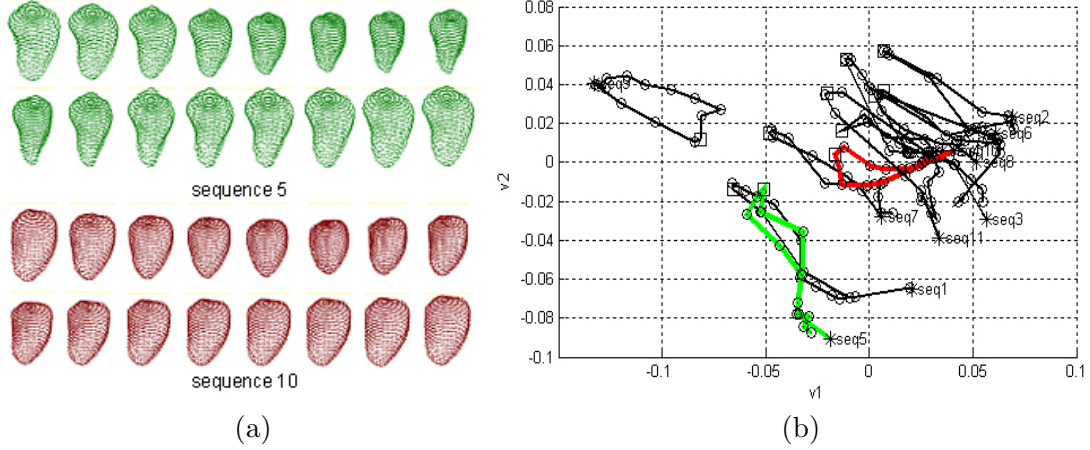


Figure 4.1: Manifold embedding for LV motion patterns. (a) Two LV boundary mesh sequences. (b) The 11 sequences embedded in a 2D subspace. Note: The end diastolic (ED) phase has larger volumes and represented as stars in (b), while the end systolic (ES) phase has smaller volumes and represented as squares in (b).

control the LV motion are much less than its original dimensionality, the aligned 3D shape vectors lie on a low-dimensional manifold. Given the whole set of 3D training shape vectors,  $M = \{\mathbf{m}_0, \dots, \mathbf{m}_i, \dots, \mathbf{m}_n\}$  where  $\mathbf{m}_i \in R^d$ , there exists a mapping  $F$  which represents  $\mathbf{m}_i$  in the low-dimensions as

$$\mathbf{m}_i = F(\mathbf{v}_i) + \mathbf{u}_i \quad i = 1, 2, \dots, n \quad (4.4)$$

where  $\mathbf{u}_i \in R^d$  is the sampling noise and  $\mathbf{v}_i \in R^q$  denotes the original  $i$ -th shape  $\mathbf{m}_i$  in the low-dimensional subspace. We set  $q = 2$  because the higher dimensions did not provide additional accuracy in our experiments. The nonlinear mapping  $F$  is the transformation from the low-dimensional subspace to the original space.

We apply ISOMAP [130] to embed the nonlinear manifold into a low-dimensional subspace. We start by finding the seven closest neighbors (seven was found to be the best) of each point  $\mathbf{m}_i$  in the original space  $R^d$  and connect the neighbors to form a weighted graph  $G$ . The weights are calculated based on the Euclidean distance between each connected pair of vectors. We then calculate the shortest distance  $d_G(i, j)$  between any pair of points  $\mathbf{m}_i$  and  $\mathbf{m}_j$  in the graph  $G$ . The final step is to apply the standard multiple dimensional scaling (MDS) to the matrix of graph distance  $\{d_G(i, j)\}$ . In this way, the ISOMAP applies a linear MDS on the local patch but preserve the geometric

distance globally using the shortest path in the weighted graph  $G$ .

Figure 4.1a shows two annotated LV motion sequences. Figure 4.1b shows several LV motion representations in a low-dimensional subspace. An interesting but expected observation is illustrated in Figure 4.1b. The LV motion is almost periodic because one cycle of heart beat starts from ED and returns to ED. In total we applied manifold learning on 36 annotated LV motion sequences. In order to make the figure readable, we only show 11 sequences in Figure 4.1b.

**Hierarchical K-means Clustering.** Given all the motion cycles shown on the embedded subspace, we applied a hierarchical K-means to learn the motion modes. First, we apply K-means on all the training ED shapes and  $K$  is taken as four. (Four was chosen using a pilot experiment.) After this step, we align all motion sequences in one group by moving their ED vectors to their cluster center. In this way we cancel the translation among the shapes in one subgroup, but focus on the difference in motion patterns. Each aligned sequence is transformed to a  $2 \times 16$  dimensional vector, where two represents the reduced dimensionality and 16 represents the number of frames. K-means is applied again within each group and the cluster center in the 32 dimensional space corresponds to a motion mode. In this step the  $K$  is decided by evaluating the difference between the cluster center and each vector within the group. Each motion mode is a weighted sum of all sequences that are clustered into the same group. The weights are proportional to their Euclidean distance from the cluster center. The geodesic distance in the original manifold is modeled by Euclidean distance in the embedded low-dimensional subspace.

#### 4.2.2 Learning the ED Detector

In this step we train a 3D detector to locate the pose of LV in the motion sequence. We first calculate a mean shape by averaging all the LV in the ED frames of the annotated training sequences. A principal component analysis (PCA) shape space is calculated for all the ED shapes at the same time. In order to automatically initialize the tracker, we need to find the similarity transformation from the mean shape to the LV in the ED frame for each sequence. Discriminative learning based approaches have

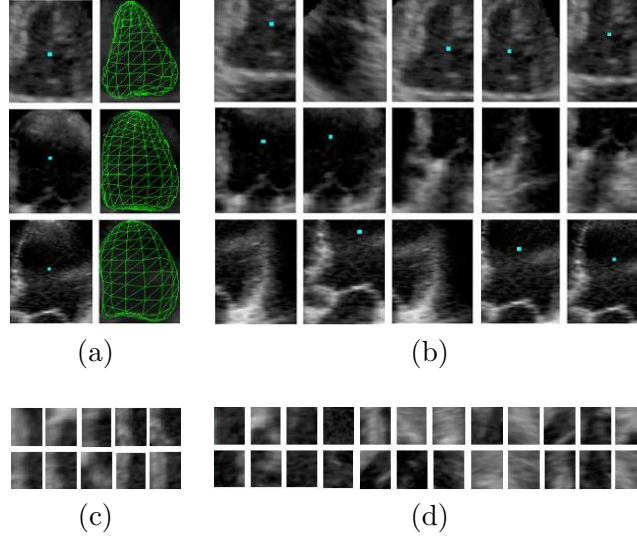


Figure 4.2: The positive (a), (c) and negative samples (b), (d) used for training. (a) and (b) Training samples for the detector. (c) and (d) Training samples for the the boundary classifiers.

proven to be efficient and robust for 2D object detection [142]. The object is found by scanning the classifier over an exhaustive range of possible locations, orientations and scales in an image. However, it is challenging to extend them to 3D objection detection problems since the number of hypotheses increases exponentially with respect to the dimensionality of the parameter space. As the posterior distribution is often clustered in a small region, it is not necessary to search the parameter space exhaustively.

Marginal space learning (MSL) [166], [6] is used to learn an ED detector to locate LV in the first frame efficiently. The idea for MSL is to incrementally learn classifiers on projected marginal spaces. We split the estimation into position detection, position-orientation detection and full similarity transformation detection. The MSL reduces the number of testing hypotheses by six orders of magnitude in our applications, which makes the directly training of the detector on 3D volumetric data a feasible procedure.

### 4.2.3 Learning the Boundary Classifiers

After we obtain the pose of the LV in the ED frame, we need to segment its boundary to automatically start the trackers. Active shape models (ASM) [25] is used to deform an initial estimate of a nonrigid shape. The non-learning boundary classifier

using gradients in the original ASM does not work in our application due to the complex background and weak edges in 3D ultrasound. Learning based methods can exploit image evidences to achieve robust boundary detection. Boundaries with different orientation are usually detected on prerotated images [31]. Since 3D volume rotation is very time consuming, we use steerable features for boundary detection and avoid rotating the 3D volume. For each boundary point (289 in total), we sample several points from the volume around it under a special pattern, which embed the orientation information into the distribution of the sampling. A few local features for each sampling point, such as voxel intensity and gradients, etc., are calculated. The advantages of steerable features are that they combine the advantages of both local and global features.

Two boundary classifiers, one for LV motion close to the ED phase and the other for LV motion close to ES, are trained using probabilistic boosting tree (PBT) [135]. The PBT ensembles many strong classifiers into a tree structure. The widely used cascade boosting [142] can be treated as a special case in PBT. The learned boundary classifiers are used to automatically segment the boundary of LV, to initialize the trackers, and also used as the detection tracker for the following frame. In Figure 4.2 we show some positive and negative training samples used for training both the detector and the boundary classifiers.

### 4.3 Tracking Procedure

The tracking procedure on a testing sequence contains four steps. The  $p(\mathbf{x}_0)$  is initialized using the learned ED detector and the ED boundary classifier. At time  $t - 1$ , registration based reverse mapping and one-step forward prediction are used to estimate the next state  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$ . We then apply two collaborative trackers and robust data fusion to estimate the measurement density  $p(\mathbf{z}_t | \mathbf{x}_t)$ . In order to obtain smooth tracking of LV, each boundary point is mapped to a shape constrained 3D boundary. The final results are obtained by maximizing the posterior probability in (4.2). These *prediction* (4.1) and *updating* (4.2) steps are performed recursively for each frame in one sequence.

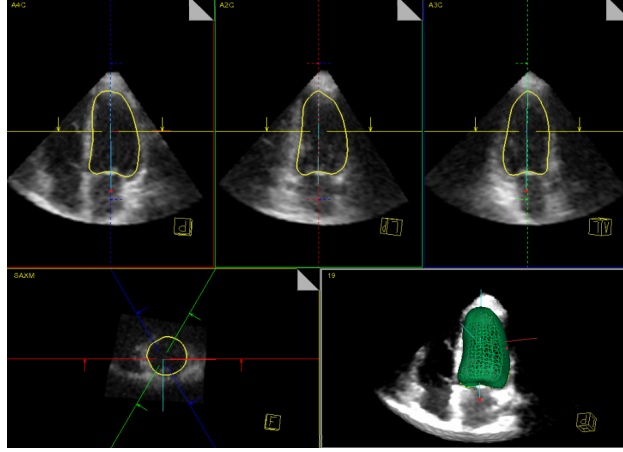


Figure 4.3: The four canonical view and 3D representations of the segmentation result (automatic tracking initialization) of LV.

#### 4.3.1 Initialization of Tracking

In order to initialize the boundary tracking of LV in an unsupervised manner, we need to automatically detect and segment LV in the ED frame of a testing sequence. Given the ED frame, all positions are scanned by the trained position detector. The top 100 candidates  $(x_i, y_i, z_i)$ ,  $i = 1, 2, \dots, 100$  are kept. Each candidate is then expanded using 1000 hypothesis on orientations  $(\psi_{ij}, \phi_{ij}, \theta_{ij})$ ,  $j = 1, 2, \dots, 1000$ . The trained position-orientation detector is applied for each candidate and the best 50 are kept. These 50 candidates are scaled using 1000 hypothesis  $(sx_{kl}, sy_{kl}, sz_{kl})$ ,  $k = 1, 2, \dots, 50$ ,  $l = 1, 2, \dots, 1000$ . Evaluated by the position-orientation-scale detector, the best 100 candidates are averaged to produce the final similarity transformation estimation.

After the similarity transformation is calculated, the LV mean shape is registered and superimposed on the ED frame of the testing sequence as the initial position. For each boundary point we search  $\pm 12$  mm range on the normal directions of the boundary. (The value  $\pm 12$  mm was set considering both speed and accuracy.) The learned boundary classifier is used to move each boundary point to its optimal position where the estimated boundary probability is maximized. Figure 4.3 shows the tracking initialization result.

### 4.3.2 One-Step Forward Prediction

In this step we calculate the motion prior (state model)  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  using the learned motion modes. At time  $t-1$ , we first transform the current 3D shape  $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$  to the corresponding frame of *each* motion mode in the 4D GPA coordinate system. Thin plate spline (TPS) transformation [87] is applied to perform this mapping. The TPS is a nonrigid transformation between two 3D point sets. The transformation  $T$  contains an affine mapping plus a wrapping coefficient matrix. We used 72 uniformly sampled points from 289 boundary points to estimate the TPS transformation  $T$  by minimizing

$$E_{TPS}(T) = \sum_{i=1}^{72} \|\mathbf{w}_i - T(\mathbf{b}_i)\|^2 + \lambda f(T) \quad (4.5)$$

where  $\mathbf{w}_i$  denote the 3D mesh point on the learned motion modes and  $\mathbf{b}_i$  denote the points on the testing object's boundary. The  $f(T)$  is a function containing a kernel which represents the internal structure relationship of the point set. The regularization parameter  $\lambda$  is chosen as 1.5. We refer the readers to [87] for more details.

The prediction is applied on the motion mode which minimizes the previous 1 to  $t-1$  *accumulated* TPS registration errors

$$I = \arg \min_i \sum_{j=0}^{t-1} \min_T E_{TPS}(\mathbf{x}_j, \mathbf{q}_{i,j}, T) \quad (4.6)$$

where  $i = 1, \dots, Q$  represents the number of motion modes. The  $\mathbf{q}_{i,j}$  is the  $i$ -th motion mode in the  $j$ -th frame and the  $E_{TPS}(\mathbf{x}_j, \mathbf{q}_{i,j}, T)$  is the registration error.

After we found the correct motion mode  $\mathbf{q}_{I,t}$  using (4.6), the final prediction result in the real world coordinate system of the  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  is obtained using the reverse mapping  $T^{-1}$ . A motion mode generated by reverse mapping using TPS is shown in Figure 4.4.

There could be motion mode changes during the prediction when the prediction starts from one motion mode and jumps to another mode during tracking. This corresponds to the LV motion which starts from an ED shape in one learned motion mode, but has a motion trajectory close to another mode. This is the reason we apply the *accumulated* TPS registration error based one-step forward prediction. The algorithm



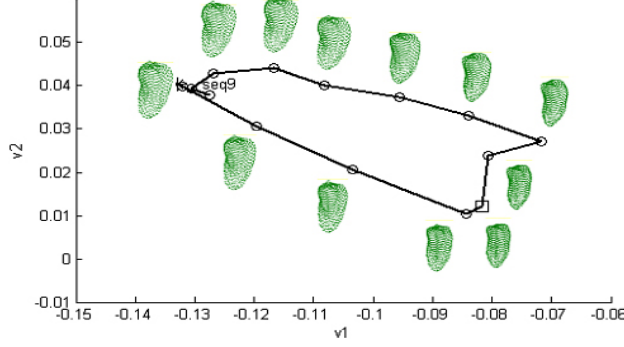


Figure 4.4: The LV boundaries in 3D world coordinates of a motion mode. The results are calculated using TPS reverse mapping and superimposed in the two-dimensional reduced subspace.

provides accurate motion prior for boundary tracking.

### 4.3.3 Collaborative Trackers

Given the motion prior  $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$  learned using one-step forward prediction on the motion manifold, for each boundary point the learned boundary classifiers are used to search in its  $\pm 12$  mm range on the normal direction. The optimal position is found by maximizing the boundary probability. The ED boundary classifier is used when the frame index is close to ED and the ES boundary classifier is used when it is close to ES. The final position using *detection* tracker is obtained by maximizing  $p(\mathbf{z}_t|\mathbf{x}_t, r_1)$  in (4.3).

In order to compensate the disadvantages of detection tracking mentioned in the introduction, a 3D *template* tracker is also applied. Given  $\mathbf{x}_t$  a 3D boundary point and its neighborhood  $N(\mathbf{x}_t)$ , let  $G(\mathbf{x}_t, \mu)$  denotes the transformation of the template. (The neighborhood was chosen to be a  $13 \times 13 \times 13$  cube based on experiments.) The goal is to search the best transformation parameters which minimize the error between  $N(\mathbf{x}_t)$  and  $G(\mathbf{x}_t, \mu)$ .

$$\mu = \arg \min_{\mu} \sum_{\mathbf{x}_t \in N(\mathbf{x}_t)} [G(\mathbf{x}_t, \mu) - N(\mathbf{x}_t)]^2. \quad (4.7)$$

Because there is only a small change of parameter  $\mu$  between adjacent frame, the

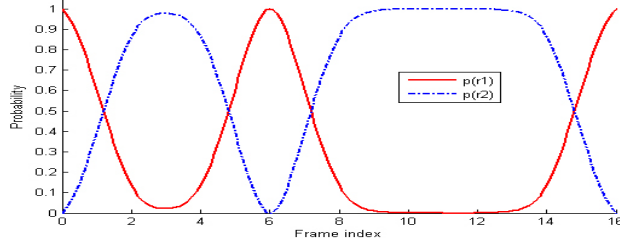


Figure 4.5: The prior  $p(r_1)$  for detection tracker (red solid line) and  $p(r_2)$  for template tracker (blue dotted line). The ED phase has frame index zero and ES phase is around frame six.

minimization of (4.7) can be solved by linearizing the expression

$$G(\mathbf{x}_t, \mu) = G(\mathbf{x}_t) + \frac{\partial G(\mathbf{x}_t, \mu)}{\partial \mu} d\mu. \quad (4.8)$$

At the end the result  $p(\mathbf{z}_t | \mathbf{x}_t, r_2)$  is obtained.

Although the template matching algorithm is not robust and only works under the assumption of small inter-frame motions, it preserves temporal consistence and its disadvantages can be compensated by the one-step forward prediction and the detection tracker. Template updating is a key issue in template tracking. If we update the template in each frame only based on the previous template tracking result, the error will be inevitably accumulated and finally results in the template drifting mentioned in [15], [94]. Generally it is difficult for template tracking to recover from drifting without special processing. In our method we update the template using the previous collaborative tracking result. Because the learned motion prior is enforced and detection is used, this updating scheme can help the template tracker to recover from the template drifting. As shown in [107], [138], learned motion prior is quite effective to help tracking to recover from failures.

#### 4.3.4 Data Fusion

Fusion of the collaborative tracking is obtained by defining prior distribution  $p(r_1)$ ,  $p(r_2)$  in (4.3). Based on domain expert's knowledge, both priors were designed as the exponential functions of  $t$ , which is illustrated in Figure 4.5. We show only one heart beat cycle which contains 17 frames. In order to reject outliers and achieve robustness, we apply a rectangle filter of  $[-12 \ 12]^3$  mm on the final data fusion results to erase

the motion replacements which are larger than this size between adjacent frames. The corresponding position of the eliminated boundary point is recalculated based on the bicubic interpolation of its neighbors. After this step we obtained the  $p(\mathbf{z}_t|\mathbf{x}_t)$ .

#### 4.3.5 Postprocessing and Projection

Due to speed considerations, the detection tracker is designed to search on the normal direction of the boundary, and the template tracker is searching along the direction of the gradients. Both of them can not provide smooth tracking results. Let  $B_f$  denotes the 3D boundary point-set after the data fusion step. We project  $B_f$  onto the PCA shape space calculated from the training stage, and obtain a smooth boundary point set  $B_s$ . A surface of the smooth boundary  $B_s$  is constructed using the 3D triangulation. Each boundary point  $P$  on  $B_f$  is projected onto the smooth surface by finding the triangle  $S \in T = \{\text{all triangles of } B_s\}$  which minimize the square distance

$$\text{dist}(s, t) = [S(s, t) - P]^2 \quad (4.9)$$

where  $S(s, t) = \mathbf{b} + s\mathbf{e}_0 + t\mathbf{e}_1$  with  $(s, t) \in D = \{(s, t) : s \in [0, 1], t \in [0, 1], s + t \leq 1\}$ . The  $\mathbf{b}$  is one vertex in the triangle and  $\mathbf{e}_0$  and  $\mathbf{e}_1$  represent two edges. In this way, we keep the tracking results and achieve smooth motion tracking of the LV.

### 4.4 Experimental Results

We collected 67 annotated 3D ultrasound LV motion sequences. The 4D  $(x, y, z, t)$  motion sequences contain from 11 to 25 3D frames. In total we have 1143 ultrasound volumetric data. Our dataset is much larger than those reported in the literature, e.g., 29 cases with 482 3D frames in [68], 21 cases with about 400 3D frames in [106] and 22 cases with 328 3D frames in [167].

The imaging protocols are heterogeneous with different capture ranges and resolutions along each dimension. The dimensionality of 27 sequences is  $160 \times 144 \times 208$  and the other 40 sequences is  $160 \times 144 \times 128$ . The  $x, y$  and  $z$  resolution ranges are  $[1.24, 1.42]$ ,  $[1.34, 1.42]$  and  $[0.85, 0.90]$  mm. In our experiments, we randomly select 36 sequences for training and the rest are used for testing.

Table 4.1: The *point-to-mesh* (PTM) errors measured in millimeters using three tracking algorithms.

	Mean	Variance	Median
3D optical flow	2.68	1.28	2.39
Tracking by detection	1.61	1.24	1.31
Collaborative trackers	<b>1.28</b>	<b>1.11</b>	<b>1.03</b>
	Min	Max	80%
3D optical flow	0.94	10.38	3.23
Tracking by detection	0.59	9.89	1.89
Collaborative trackers	<b>0.38</b>	<b>9.80</b>	<b>1.47</b>

The accuracy is measured by the *point-to-mesh* (PTM) error. All 3D points on each frame of the testing sequence are projected onto the corresponding annotated boundary of the test set. The projection distance from the point to boundary is recorded as the PTM error,  $e_{ptm}$ . For a perfect tracking, the  $e_{ptm}$  should be equal to zero for each 3D frame. In Table 6.2, we compared the quantitative  $e_{ptm}$  using our proposed algorithm, with tracking by 3D optical flow [33] and tracking by detection [165].

The 80% column in Table 6.2 represents the sorted 80% smallest error of all  $e_{ptm}$ , and is commonly used by doctors to evaluate the usability of the system. For example, if the doctor can tolerate an error of 1.5 mm, they normally expect 80% of the errors to be smaller than this number. The mean  $e_{ptm}$  we obtained is 1.28 mm with a 80% error below 1.47 mm. These are the best results in the literature as far as we know (mean error 1.45 mm in [167], 2.5 mm in [99] and 2.7 mm in [106]). Considering the range of resolution in the test set, we actually obtained subvoxel tracking accuracy on the mean  $e_{ptm}$ .

The systolic-diastolic function is the volume-time curve which represents continuous LV volume change over the time  $t$ . It is an important diagnosis term to evaluate the health condition of the heart. In Figure 4.6, we illustrate one heart cycle of two systolic-diastolic functions. The curves for all three tracking algorithms and the ground-truth annotation are shown. Our algorithm (Tr. collab.) provides the most similar functions to the ground truth curves.

Two types of errors are frequent in tracking LV in 3D ultrasound. The first type is the leakage error,  $e_{leakage}$ , which happens on the mitral valve region. This is introduced

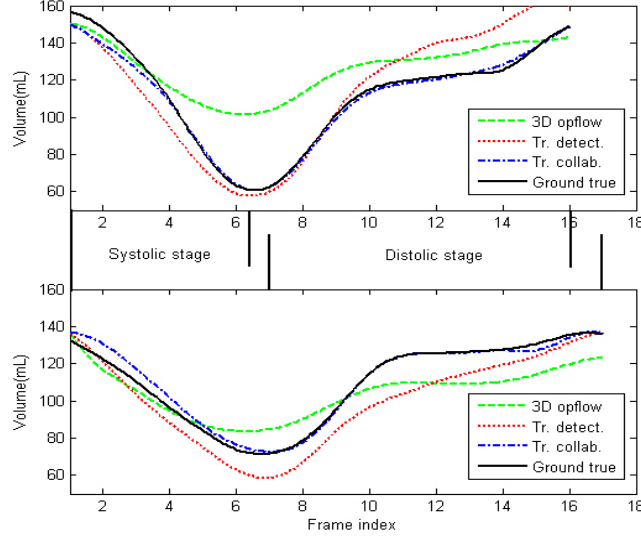


Figure 4.6: Two volume-time curves demonstrate the whole cardiac cycle, which includes the systole stage and the diastole stage.

by the similar appearance of the mitral valve to the LV boundary. A good LV boundary tracking algorithm should not follow the motion of the leaflets of the mitral valve. Tracking by detection failed on frame 8 (row 3, columns 3 and 4 in Figure 4.7b) because it always searches for what it learned in the training stage, but ignores all the local image information and temporal consistency.

The second type is the shrinkage error,  $e_{shrink}$ , which happens on the apex region. The 3D optical flow failed on the frame 6 (row 2, columns 5 and 6 in Figure 4.7c) because of the low image quality around the apex region. This proves that motion prior is necessary to obtain enough shrinkage for LV tracking in the 3D echocardiography because of the low quality of ultrasound imaging.

Using our algorithm, shown in Figure 4.7a, none of the errors are observed. The most important practical consideration for 3D tracking is computational complexity. One of the major reasons to propose marginal space learning, steerable features, registration based reverse mapping and one-step forward prediction is the speed. Our currently C++ implementation requires  $1 - 1.5$  seconds per frame containing  $160 \times 148 \times 208 = 4,925,440$  voxels and about 20 seconds for the whole sequence. Our implementation is at least two times faster than the slice-cut algorithm presented in [62], even if they are not working on 3D volumetric data directly, and about hundreds of times faster than

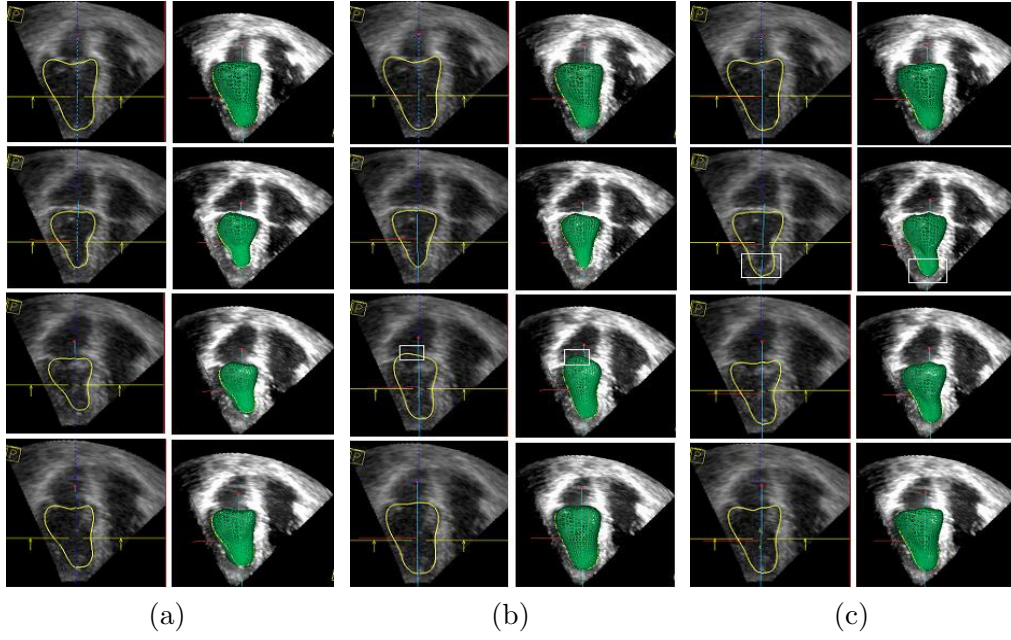


Figure 4.7: Comparative tracking results on a testing sequence with 12 frames. (a) Tracking results using the proposed method. (b) Tracking by detection. (c) Tracking using 3D optical flow. The rows correspond to frame index 1, 6, 8 and 10.

[99] which reported using a MATLAB implementation.

## Chapter 5

### Multiple Class Object Recognition in Medical Images

Breast cancer is a malignant neoplasm that can affect both women and men. It is the leading cancer in both white and African American women, with more than 178,480 new cases for an estimated to be diagnosed in 2007 where 2030 cases are men, and will be responsible for estimated 40,460 deaths [3]. It is the second most common cause of cancer death in white, black, Asian/Pacific Islander and American Indian/Alaska Native women [3], [137]. The incidence of breast cancer among women has increased gradually from one in 20 in 1960 to one in eight today. At this time there are slightly over 2 million breast cancer survivors in the United States. Women living in North America have the highest rate of breast cancer in the world [3].

In spite of the increase in the incidence of the disease, the death rates of breast cancer continue to decline. This decrease is believed to be the result of earlier detection through screening and analysis as well as improved treatment [3]. Some of the common methods screening of breast cancer include examination by a physician, self-performed routine examinations and routine mammograms. When suspicious lesions are detected by these methods, a fine needle aspirate or biopsy can be performed and the obtained tissue is examined [116]. The extracted tissue is mounted on glass slides and examined by a surgical pathologist to make decision of benign or cancer. Diaminobenzidine (DAB) and hematoxylin are standard staining methods used for breast cancer.

There has been increasing interest in investigating computer assisted diagnosis (CAD) system to help breast cancer diagnosis, e.g. [128]. However, to our knowledge, most of these studies were relatively limited in scale because the computation is often a bottleneck. In this chapter, we report a Grid-enabled framework for analyzing imaged breast tissue specimens [150], [151]. We discriminate between benign and cancer

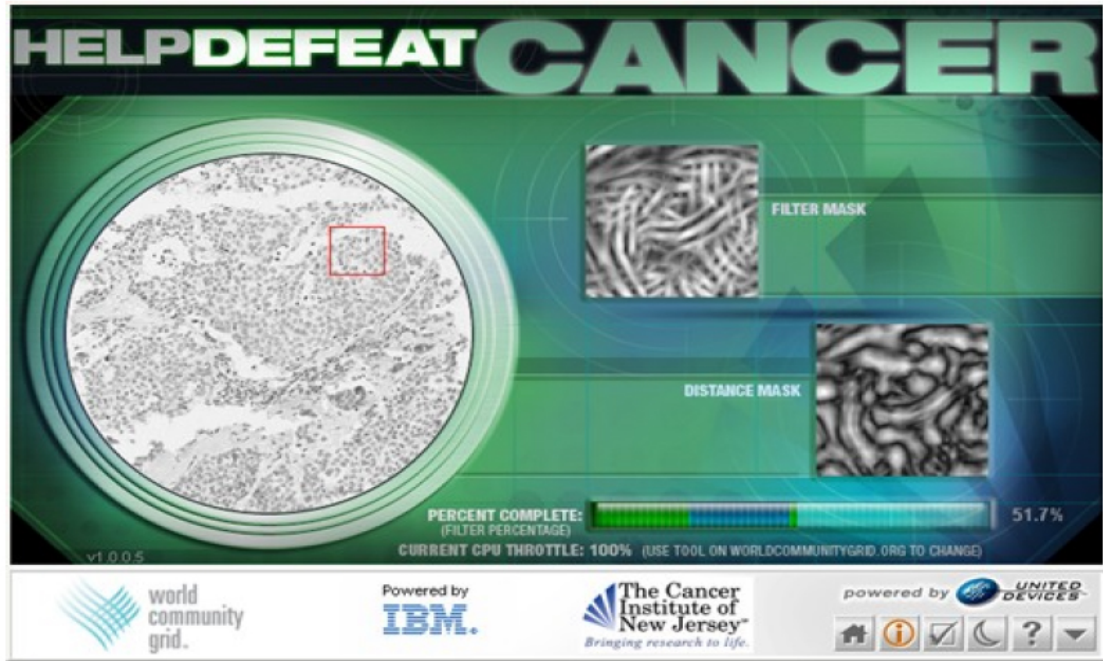


Figure 5.1: A screenshot of Help Defeat Cancer (HDC) client running on IBM world community grid (WCG).

breast tissues using the results obtained from the Grid. Without using the Grid, the filtering and generation of universal texture library would require 210 days of computation for 3744 samples because of the large image size and computational complexity of the process.

In Section 5.1 we introduce the IBM World Community Grid (WCG). In Section 5.2 and Section 5.3, we describe the data generation and analysis procedures. Section 5.4 provides the experimental results.

## 5.1 IBM World Community Grid

We recently undertook a collaborative project with IBM, the "Help Defeat Cancer" (HDC) [63] project, which enabled us to utilize the massive computational power of the World Community Grid (WCG) to analyze the breast cancer specimens. A screenshot of one of the thousands of distributed client computers participating in the HDC is shown in Figure 5.1. IBM World Community Grid [64] (WCG) is a philanthropic project which utilizes otherwise unused CPU cycles from personal computers around



the world and aggregates the combined computational power. WCG was established to address challenging large scale non-profit research projects which can benefit humanity. It takes advantage of otherwise wasted energy and at the same time creates a virtual supercomputer that by some measures exceeds the capacity of traditional supercomputers. The result is that some otherwise impractical or intractable research projects can be brought to successful completion. Investigators can submit a research proposal for consideration by the WCG project committee. If approved by the advisory board, the project is run at no cost to the research team. Findings are subsequently placed in the public domain. Suitable research areas include, but are not limited to biomedical, climatology, environment, conservation and emergency preparedness.

WCG enabled the most computationally intensive components of the Help Defeat Cancer (HDC) project to run at optimal speed, thereby increasing the accuracy and sensitivity with which expression calculations and pattern recognition procedures were conducted. By harnessing the collective computational power of WCG, we were able to analyze a larger set of cancer tissue specimens than what would be possible using traditional computer resources. This added level of speed and sophistication led to improved capacity to detect subtle changes in measurable parameters, and prognostic clues which are difficult to observe by visual inspection alone.

Imaged pathology specimens were generated using a high-throughput whole slide scanner and transferred from laboratories within Robert Wood Johnson Medical School (RWJMS) and The Cancer Institute of New Jersey (CINJ) to the secure Boulder Colorado IBM hosting site where World Community Grid servers reside. As results were computed, they were returned to the servers at RWJMS and CINJ. The total of the transfers approached one terabyte of data. About 2909 years of run-time in the form of slightly more than 5 million work packages were harvested from the personal computers contributed to World Community Grid. This includes an approximate 3 times redundancy of work to ensure that the computations were not in error or tampered with. Because of the fairly large working set memory required for the program, only machines with over 1 GB of RAM were selected to run the project.

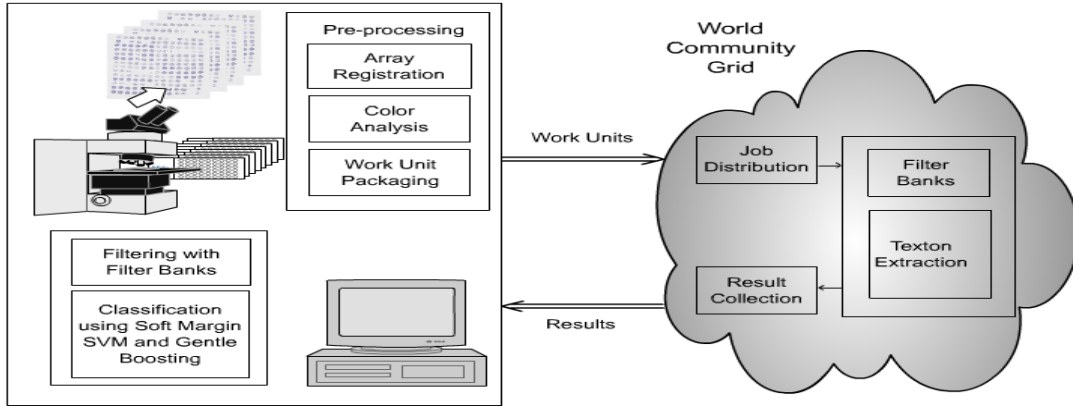


Figure 5.2: The work-flow and logical units of the Grid-enabled tissue microarray.

## 5.2 Data Generation and System Framework

The Tissue microarrays (TMA) used in the HDC project was collected from The Cancer Institute of New Jersey, Yale University, University of Pennsylvania and Imagenex Corporation (San Diego, CA). To date over 300 slides containing cohorts of hundreds of tissue discs each and originating from 45 TMAs were digitized at  $40\times$  resolution using a Trestle MedMicro virtual microscopy system. The output images typically contain 1-3 billions of pixels and were stored as a compressed tiled TIFF file sized at 0.5 to 2 Gigabytes. Our registration protocols [16] were applied to the scanned images to identify rows and columns of the tissue arrays. Any tissue cores that suffered from exceedingly pronounced artifacts were excluded from the study. Images of each tissue core were systematically extracted from the archive and packaged as workunits for the HDC project. The dimension of each image was  $1200 \times 1200$ . The specimens under study had previously been stained with hematoxylin and hematoxylin & eosin. A texton extraction algorithm was applied on the staining maps of the two dyes which were generated using color decomposition [16]. Each of the resulting staining maps as well as the luminance measure generated from the original color image were uploaded as separate workunits to the WCG. The work-flow and logical units are shown in Figure 5.2.

### 5.3 Data Analysis

In this section, we explain the methods used to generate and analyze the image features for automatic classification of breast tissue specimens. Textures and intensities were used as feature measures to classify the staining profiles of the imaged tissues. Because the feature vectors lie in a high dimensional space, we applied a nonlinear dimension reduction method to decrease the dimensionality. Through iterative experiments we determined that among several different classification algorithms, the gentle AdaBoost classifier provided the best overall performance in the reduced subspace.

#### 5.3.1 Features

It can be found that the difference in texture can be used as the discriminative features to separate different types of breast tissues. Traditional texture analysis includes Law’s moment [76], cooccurrence matrices [56], run length matrices [46] and autoregressive models [92] et. al.

In recent studies, texture has been represented using texton. Textons are defined as conspicuous repetitive local features that humans perceive as being discriminative between textures. Unlike many other texture features that describe each texture as a constant relationship – a number, a data vector or a set of model parameters – between each pixel and its surroundings, the concept of a texton supports the existence of numerous distinct textual components in each texture. Therefore, it has advantages in describing textures that have high-level components. Texton based texture analysis has been widely used in many fields of texture related research, including classification [81], [140], [28], [136], segmentation [159] and synthesis [58].

The feature vector is composed of eight *LoG* filter responses, four Gaussian filtering responses and the bar and edge filtering response within six different directions. In total, each image pixel was represented by a 48 dimensional feature vector. The image filtering response generated using the collective computation power of the World Community Grid were gathered together and clustered using *K*-means, where *K* was set to 4000 in our experiments. The cluster centers, called textons, were used to generate

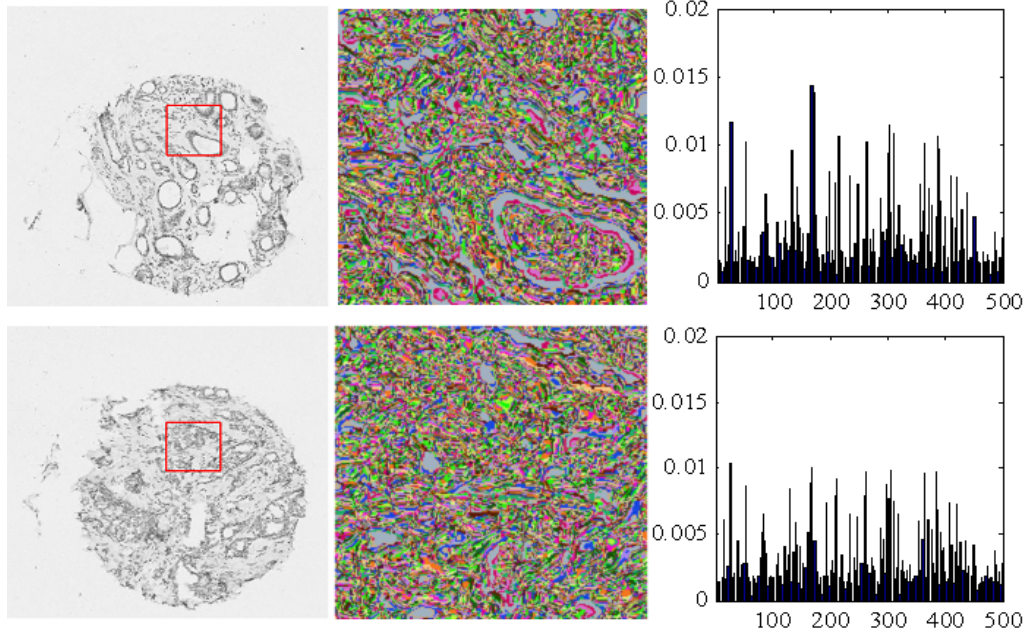


Figure 5.3: Two breast tissue images on the left, benign (on the top) cancer (on the bottom). In order to keep the figure readable, only the texton maps of red rectangle regions were shown in the middle with different colors representing different textons. In practice the texton histograms of full images, shown on the right, were used for classification.

the texton library. The appearance of each breast tissue image was modeled by a compact quantized description called texton histograms. Texton histograms are created by assigning each pixel filter response in the image to its closest texton in the generated texton library, which was calculated using

$$h(i) = \sum_{j \in I} \text{count}(T(j) = i) \quad (5.1)$$

where  $I$  denotes breast tissue image,  $i$  is the  $i$ th element of the texton dictionary,  $T(j)$  returns the texton assigned to pixel  $j$ . In this way, each breast tissue image was modeled as a texture modes distribution, the texton histogram. Each image was mapped to one point in the high dimension space  $R^d$ , where  $d = K = 4000$  is the number of textons. Figure 5.3 shows the texton histograms of two breast tissue images.

### 5.3.2 Dimension Reduction

Each breast cancer image is represented by a vector in the  $d = 4000$  dimension space in which we have to consider the “curse of dimensionality”. Traditionally, linear dimension reduction methods like multidimensional scaling (MDS) and principle component analysis (PCA) [35] have been used, which assume that the data can be represented by a lower dimensional linear subspace. In other cases, the data can be modeled by a low-dimensional nonlinear manifold, and nonlinear methods such as locally linear embedding (LLE) [117], isometric feature mapping (ISOMAP) [130] and local tangent space alignment (LTSA) [164] are more appropriate.

Given a set of feature vector  $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_i, \dots, \mathbf{z}_n\}$  where  $\mathbf{z}_i \in R^d$ . There exists a mapping  $T$  which can represent  $\mathbf{z}_i$  in the low dimension as

$$\mathbf{z}_i = T(\mathbf{x}_i) + \mathbf{u}_i \quad i = 1, 2, \dots, n \quad (5.2)$$

where  $\mathbf{u}_i \in R^d$  is the sampling noise and  $\mathbf{x}_i \in R^{d'}$  denotes the representation of  $\mathbf{z}_i$  in low-dimensional space.

PCA finds the mapping  $T$  which best represents the variance of data  $\mathbf{z}_i$  in the original high-dimensional space. The low-dimensional space  $R^{d'}$  is spanned by the  $d'$  largest eigenvectors of the covariance matrix of  $\mathbf{z}_i$ . MDS finds the mapping  $T$  which preserves the pairwise distances between  $\mathbf{z}_i$ -s in  $R^{d'}$ . If the data have certain geometric structure which can be modeled as a low-dimensional linear manifold, they are not expected to perform well.

Although most of the geometric distances can not be used directly on the manifold, linear dimension reduction methods can be applied locally and Euclidean distances are valid on the local tangent space. The LLE preserves the geometry of the data points by representing points using their local neighbors. The ISOMAP approach applies a linear MDS on the local patch, but aim to preserve the geometric distance globally using the shortest path in the graph. The LTSA method maps each data point from the original space into the tangent space and align it to give a global coordinate.

If the data are approximately sampled from a low-dimensional manifold, the non-linear methods generally perform better [117]. Otherwise the linear methods may be

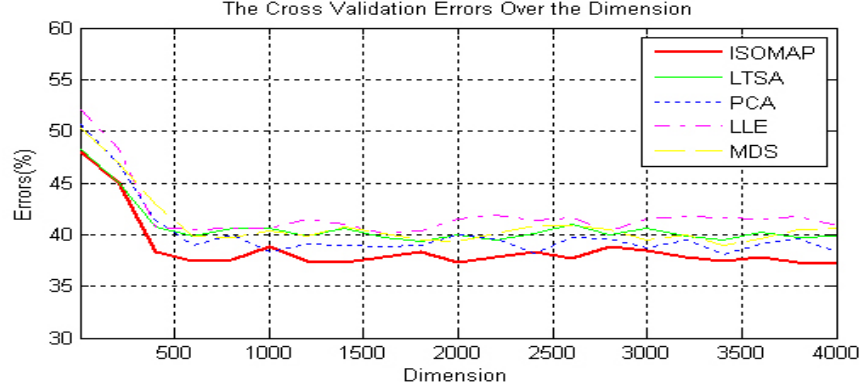


Figure 5.4: The five different dimension reduction algorithms shown as five-fold cross-validation errors for different reduced dimensions.

preferred because of their simplicity. In order to compare the performance of these methods, we used cross-validation (CV)

$$CV(\gamma) = \frac{1}{N} \sum_{i=1}^N |y_i - f^{-k(i)}(\mathbf{x}_i, \gamma)| \quad (5.3)$$

where  $\mathbf{x}_i$  is the feature vector in  $R^{d'}$ ,  $y_i = \{+1, -1\}$  represents the cancer and benign breast tissue labels. The  $f^{-k}(\mathbf{x}_i, \gamma)$  denotes the classification results using the  $\gamma$ -th dimension reduction method with the  $k$ -th part removed from the training data, and the number of partitioning  $k = 5$  in our experiments. We tested the five different algorithms from 3 to 4000 dimensions and determined that ISOMAP is better (Figure 5.4). Good performance using ISOMAP was also reported in [79] for tissue characterization. Based upon these comparative results, ISOMAP with 500 dimensions are used for all subsequent operations.

### 5.3.3 Classification

In [103], the  $k$ -nearest neighbor ( $kNN$ ) and classification tree ( $C4.5$ ) were integrated into a Bayesian framework for characterizing breast tissues. However, in our case, each training sample was represented by a feature vector  $\mathbf{x}_i$  in the reduced subspace  $R^q$  where  $q = 500$ . This is still a relatively high dimension where the maximal margin classifiers such as support vector machine (SVM) [26] and boosting [44] are better suited. We conducted experiments to compare the performance of four boosting algorithms, the standard AdaBoost, the gentle AdaBoost, the real AdaBoost and LogitBoost with

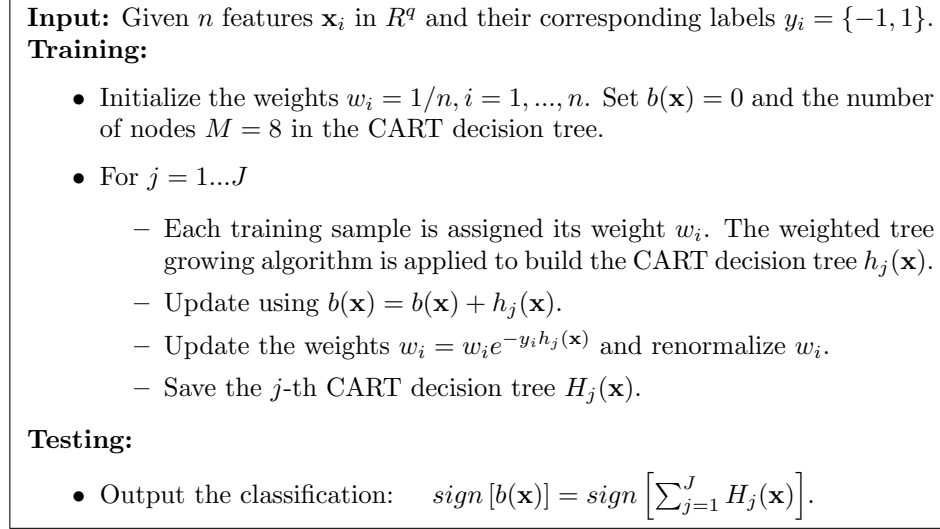


Figure 5.5: The binary gentle AdaBoost using an eight nodes classification and regression tree (CART) as the weak learner.

$kNN$ , Bayesian classifier and SVM. The results showed that the maximal margin classifiers [26], [44], such as SVM and boosting, which simultaneously minimize the empirical classification error and maximize the geometric margin, outperformed all the other algorithms. In order to separate two subtypes of breast cancers from the benign, the best binary classifier in our experiments (the gentle AdaBoost) was extended to a multi-class algorithm

The  $kNN$  consists of assigning all the features into  $k$  most similar cluster centers based on certain similarity measurements. The final label was determined by majority voting from  $k$  candidates.

Let  $\mathbf{x} \in X$  represent the low level feature in the reduced subspace  $R^q$ , the *Bayesian classifier* is designed to maximize a-posterior (*MAP*) probability

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)p(C_i)}{\sum_{k=1}^K p(\mathbf{x}|C_k)p(C_k)} \quad (5.4)$$

and the Bayesian classifier determines the class  $C_i$  by maximizing the posterior probability  $p(C_i|\mathbf{x})$ .

The *support vector machine* (SVM) was first introduced in [26] for binary classification problem. The strategy is to construct the linear decision boundaries in a large

transformed version of the original feature space. The SVM simultaneously minimizes the empirical classification error and maximizes the geometric margins by minimizing the regularization penalty

$$\frac{1}{2} \|\mathbf{w}\|^2, \text{ subject to } y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) - 1 \geq 0 \quad (5.5)$$

When the examples are not linearly separable, the optimization can be modified by adding a penalty for violating the classification constraints. This is called *soft margin SVM* which minimizes

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i, \text{ subject to } y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) - 1 + \xi_i \geq 0 \quad (5.6)$$

where  $\xi_i$  are called slack variables which store the deviation from the margin and  $C$  is the soft penalty to balance the training errors and margins. In (5.5) and (5.6),  $\mathbf{w}$  is the slope of the decision hyperplane and  $w_0$  is the offset. The  $\mathbf{x}_i$  denotes the feature vector, and  $y_i$  is the ground true labels. We minimize (5.6) by maximizing the dual problem of (5.6) which involve a feature mapping  $\phi(\mathbf{x})$  through an inner product. The inner product can be evaluated without ever explicitly constructing the feature vectors  $\phi(\mathbf{x})$  but through a kernel function  $\kappa(\mathbf{x}, \mathbf{x}')$ . In our algorithm, we proposed to use a nonlinear Mercer kernel [122] based on  $\chi^2$  distance. It was shown that among other choices of distance functions between histograms,  $\chi^2$  distance performed the best for the texture similarity measure. The kernel function is defined as

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{\tau} \chi^2(\mathbf{x}, \mathbf{x}')\right) \quad (5.7)$$

where

$$\chi^2(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \sum_{l=1}^{500} \frac{(\mathbf{x}(l) - \mathbf{x}'(l))^2}{\mathbf{x}(l) + \mathbf{x}'(l)}. \quad (5.8)$$

The detailed description is given in Figure 5.6.

*Boosting* works by sequentially applying a classification algorithm on a reweighted version of the training data and producing a sequence of weak classifiers  $h_j(\mathbf{x})$ ,  $j = 1, 2, \dots, W$  where  $W = 40$  in our case represents the number of iteration rounds of each boosting algorithm. The strong classifier is assembled from all the weak classifiers  $h_j(\mathbf{x})$



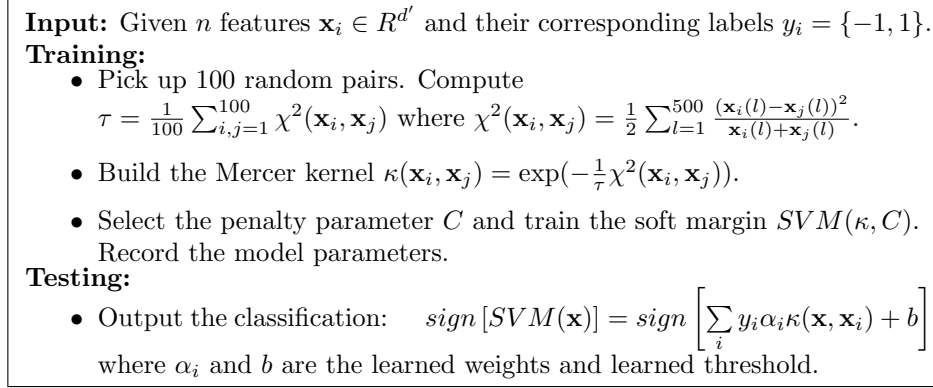


Figure 5.6: Soft Margin SVM using the Mercer kernel based on  $\chi^2$  distance.

to minimize the exponential cost function  $\exp(-yh_j(\mathbf{x}))$ , where  $y$  represents the label of the training sample  $\mathbf{x}$ . In the standard binary AdaBoost classification, the labels were decided by weighted voting to produce the final prediction

$$\hat{y} = sign \left( \sum_{j=1}^W \alpha_j h_j(\mathbf{x}) \right) = sign(H(\mathbf{x})) \quad (5.9)$$

where  $H(\mathbf{x})$  is the learned strong classifier. The  $\alpha_j$  is the weight of the  $j$ -th weak classifier  $h_j(\mathbf{x})$  and is computed during training. All the boosting algorithms are designed to minimize an exponential cost function  $\exp \left( -y \sum_{j=1}^W \alpha_j h_j(\mathbf{x}) \right)$ . If the weak classifier  $h_j(\mathbf{x})$  returns a discrete class label  $\{-1, +1\}$ , the boosting algorithm is called *AdaBoost*. Instead of making a hard decision, if the weak classifier  $h_j(\mathbf{x})$  returns a real value prediction like a probability mapped to the interval  $[-1, +1]$ , it is called *real AdaBoost*. The *gentle AdaBoost* is a modified version of the real AdaBoost algorithm, which applies Newton step rather than exact optimization at each step of minimizing the loss function. The *LogitBoost* is another boosting algorithm which uses Newton steps to fit an additive logistic regress model based on maximum likelihood. The weak classifier we used was an eight node classification and regression tree (*CART*).

We experimentally tested each of the classification algorithm. The gentle AdaBoost using an eight node *CART* decision tree provided the best results for binary classification problem. Figure 5.5 shows the details of the gentle AdaBoost algorithm.

Multi-class experiments were designed to determine the capacity of the system to

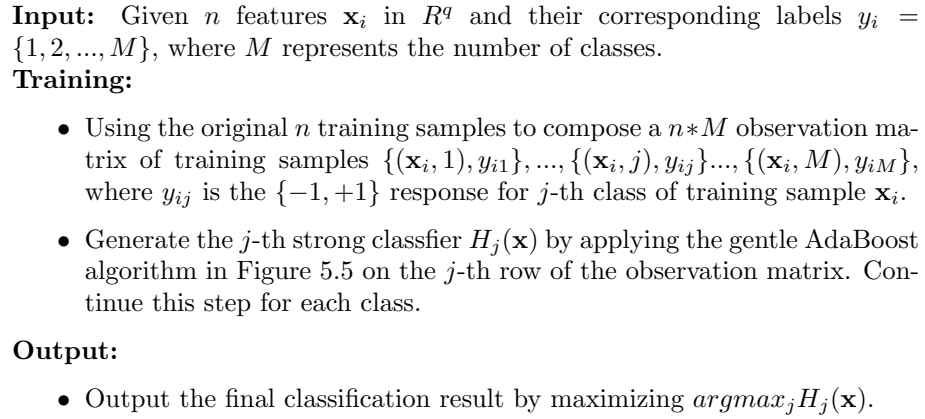


Figure 5.7: The multi-class gentle AdaBoost using an eight nodes classification and regression tree (CART) as the weak learner.

subclassify different types of cancer. Given a  $M$ -class classification problem and  $N$  training samples  $\{\mathbf{x}_1, y_1\}, \dots, \{\mathbf{x}_N, y_N\}$ . The  $\mathbf{x}_i \in R^q$  denotes the  $i$ -th feature vector in the reduced subspace and  $y_i \in \{1, 2, \dots, M\}$  represents the corresponding ground-truth class labels. The target is to find a strong classifier which minimizes a multi-class exponential loss function  $\sum_{j=1}^M \exp(-y_i H_j(\mathbf{x}))$  where  $H_j(\mathbf{x})$  is the  $j$ -th strong classifier. This is equivalent to run separate boosting algorithms in an one-against-all manner. One-against-all boosting constructs  $M$  binary classifier, each of which is used to separate one class from all the others. The  $j$ -th strong classifier was trained using boosting with all the training samples satisfying  $y_i = j$ ,  $i = 1, 2, \dots, N$  as positive and all the others as negative. As the gentle AdaBoost outperformed the other methods in the previous binary classification, we extended it to classify two different subtypes of cancers from benign tissue images in the multi-class experiment. The multi-class gentle AdaBoost algorithm is shown in Figure 5.7.

## 5.4 Experiments

The tissue microarrays used in our experiments were prepared by different institutes: the Cancer Institute of New Jersey, Yale University, University of Pennsylvania and Imgenex Corporation, San Diego, CA. To date over 300 immunostained microscopic specimens, each containing hundreds of tissue image, were digitized at  $40\times$  volume

Table 5.1: The confusion matrix of three classes with column labels as inferred class and row labels as ground-truth classes

(%)	Benign	Cancer I	Cancer II
Benign	<b>84.5</b>	6.4	7.1
Cancer I	6.8	<b>81.2</b>	13.8
Cancer II	8.7	12.4	<b>79.1</b>

scan using the Trestle MedMicro, a whole slide scanner system. The output images typically contain a few billions of pixels and are stored as a compressed tiled TIFF file sized at about two gigabytes. The registration protocol proposed by [16] was applied to automatically identify the rows and columns of the tissue arrays. Staining maps of the two dyes, diaminobenzidine (DAB) and hematoxylin, were generated from specimens and each of the two staining maps as well as the luminance of the original color image were submitted to the IBM World Community Grid for batch processing.

We have analyzed 3744 breast cancer tissues (674 hematoxylin and 3070 DAB staining) from the 100,000 images processed on the Grid. Without the Grid, it would require about 210 days of computation to generate the texton library even with an efficient C++ implementation on a PC with P3 1.5GHz processor and 1G RAM. However, we can build this universal texton library in less than 40 minutes in the largely distributed computing system [7].

The labels of all breast tissues from the hospitals and institutions are independently confirmed by certificated surgical pathologists. The dataset used in these experiments consisted of 611 benign and 3133 cancer samples. Each of the five algorithms was applied 10 times, using different parts of the training images drawn by random sampling. Figure 5.8 shows the average classification results. Because there were more positive samples than the negative samples, we obtained higher false positive errors but lower false negative errors (Figure 5.8b) than the average error (Figure 5.8a).

The experimental results are presented for studies in which the original gentle AdaBoost algorithm was modified to accommodate multi-class classification. Based on the direction of the clinical pathologist, we separated six subtypes of cancer tissues into two sub-groups: cancer class I which contains DCIS and LCIS and cancer class

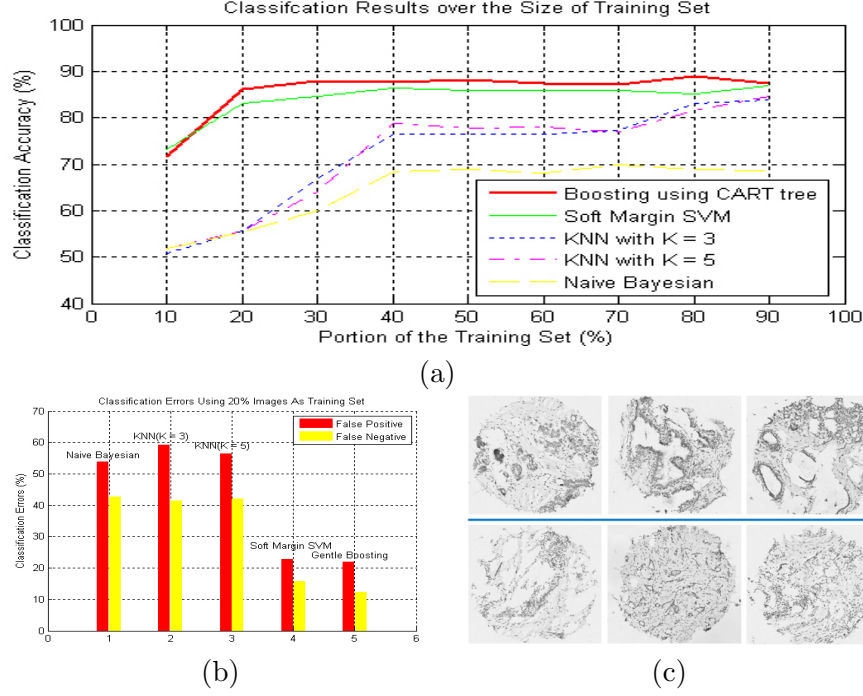


Figure 5.8: The classification results. (a) The accuracy as the function of the size of training set using Gentle Boosting, Soft Margin SVM,  $KNN$  with  $K = 3$  or  $5$  and naive Bayesian. (b) The false positive and false negative errors using 20% images as training set. (c) Some misclassified samples. The upper row is false positive and lower row is false negative.

II containing IDC, ILC, LNN and STM. The dataset is consisted of 611 benign, 1103 cancer class I and 2030 cancer class II. 30% of the images in each class were randomly selected for training and the remaining 70% was used for testing. The classification accuracy is reported on the left of Figure 5.1. The confusion matrix is presented on the right of Figure 5.1. Figure 5.9 shows some correct classification samples and failed cases. The left most three columns are correctly classified samples, and the right most fourth column shows the failed cases. The first row is the benign tissue where the last one is misclassified as cancer class II. The second row represents cancer I while the last tissue image is misclassified as benign. The last row is the cancer II, and the last image is misclassified as cancer I. In Figure 5.9 we show the large intra-class variances and inter-class similarities which produced the classification errors.

From all these experiments, it was shown that the gentle AdaBoost provided satisfactory results on both binary and multi-class classification of breast tissue images.

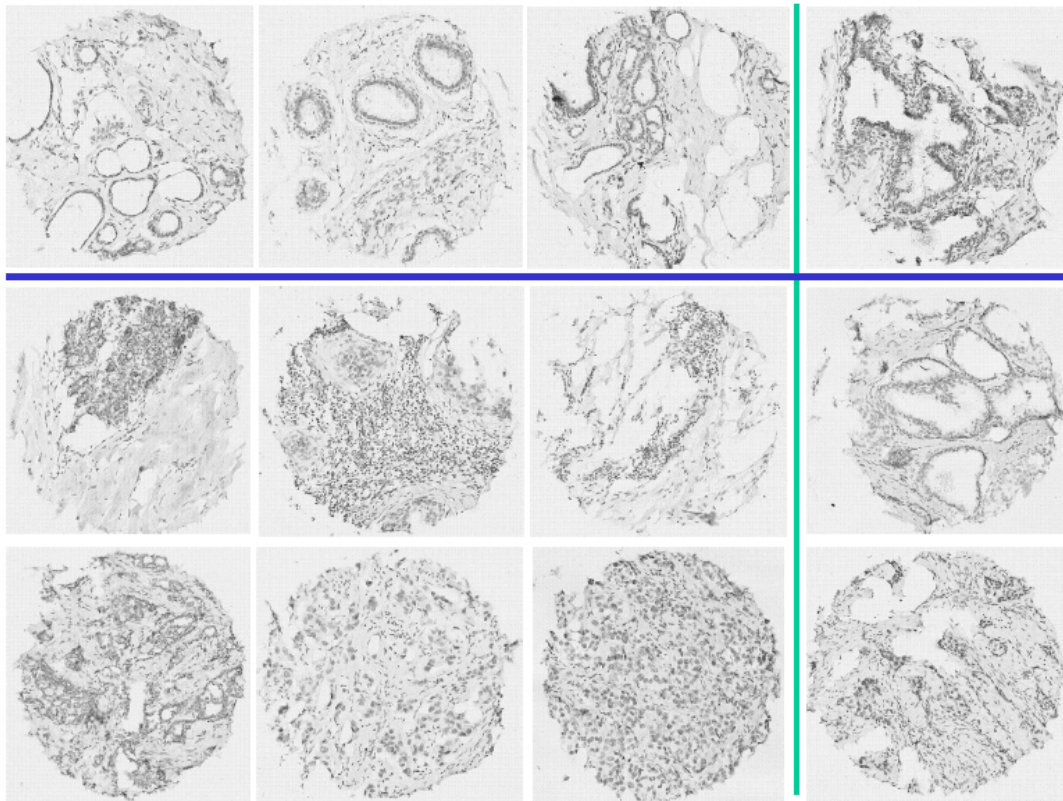


Figure 5.9: The multi-class classification results using the gentle AdaBoost. The left three columns are correct classified samples and the right fourth column shows the failed cases. The first row is the benign samples. The second and third rows are the cancer samples.

We obtained an average 89% accuracy in separating benign from cancer tissue and an average accuracy of 80% in classifying two types of breast cancers from benign. In both cases only 30% of the images were used in the training.

## Chapter 6

### A Fast Nonlinear Image Registration Using Landmarks and Robust Estimation

#### 6.1 Introduction

A set of image acquired at different time, or from different perspectives, will be in different coordinate systems. It is therefore critical to align those images into the same coordinate system before applying any image analysis. Image registration is the process to determine the linear/nonlinear mapping  $T$  between two images of the same object or similar objects. Throughout the whole chapter, we will define one of the image as the fixed image, the other image which will be registered to the fixed image is defined as moving image. Image registration is widely used in remote sensing, image fusion, image mosaicing and especially medical image analysis. Different imaging modalities, such as CT, MRI, ultrasound and PET, etc., have their own advantages and disadvantages. It would be useful for the doctors to make the decision based on different imaging modalities. However, it is very common that a patient's image, either same modality or multiple modalities, was acquired under different positions or deformations when it was taken. In Figure 6.1, we show two sets of examples which require the image registration. Figure 6.1a shows three images of the same person which have different viewpoints. In Figure 6.1b, although all the images are roughly aligned, they were taken from three different persons. For either situation, the image registration is one of the key challenges.

Image registration can be separated as rigid registration [90, 59] and non-rigid registration [80, 168]. Given the fixed and moving images, both of the problems can be described as finding a linear/nonlinear transformation which maps each point in the fixed image to a point in the moving image. There are usually two ways to describe

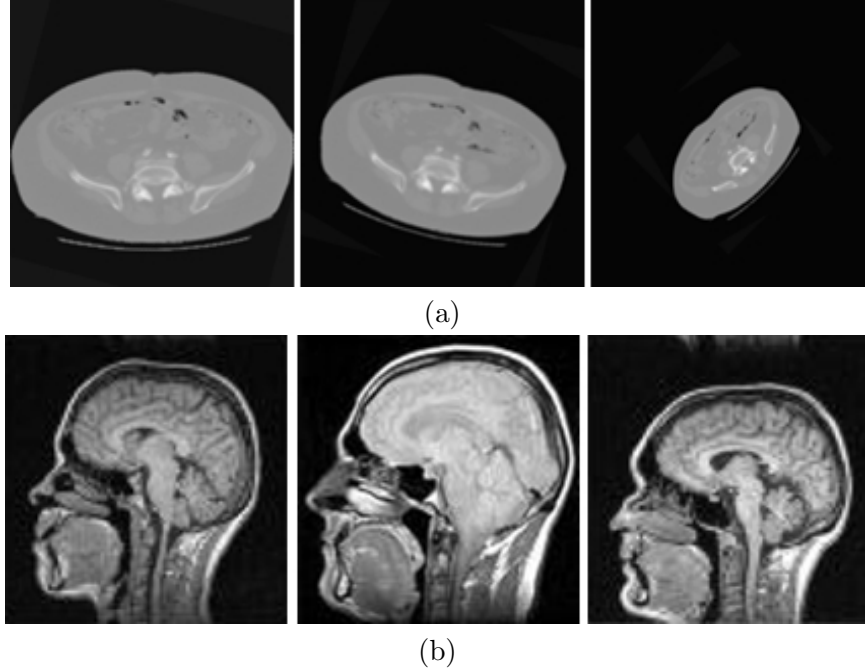


Figure 6.1: The image registration examples. (a) Three images which have different view points. (b) Three side view images which were taken for three different persons.

the transformations. The first method is to define a dense nonparametric model and estimate the position of each point after the registration. Many nonrigid registration algorithm based on elastic deformations, such as fluid deformation based algorithm [13, 2] and Demon's algorithm [133, 141], fall into this category. The second approach is to model the transformation by a function with some parameters. By estimating the parameters, the algorithm can register each point in the fixed image with the moving image. B-spline based image registration algorithm belongs to this group. The modeling of the nonlinear deformation using B-spline was introduced in [129]. Mutual information was used in [115] as the cost function where the transformation is modeled as B-spline. A block nonlinear Gauss-Seidel algorithm is used in [98] to minimize the cost function in 2D image registration, where the transformation is modeled as B-spline. B-spline transformation is extended to model transformation for 3D image registration in [101].

In this chapter, we will introduce a fast and robust image registration algorithm for both 2D and 3D images which belongs to the second category. The algorithm starts from an automatically detection of the landmarks followed by a coarse to fine estimation of

the nonlinear mapping. For 2D images, multiple resolution oriental histograms is used to obtain fast affine invariant local descriptor of the detected landmarks. For 3D images, considering both the speed and accuracy, the global nonrigid linear registration is first applied to pre-align two 3D images. Simple template matching is further used to obtain the point correspondence between landmarks in the fixed and moving images. Because there is a large portion of outliers in the initial landmark correspondence, a robust estimator, RANSAC [42], is applied to reject outliers. The final refined inliers are used to robustly estimate a thin spline transform (TPS) [20] to complete the final nonlinear registration. The algorithm is completely unsupervised and computationally efficient. Due to the small number of landmarks, the method runs significantly faster than many widely used nonlinear registration algorithms, such as Demon's and B-spline. Compared with the state-of-the-art, the proposed algorithm can also handle large transformation and deformation while still provide good registration results. The parallelization of the proposed algorithm on an IBM Cell/B.E. processor was also explained in details.

In Section 6.2 we introduce the image registration algorithm. In Section 6.3 we will review the current parallelization algorithm for image registration. In Section 6.4, we describe the parallelization implementation using a IBM Cell/B.E. The experimental results were shown in Section 6.5 and Section 6.6 concludes the paper.

## 6.2 Image Registration

The image registration algorithm starts from automatic detection of a set of landmarks in both fixed and moving images, followed by a coarse to fine estimation of the nonlinear mapping using the landmarks. Robust estimation is used to find the robust correspondence between the landmarks in the fixed and moving image. The refined inliers are used to estimate a nonlinear transformation  $T$  and also wrap the moving image to the fixed image.



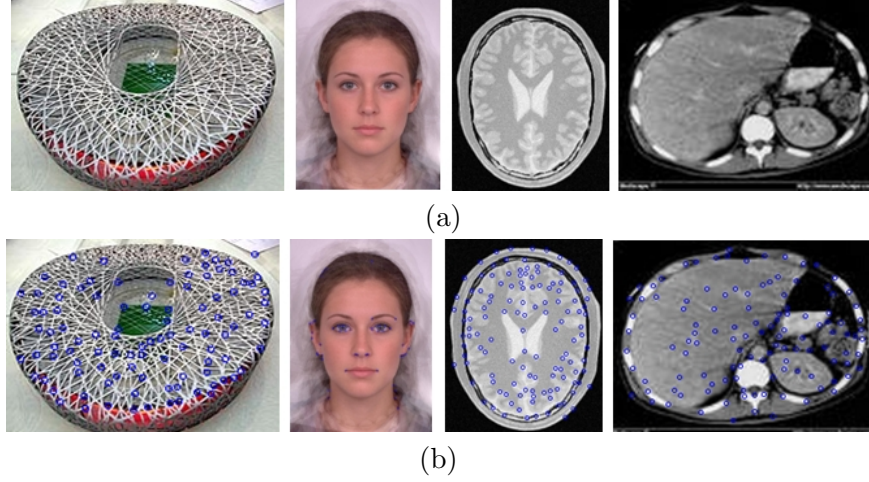


Figure 6.2: The landmark detection using Harris corner detector. (a) The original images. (b) The images with the landmarks overlayed.

### 6.2.1 Landmark Detection and Matching

The automatic landmark detection is the procedure of accurate detection of the prominent or salient points in the image. Harris corner detector is applied to find the point with the large gradient on all the directions (x and y direction for 2D image and x, y and z directions for 3D image). The original computation in the Harris corner detector involves the exact computation of eigenvalues. Instead, we are using determinant and trace to find the corners where  $\alpha$  is chosen as 0.1.

$$F = \det(A) - \alpha \text{trace}(A) \quad (6.1)$$

Some landmark detection examples are shown in Figure 6.2.

After we detect the landmarks, we can extract features from the neighborhood of each landmark. For 2D images, the local orientation histograms are used as the features for landmark matching. The image is first convolved with the orientation filters. The filtering response in the neighborhood around the landmarks is computed to form the orientation histogram. Let  $G_x(i, j)$  and  $G_y(i, j)$  represent the gradients on pixel  $p(i, j)$  along  $x$  and  $y$  direction, respectively. The orientation histogram  $h_k$  is defined as

$$h_k(i, j) = \begin{cases} C(i, j), & \theta(i, j) \in \text{bin}_k \\ 0, & \text{otherwise} \end{cases} \quad (6.2)$$

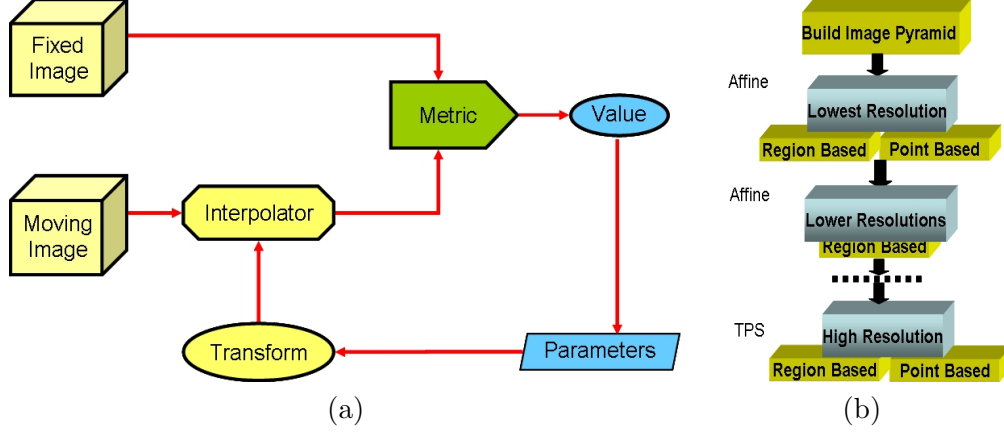


Figure 6.3: The procedure for 3D image registration. (a) The procedure for the region based image registration. (b) The flowchart for the whole coarse to fine image registration algorithm.

where

$$C(i, j) = \sqrt{G_x(i, j) + G_y(i, j)} \quad (6.3)$$

and

$$\theta(i, j) = \arctan \left( \frac{G_y(i, j)}{G_x(i, j)} \right). \quad (6.4)$$

The orientation histogram encodes the directions of the edges at each landmark point and is proven to be an effective when the training samples are small [83].

For 3D image, considering speed, image intensity template in the neighborhood of pixel  $p(i, j)$  is used for 3D landmark matching. Region based coarse to fine image registration was also used in 3D image to increase the accuracy for landmark matching considering both the speed and robustness. In Figure 6.3a we show the procedure for the region based 3D image registration. The cost function used in our algorithm is the mean square distance metric for single modality image registration and the mutual information [109] for multiple modalities. The optimization is the simple gradient decent method. However, for 3D similarity transformation, because rotation is not on the vector space, quaternion [55] based gradient decent is used instead for optimization. The region based and landmark based image registration can benefit each other because:

- the coarse to fine region registration increase the robustness of landmark based registration;

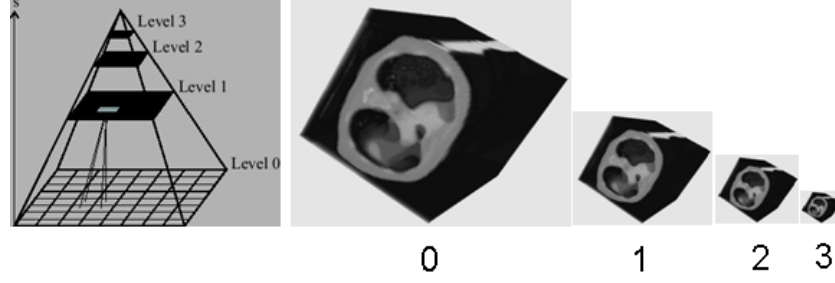


Figure 6.4: An example of a three layer Gaussian image pyramid and their corresponding 3D volumetric data.

- the landmark based registration increase the speed and help the region based optimization to recover from local minimal.

The whole coarse to fine image registration procedure using both region and landmarks is shown in Figure 6.3b. We should emphasize that the Gaussian image pyramid was applied to reduce the noise and increase the robustness. The levels of the pyramid is automatically calculated by setting a fixed size ( $64 \times 64 \times 64$ ) in the lowest resolution. The number of pyramid levels is then calculated given the actual size of the input image. The Gaussian kernel is 0 mean and  $\sigma = 5$ . This adaptive selection of the image pyramid levels can improve the scalability of the algorithm, because large image will be assigned more levels while smaller image will has less levels. One example of a three layers Gaussian pyramid built on the whole body MRI image is shown in Figure 6.4.

### 6.2.2 Robust Estimation and Nonlinear Image Registration

Because the original matching landmark sets contain missing landmark matching or error landmark matching. RANdom SAmple Consensus (RANSAC) [42] is used to reject outliers and robustly estimate the transformation. The algorithm is listed in Figure 6.5. The RANSAC robust estimator randomly choose the minimal subset of the landmarks to fit the model. Measured by a cost function, the points within a small distance will be considered as a consensus set. The size of the consensus set is called the model support  $M$ . The algorithm is repeated multiple times and the model with largest support is called the robust fit. In Figure 6.6 we show the results of applying robust estimation to reject the outliers in the original matching landmarks. The Harris

**Input:** Given  $N$   $m$ -tuples of data, threshold  $\tau$ , and probability  $p$

**Algorithm:**

- $K = \infty, k = 0, p^* = [], M^* = 0$
- while  $K > k$  do
  - Choose a sample set  $S$  from  $N$  with size  $m$
  - Compute the model parameter  $p$  from the sample set  $S$
  - Count the number  $M$  of inliers using  $p$  and  $\tau$
  - if  $M > M^*$ 
    - \*  $M^* = M$
    - \*  $p^* = p$
    - \*  $w = M/N$
    - \*  $K = \text{ceil} \left( \frac{\log(1-p)}{\log(1-w^m)} \right)$
    - \*  $k = k + 1$
  - end if
- end while

**Output:** parameter  $p^*$

Figure 6.5: The RANSAC algorithm

corner detector detected 32 landmark pairs in Figure 6.6(b). Based on the assumption of an affine transformation, the RANSAC found 8 inliers (shown in Figure 6.6(c)) and the rest 24 matching landmarks are rejected as outliers under the assumption for Affine transformation.

The thin plate spline transform (TPS) is used to estimate the nonlinear transformation between the fixed and moving image based on the robust landmark correspondence. By minimizing the bending energy of the mapping transformation  $T$  defined in equation (2.33), TPS can provide a smooth matching function for each point in both images. The resulting nonlinear transformation is applied to map the moving image to the fixed image.

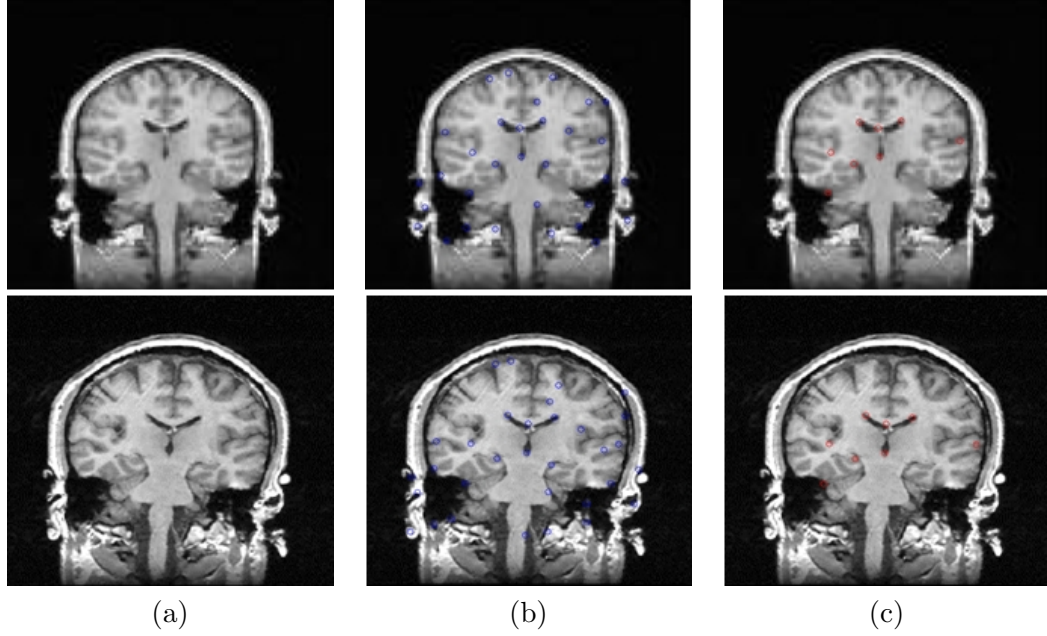


Figure 6.6: Apply the robust estimation to find the robust landmark correspondence. (a) The original fixed and moving image. (b) The original pairs of matching points. (c) The robust matching points after rejecting the outliers.

### 6.3 Current Parallelization Method For Image Registration

While research effort continues to explore methods and strategies for more efficient and rapidly converging computational methods, increasing attention has been given to hardware architecture-based parallelization and optimization algorithms for the emerging high performance architectures, such as cluster and grid computing, advanced graphical processing units (GPU) and multicore computers. In [19] a parallel implementation of multimodal registration is reported for image guided neurosurgery using distributed and grid computing. The registration time was improved dramatically to near real-time. In [144] a distributed computation framework is developed, which allowed the parallelization of registration metrics and image segmentation/visualization algorithms while maximizing the performance and portability. One key deterring factor in adopting supercomputer-based, cluster-based or grid computing architectures is availability and cost. Even for some large clinical institutions, financial limitations can be a major hurdle. The recent emergence of low cost, high computing power, multi-core processor

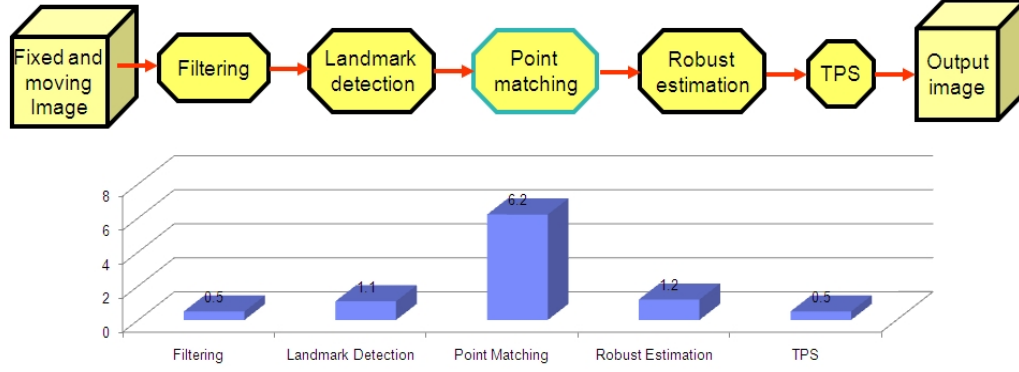


Figure 6.7: The time profile for the entire nonlinear registration algorithm

systems have opened up an alternative venue for developing cost-effective high performance medical image registration techniques. In [102], a close to real time implementation of a mutual information based linear registration is reported. The algorithm is designed based on the Cell Broadband Engine (Cell/B.E.) multicore processor architecture. General parallel data mining algorithms on the Cell/B.E. are reported in [10, 34]. In [47], a high performance distributed sort algorithm is proposed for the Cell processor. As an application, the implementation of the Radioastronomy image synthesis on the Cell/B.E. is discussed in [139]. According to our knowledge, this is the first study reporting fast and robust *landmark based nonlinear* image registration algorithm on a multicore platform.

#### 6.4 Parallelization on the IBM Cell/B.E.

The adaptive multi-resolution region and landmark based image registration algorithm can provide good registration results, but requires relatively time consuming point matching procedure. In Figure 6.7 we show the executive time profile for each step in our algorithm for 2D image registration. It is quite obvious that the bottleneck is the landmark matching part. Therefore we implemented the whole landmark matching routine on the IBM Cell/B.E. processor.

The IBM Cell/B.E. [69, 15] is a multi-core chip with a relatively high number of cores. It contains a Power Processing Element (PPE) which has the similar function and configuration as the regular CPU. It also has multiple cores which are optimized for



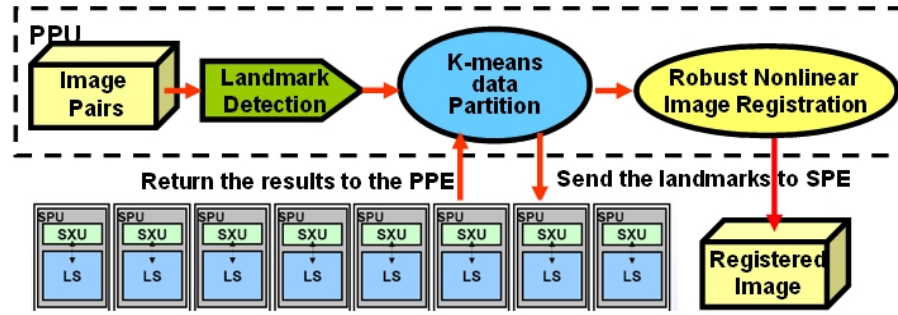
Figure 6.8: The multi-core architecture of the IBM Cell/B.E.

single precision float point algorithm, the Synergistic Processing Element (SPE). The PPE contains 32K  $L_1$  cache, 512K  $L_2$  cache and a large amount of physical memory (2G in our case). Unlike the PPE, the SPE has a quite different architecture compared with the standard CPU. The SPE operates on a 256KB local store to hold both code and data. The SPE also support 128 bit Single Instruction, Multiple Data (SIMD) instruction set for effective vector operations. The data transfer between SPE and PPE is through the direct memory access (DMA). DMA is quite time consuming so that a good parallel implementation should minimize the number of DMA operations. The heterogeneous multi-core architecture of the Cell/B.E is shown in Figure 6.8. The parallel implementation of a linear registration algorithm was reported in [102]. To our knowledge, our work is the first parallel implementation of a hybrid landmark and region based nonlinear image registration on the cell/B.E..

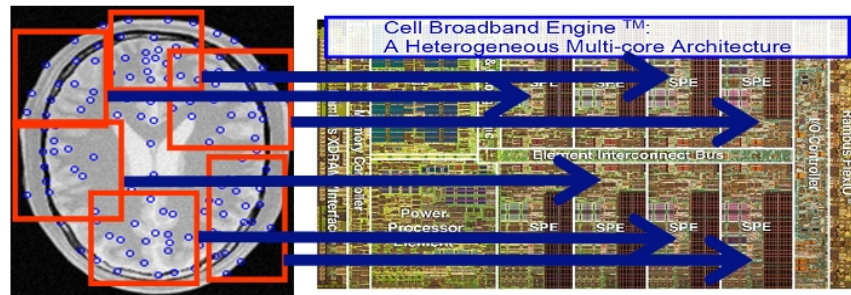
#### 6.4.1 Data Partitioning and Parallelization

Given all the detected landmarks in the fixed image, we first apply the  $K$ -means algorithm to cluster them based on their Euclidean distance in the image, where  $K = 16$  is set to be the number of the computing units in the IBM Cell Blade machine. Based on the boundary landmarks in each cluster center, we can calculate the largest and smallest coordinates to crop the sub-image accordingly. Because we know the size of the code running on the SPE unit in advance, we can compute the maximal size of the sub-image that can be stored on a single core (e.g. single SPE). If the image patch can





(a)



(b)

Figure 6.9: The procedure of the parallel image registration algorithm. (a) The flow chart of the parallelization. (b) The data partitioning using K-means clustering.

fit into the local storage, the whole cluster of landmarks and their corresponding image patch are sent to the SPE for parallel processing using just one direct memory access (DMA). Because the number of DMA is critical for the performance of the parallel algorithm, the advantage of applying  $K$ -means to group the landmarks into clusters is to minimize the number of direct memory access operations. Landmarks which are spatially close to each other are grouped together and transferred into one computing core (e.g. one SPE in a Cell Processor) using one DMA call. In Figure 6.9a we show the procedure of parallelizing the registration algorithm. The  $K$ -means clustering and job scheduling to 16 PPEs are illustrated in Figure 6.9b.

Thread for each core or Synergistic Processing Element (SPE) is created only once in order to reduce the overhead related to thread creation. The point matching for our registration implementation is executed twice, one during the stage of linear registration and on during the stage of nonrigid registration. If we create an SPE thread each time when a point matching process starts, then all the SPE threads have to be created



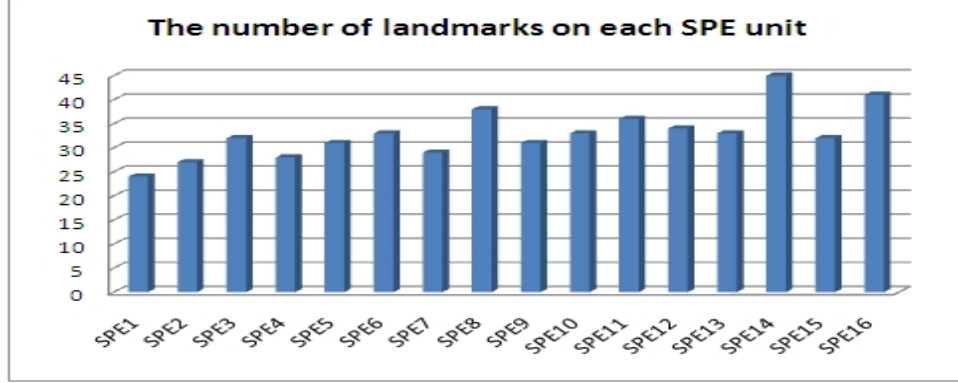


Figure 6.10: The landmark distribution on each SPE unit of the IBM Cell/B.E.

and destroyed repeatedly resulting additional execution time. In our implementation, all SPE threads are created before the first execution of the point matching process. These threads will be kept alive until the second point matching process is completed. It is true that the created SPE threads will be idle between the end of the first pass and the start of the second pass point matching processes, since the tasks in between are executed on PPE. This however does not cause any performance problem because those idle threads remain in the SPE spaces, which essentially has no impact to the main PPE thread.

Both the main core Power Processing Element (PPE) and SPEs are kept busy in sharing the point matching tasks. Initially we partitioned the landmark points into clusters and distribute evenly the clusters to the available SPEs. The main PPE thread waits for the completion of all active SPE threads before it proceeds to the next step of the registration process. Apparently this design overlooked the PPE computing resource, though it is indeed a different type of processor from SPE processors. It turns out that for some images, there may exist situations in which the sub-image enclosing the cluster is a little bit too large to send to a SPE by one DMA command and the space has to be further partitioned and then transferred to the SPE in more than one step. Even though such cases are rare, the second round partition is still critical to speed up the whole registration process.

The purpose of applying  $K$ -means clustering for data partitioning is to decrease the number of direct memory access (DMA) operations. In order to fully utilize each SPE

Table 6.1: The break points for the three algorithms

	Translation(Voxel)	Scale	Rotation(degree)
ITK	>20	>1.4	>15
MedINRIA	>20	>1.3	>15
Our algorithm	> <b>30</b>	> <b>2.5</b>	> <b>45</b>

computing unit, the work load should be balanced. Because our algorithm selects the landmarks considering their spatial relationships, the  $K$ -means clustering intends to provide similar amount of landmarks in each cluster. In Figure 6.10 we show a typical work load distribution for one image pair on 16 SPE, it is clear that the number of landmarks assigned to each cluster roughly form a uniform distribution.

#### 6.4.2 Landmarks assembling and transformation estimation

The main processor or main core (e.g. PPE) is responsible to spool and destroy all the threads of computing cores (e.g. SPE). As we shown in Figure 6.9a, the PPE is also in charge of assembling all the matching points returned from each SPE and converts the results back to the original image coordinate systems. Robust estimation is applied to reject outliers and preserve the robust landmark correspondence. Nonlinear transformation is finally estimated to register the fixed image and the moving image.

### 6.5 Experimental Results

We first test our algorithm using the simulated affine transformations. Forty simulated 3D human abdomen MRI images are generated by applying forty simulated deformations. The algorithm is compared with the multiple resolution affine registration implemented in ITK [66] and also the free software MedINRIA developed by INRIA [95]. The registration accuracy is evaluated based on whether the algorithm can successfully recover the affine transformation parameters.

$$E = \max \left\{ \frac{|p_t^* - p_t|}{\delta_t}, \frac{|p_r^* - p_r|}{\delta_r}, \frac{|p_s^* - p_s|}{\delta_s} \right\}. \quad (6.5)$$

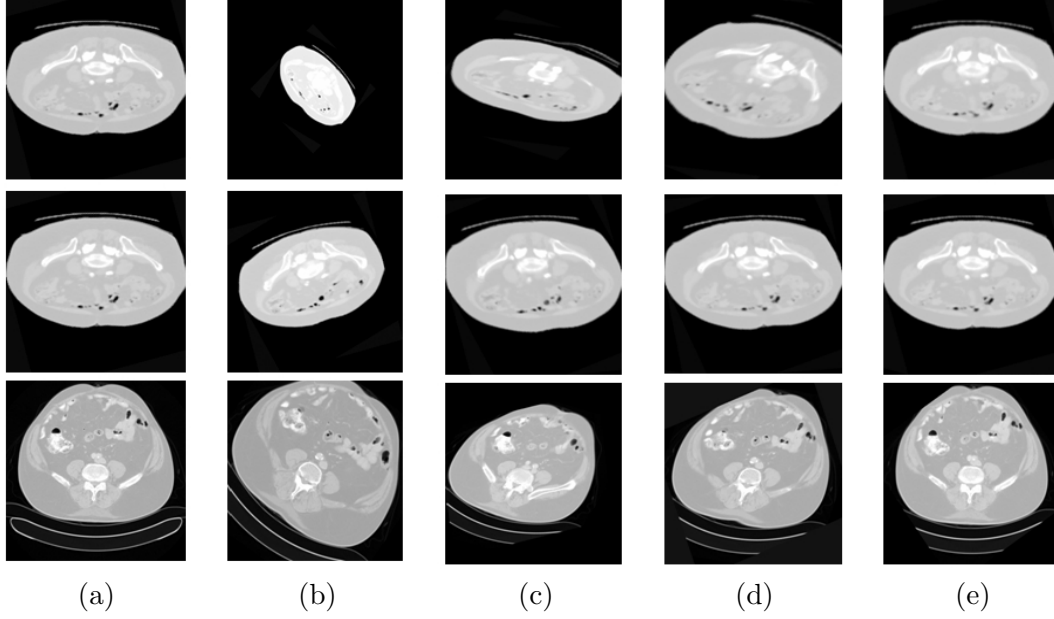


Figure 6.11: Comparative registration results on the human abdomen CT image pair. (a) The fixed image. (b) The moving image. (c) The registration algorithm using the MedINRIA [95]. (d) The registration algorithm using ITK [66]. (e) The registration results using our algorithm.

The  $p_t^*, p_r^*, p_s^*$  represent the estimated translation, rotation and scale parameters. The  $p_t, p_r, p_s$  are the ground true transformation parameters. The  $\delta_t = 1, \delta_r = 0.5, \delta_s = 0.01$  are the normalization factors for translation, rotation and scale, respectively. The registration is treated as success if  $E \leq 1.0$ . Our proposed algorithm can recover 95% of the image pairs while the ITK and MedINRIA can only recover 70% and 50% for the image pairs with large deformation. We also list the parameters which will fail the algorithms (break points) for all three methods in Table 6.1. From the experimental results we show that the proposed algorithm can accurately register two images even with 2.5 times scale difference and 45 degree of rotation. It is clear from this study that our algorithm demonstrate more robust registration for large deformations compared with commonly used registration algorithm [66, 95]. Some representative results are shown in Figure 6.11.

The parallelization code was built for two different platforms. The parallel version was tested on an IBM BladeCenter Q21 featuring 2GB of RAM and two processors running at 3.2 GHz configured as a two-way, symmetric multiprocessor (SMP) [100].

Table 6.2: The statistics of the running speed for 10 trials using the sequential and parallel multicore platform

	Mean	Variance	Median
x86	5.95	0.26	5.82
Cell/B.E. (PPE only)	6.13	0.001	6.12
Cell/B.E. (16 SPEs)	<b>0.60</b>	<b>0.0067</b>	<b>0.60</b>
	Min	Max	80%
x86	5.33	6.93	6.22
Cell/B.E. (PPE only)	6.12	6.14	6.13
Cell/B.E. (16 SPEs)	<b>0.50</b>	<b>0.70</b>	<b>0.70</b>

A thread running on a PPE can communicate with all 16 SPEs. The Cell/B.E. SDK 3.0 and GCC compiler were used to implement and compile the algorithm. In all our experiments, there was one main thread running on one of the PPEs and up to 16 threads on the SPEs. The sequential version was running on an x86 machine at 2.6 GHz and 4G memory. The compiler is also GCC.

All the image pairs used for testing have dimensionality  $(192 \times 192)$ . Because the point matching procedure dominates the running time of the registration algorithm, this step is the only part parallelized on the Cell/B.E. For fair comparison we run each implementation (sequential and parallel) 10 times. The statistics of the running speed on two platforms are shown in Table 6.2. Please notice that x86 refers to the sequential implementation on a x86 machine running with Linux. The Cell/B.E. (PPE only) denotes the running time on the multicore Cell processor using only the main processor PPE. The Cell/B.E. (PPE only) represents the parallel running time by fully utilizing all the 16 computing cores (SPEs). The 80% column in Table 6.2 represents the sorted 80% smallest running times of all 10 trials, and is commonly used to evaluate the performance of the system. Using the multicore platform, we roughly achieved 10 times of speedup over its corresponding sequential implementation. In total, the parallel version of the algorithm can register a pair of image  $(192 \times 192)$  in less than five seconds.

## 6.6 Conclusion

In this chapter, we have described a robust and accurate 2D/3D image registration algorithm. The 2D version of the image registration algorithm is implemented on the IBM Cell/B.E. We have achieved about 10 times speed up, which allows our algorithm to complete the nonlinear registration of a pair of images ( $192 \times 192$ ) in less than five seconds. Our proposed data partitioning approach and the parallelization schema are independent to the parallel platforms and generic by its design, therefore it can be extended to other point matching related applications and other parallel platforms. The work discussed here is a part of an ongoing effort in developing full scale parallelization of a set of registration algorithms including the one reported in this paper. Future work will also include the generalization of the parallelization algorithm to handle 3D images.

## Chapter 7

### Conclusions

This thesis proposed several novel algorithms for multiple class image segmentation. For the blood smear image, the  $L_2E$  robust estimation and robust color GVF snake are applied to delineate both the nuclear and cytoplasm of the imaged cells. A multiple object segmentation algorithm based on concave vertex graph is proposed to separate touching cells. A variational framework using coarse to fine shape alignment and semi-parametric density approximation are proposed to handle occluded image segmentation.

We also present an object-based segmentation on a large number of objects. Using mean-shift patches, the multiple class object-based segmentation is achieved by combining top-down and bottom-up information. A novel affine invariant descriptor is proposed to model the spatial configurations of the landmarks. The algorithm is computationally efficient and has been tested for three real datasets where the largest one contains 21 different classes.

Tracking the left ventricle (LV) in 3D ultrasound data is a challenging task because of the poor image quality and speed requirements. Many previous algorithms applied standard 2D tracking methods to tackle the 3D problem. However, the performance is limited due to increased data size, landmarks ambiguity, signal drop-out or non-rigid deformation. We present a robust, fast and accurate 3D LV tracking algorithm in this thesis. A novel one step forward prediction is utilized to generate the motion prior using motion manifold learning. Collaborative trackers are integrated to achieve both temporal consistency and failure recovery.

Next we also introduce a Grid-enabled computer aided diagnosis system to perform automatic analysis of imaged histopathology breast tissue specimens. Both linear and nonlinear dimension reduction techniques are compared, and the best one (ISOMAP)

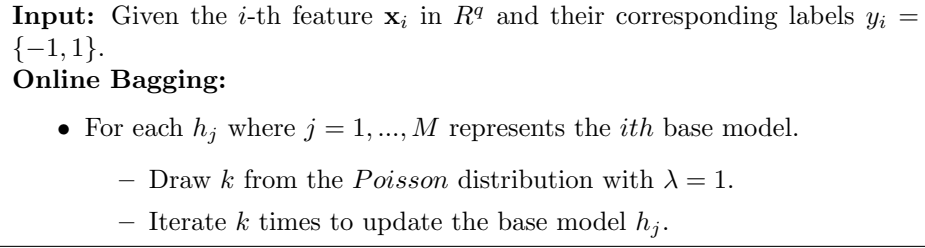


Figure 7.1: The online bagging algorithm.

was applied to reduce the dimensionality of the features. The experimental results show that the Gentle Boosting using an eight node decision tree as the weak learner provides the best result for classification.

Finally we present an image registration algorithm for both 2D and 3D images. This hybrid framework combine the advantages of both region based and point based approach. The parallelization of the algorithm is further implemented on an IBM Cell Broadband Engine. The speed up version of the algorithm can register two 2D image pairs ( $128 \times 128$ ) in less than 5 seconds.

## 7.1 Direction for Future Research

### 7.1.1 Online Learning for Object Recognition and Tracking

Boosting is one of the most successful machine learning algorithms in recent literature. However, most of the current boosting algorithm is trained on batch mode. This is not the case for many real applications because the entire training set may not be available at the training time. It might be impossible or inconvenient to retrain all the classifiers when new training samples arrive. Online learning algorithms can process each training sample when it arrives individually. It also requires the classifiers to reflect the properties learned from the previous training samples.

The idea of online boosting is generated from the online bagging [108]. Given a training set  $T$  with  $N$  training samples  $\{\mathbf{x}_1, y_1\}, \dots, \{\mathbf{x}_N, y_N\}$ . The  $\mathbf{x}_i \in R^q$  denotes the  $i$ -th feature vector and  $y_i \in -1, 1$  represents the corresponding ground-truth class labels. The standard bagging creates  $M$  models. Each model is trained on a bootstrap

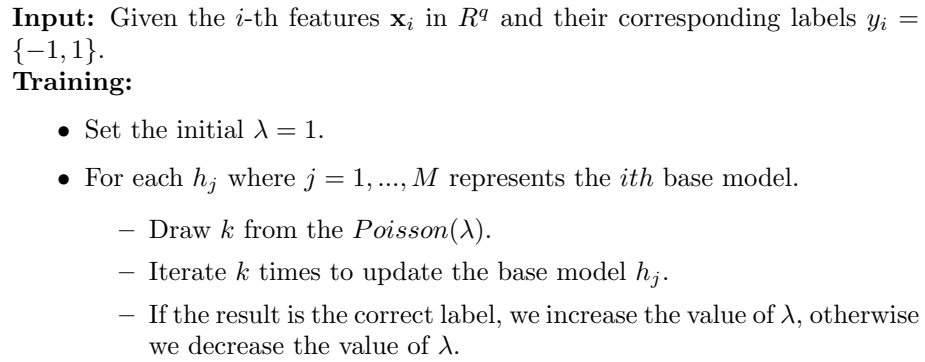


Figure 7.2: The online boosting algorithm.

sample of size  $L$  which is randomly sampled from  $T$  with replacement.

Each base model contains  $K$  copies of the original training sample which is a *Binomial* distribution

$$p(K = k) = \binom{N}{K} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{N-k}. \quad (7.1)$$

It is well known that the *Poisson* distribution can be derived as a limited case of the *Binomial* distribution. When  $N \rightarrow \infty$ , the distribution of  $K$  tends to become a *Poisson* distribution.

$$p(K = k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (7.2)$$

where  $\lambda$  represents the *Poisson* distribution parameter. In the online bagging, as each training sample arrived, we can make a draw from the *Poisson* distribution and update the based model accordingly. The online bagging algorithm is listed in Figure 7.1. Similar idea can be extended to boosting. Instead of train the weak learner directly on the weighted training samples, we can draw samples according to Poisson distribution with the parameter  $\lambda$  modified by the classification errors as shown in Figure 7.2.

In online boosting, it might not be difficult to estimate the corresponding voting weights because we know the classification errors of the weak learner, although the error is only based on the current training sample instead of the whole set in the off-line version. The major difficulty comes from the weight distribution for the training samples because we disregard all the previous samples in the online version. Assume we have a training set which contains 1000 samples. In original AdaBoost, the classifier



is trained on all the 1000 samples before it is tested on, e.g., the 100<sup>th</sup> testing sample. In the online version of AdaBoost, the classifier is only trained on the first 99 training samples when it is tested on the 100<sup>th</sup> test sample. If there exists a large nonlinearity between the first 99 training samples and the rest of the training set, the weak learners will be quite different. The offline classifier can be treated as the global optimal solution on the whole training set while the online classifier is only locally optimal.

Since the members of the final classifier is not fixed after the initial learning period, the online boosting is able to adapt rapidly to non-stationary environment. However, it is not able to perform better than offline boosting using a fixed set of training samples in general [108]. It is worth to research on the online learning algorithm which can produce close performance to offline boosting.

### 7.1.2 Registration Based Heart Torsion Modeling

The motion of the myocardial region in the left ventricles can be critical to decide the health condition of the heart. Speckle tracking is widely used to track the motion in the myocardial region in the left ventricles of the human hearts. However, there are a lot randomness if we track the speckles independently. In order to keep the correspondence between frames in tracking, the registration algorithm proposed in Chapter 7 can be applied to constrain the torsion type of motion. If we decomposed the motion of the left ventricles as contraction and torsion. The torsion part can be modeled as on surface movement which is perpendicular to the contract direction. By applying mesh registration between adjacent frames, it can be shown that the torsion can be modeled accurately.

## References

- [1] P. S. U. Adiga and B. B. Chaudhuri. An efficient method based on watershed and rule-based merging for segmentation of 3D histo-pathological images. *Pattern Recognition*, 34(7):1449–1458, 2001.
- [2] E. Agostino, F. Maes, D. Vandermeulen, and P. Suetens. A viscous fluid model for multimodal non-rigid image registration using mutual information. *Medical Image Analysis*, 7(4):565–575, 2003.
- [3] American Cancer Society. *Cancer Facts and Figures 2007*. American Cancer Society, 2007 edition, 2007.
- [4] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.
- [5] S. Avidan. Ensemble tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.
- [6] A. Barbu, V. Athitsos, B. Georgescu, S. Boehm, P. Durlak, and D. Comaniciu. Hierarchical learning of curves application to guidewire localization in fluoroscopy. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.
- [7] F. Berman, G. Fox, and A. J. G. Hey. *Grid Computing: Making the Global Infrastructure a Reality*. Wiley, first edition, 2003.
- [8] E. Borenstein and J. Malik. Shape guided object segmentation. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 969–976, New York, NY, 2006.
- [9] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC.
- [10] G. Buehrer, S. Parthasarathy, and M. Goyder. Data mining on the cell broadband engine. *International Conference on Supercomputing*, pages 26–35, 2008.
- [11] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):1027–1037, 2002.
- [12] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–32, 1993.
- [13] B. Cena, N. Fox, and J. Rees. Fluid deformation of serial structural MRI for low-grade glioma growth analysis. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 3217, pages 1055–1063, Rennes, France, 2004.

- [14] T. F. Chan and L. A. Vese. A level set algorithm for minimizing the mumford-shah functional in image processing. In *Proc. IEEE Workshop on Variational and Level Set Methods*, pages 161–171, Vancouver, Canada, 2001.
- [15] T. Chen, R. Raghavan, J. N. Dale, and E. Iwata. Cell broadband engine architecture and its first implementation - a performance review. *IBM J. Res. Develop.*, 51(5):559–572, 2007.
- [16] W. Chen, M. Reiss, and D. J. Foran. Unsupervised tissue microarray analysis for cancer research and diagnosis. *IEEE Trans. Information Technology in Biomedicine*, 8(2):89–96, 2004.
- [17] Y. Chen, H. D. Tagare, S. Thiruvankadam, F. Huang, D. Wilson, A. Geiser, K. Gopinath, and R. Briggs. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.
- [18] D. Chetverikov and Z. Szabó. A simple and efficient algorithm for detection of high curvature points in planar curves. In *The 23rd Workshop of the Austrian Pattern Recognition Group*, pages 175–184, Vienna, Austria, 1999.
- [19] N. Chrisochoides, A. Fedorov, A. Kot, N. Archip, P. Black, O. Clatz, A. Golby, R. Kikinis, and S. Warfield. Toward real-time, image guided neurosurgery using distributed and grid computing. *ACM/IEEE Super Computing*, 2006.
- [20] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.
- [21] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding Archive*, 53(2):211–218, 1991.
- [22] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.
- [23] D. Comaniciu, P. Meer, and D. Foran. Image-guided decision support system for pathology. *Machine Vision and Applications*, 11:213–224, 1999.
- [24] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [25] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [26] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.
- [27] D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69:335–351, 2006.
- [28] O. Cula and K. Dana. 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1), 2004.
- [29] S. Dambreville, Y. Rathi, and A. Tannenbaum. Shape-based approach to robust image segmentation using kernel PCA. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 977–984, New York, NY, 2006.

- [30] M. Dewan, C. H. Lorenz, and G. D. Hager. Deformable motion tracking of cardiac structures (DEMOTRACS) for improved MR imaging. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.
- [31] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1964–1971, New York, NY, 2006.
- [32] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, 1998.
- [33] Q. Duan, E. Angelini, S. Homma, and A. Laine. Validation of optical-flow for quantification of myocardial deformations on simulated RT3D ultrasound. In *Proc. International Symposium on Biomedical Imaging*, pages 944–947, Washington, DC, 2007.
- [34] R. Duan and A. Strey. Data mining algorithms on the cell broadband engine. *Euro-Par 2008*, 5168:665–675, 2008.
- [35] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2000.
- [36] R. Durikovic, K. Kaneda, and H. Yamashita. Dynamic contour: A texture approach and contour operations. *The Visual Computer*, 11:277–289, 1995.
- [37] J. H. Elder and R. M. Goldberg. Ecological statistics of Gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353, 2002.
- [38] R. P. Fedkiw, G. Sapiro, and C. Shu. Shock capturing, level sets, and PDE based methods in computer vision and image processing: A review of Osher’s contributions. *Journal of Computational Physics*, 185(2):309–341, 2003.
- [39] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition Workshop on Generative-Model Based Vision*, volume 12, pages 178–186, Washington, DC, 2004.
- [40] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.
- [41] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, Madison, WI, 2003.
- [42] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.
- [43] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):1–12, 2004.
- [44] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. *Machine Learning*, pages 148–156, 1996.
- [45] B. J. Gajewski, L. R. Rilett, M. P. Dixon, and C. H. Spiegelman. Robust estimation of origin-destination matrices. *Journal of Transportation and Statistics*, 5:35–49, 2002.

- [46] R. Galloway. Texture analysis using gray level run lengthes. *Computer Graphics and Image Processing*, 4:172–179, 1975.
- [47] B. Gedik, R. R. Bordawekar, and P. S. Yu. Celsort: high performance sorting on the cell processor. *The 33rd international conference on very large databases*, pages 1286–1297, 2007.
- [48] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proc. IEEE International Conference on Computer Vision*, pages 456–463, Nice, France, 2003.
- [49] T. Gevers. Adaptive image segmentation by combining photometric invariant region and edge information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(6):848–852, 2002.
- [50] T. Gevers, S. Ghebreab, and A. M. Smeulders. Color invariant snakes. In *Proc. Ninth British Machine Vision Conference*, pages 659–670, Southampton, UK, 1998.
- [51] L. V. Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In *Proc. European Conference on Computer Vision*, volume 4, pages 642–651, Cambridge, UK, 1996.
- [52] P. F. U. Gotardo, K. L. Boyer, J. Saltz, and S. V. Raman. A new deformable model for boundary tracking in cardiac MRI and its application to the detection of intra-ventricular dyssynchrony. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 736–743, New York, NY, 2006.
- [53] I. S. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and Products*. Academic Press, 5th edition, 1994.
- [54] D. Gu, L. Yang, and L. R. Welch. A predictive, decentralized load balancing approach. In *Proc. International Parallel and Distributed Processing Symposium*, Denver, CA, 2005.
- [55] W. R. Hamilton. Elements of quaternions. 1969.
- [56] R. Haralik, K. Shanmugan, and I. Dinstein. Texture features for image classification. *IEEE Trans. on System, Man and Cybernetics*, 3:610–621, 1973.
- [57] X. He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 695–702, Washington, DC, 2004.
- [58] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. In *Proc. ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pages 229–238, Los Angeles, CA, 1995.
- [59] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Physical Medical Biology*, 46:1–45, 1998.
- [60] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2137–2144, New York, NY, 2006.
- [61] B. Hong, E. Prados, L. A. Vese, and S. Soatto. Shape representation based on integral kernels: Application to image matching and segmentation. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 833–840, New York, NY, 2006.

- [62] W. Hong, B. Georgescu, X. S. Zhou, S. Krishnan, Y. Ma, and D. Comaniciu. Database-guided simultaneous multi-slice 3D segmentation for volumetric data. In *Proc. European Conference on Computer Vision*, volume 4, pages 397–409, Graz, Austria, 2006.
- [63] IBM Help Defeat Cancer. <http://pleiad.umdj.edu/ibm>. 2007.
- [64] IBM World Community Grid. <http://www.worldcommunitygrid.org>. 2007.
- [65] M. Isard and A. Blake. CONDENSATION — Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 28(1):5–28, 1998.
- [66] ITK Software Guide. <http://www.itk.org>.
- [67] S. Jehan-Besson, M. Gastaud, M. Barlaud, and G. Aubert. Region-based active contours using geometrical and statistical features for image segmentation. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 643–646, Barcelona, Spain, 2003.
- [68] M. P. Jolly. Automatic segmentation of the left ventricles in cardiac MR and CT images. *International Journal of Computer Vision*, 70(2):151–163, 2006.
- [69] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy. Introduction to the cell multiprocessor. *IBM J. Res. Develop.*, 49(4):589–604, 2005.
- [70] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [71] Z. Khan, T. Balch, and F. Dellaert. A MCMC-based particle filter for tracking multiple interacting targets. In *Proc. European Conference on Computer Vision*, volume 4, pages 279–290, Prague, Czech Republic, 2004.
- [72] D. Kim and B. Bodt. Building a robust explanatory model from a large dataset. In *Ninth U.S. Army Conference on Applied Statistics*, volume 2, pages 98–108, Davis, CA, 2003.
- [73] E. Kreyszig. *Differential Geometry*. Dover Publications, 1st edition, 1991.
- [74] F. P. Kuhl and C. R. Giardina. Elliptic Fourier features of a closed contour. *Computer Graphics and Image Processing*, 18:236–258, 1982.
- [75] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 18–25, San Diego, CA, 2005.
- [76] K. I. Laws. *Texture Image Segmentation*. Ph.D Thesis, University of Southern California, 1980.
- [77] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, NY, 2006.
- [78] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision*, volume 1, pages 17–32, Prague, Czech Republic, 2004.

- [79] K. Lekadir, D. S. Elson, J. Requejo-Isidro, C. Dunsby, J. McGinty, N. Galletly, G. Stamp, P. M. French, and G.-Z. Yang. Tissue characterization using dimensionality reduction and fluorescence imaging. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4191, pages 586–593, Copenhagen, Denmark, 2006.
- [80] H. Lester and S. R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32(1):129–149, 1999.
- [81] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [82] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 316–323, Hilton Head, SC, 2000.
- [83] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 53–60, Washington, DC, 2004.
- [84] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. European Conference on Computer Vision*, volume 4, pages 581–594, Graz, Austria, 2006.
- [85] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):617–634, 1993.
- [86] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.
- [87] J. Lim and M. Yang. A direct method for modeling non-rigid motion with thin plate spline. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1196–1202, San Deigo, CA, 2005.
- [88] J. Liu, B. C. Vemuri, and J. L. Marroquin. Local frequency representation for robust multi-modal image registration. *IEEE Trans. Medical Imaging*, 21(5):462–469, 2002.
- [89] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [90] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [91] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.
- [92] J. Mao and A. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25:173–188, 1992.
- [93] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2118–2125, New York, NY, 2006.

- [94] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(6):810–815, 2004.
- [95] MedINRIA. <http://www-sop.inria.fr/asclepios/software/medinria/>.
- [96] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [97] T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, 1st edition, 2000.
- [98] O. Musse, F. Heitz, and J. P. Armspach. Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Trans. Image Processing*, 10(7):1081–1093, 2002.
- [99] A. Myronenko, X. B. Song, and D. J. Sahn. LV motion tracking from 3D echocardiography using textural and structural information. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4792, pages 428–435, Brisbane, Australia, 2007.
- [100] A. K. Nanda, J. R. Moulic, R. E. Hanson, G. Goldrian, M. N. Day, B. D. D’Amora, and S. Kesavarapu. Cell/B.E. blades: Building blocks for scalable, read-time, interactive and digital media servers. *IBM J. Res. Develop.*, 51(5):573–582, 2007.
- [101] V. Noblet, C. Heinrich, F. Heitz, and J. P. Armspach. 3-D deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization. *IEEE Trans. Image Processing*, 14(5):553–566, 2005.
- [102] M. Ohara, H. Yeo, F. Savino, L. Gong, H. Inoue, V. Sheinin, S. Daijavad, and B. Erickson. Realtime mutual information based linear registration on the cell broadband engine processor. In *Proc. International Symposium on Biomedical Imaging*, 2007.
- [103] A. Oliver, J. Freixenet, R. Marti, and R. Zwigelaar. A comparison of breast tissue classification techniques. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4191, pages 872–879, Copenhagen, Denmark, 2006.
- [104] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. European Conference on Computer Vision*, volume 1, pages 71–84, Prague, Czech Republic, 2004.
- [105] P. Orbanz and J. M. Buhmann. Smooth image segmentation by nonparametric Bayesian inference. In *Proc. European Conference on Computer Vision*, volume 1, pages 444–457, Graz, Austria, 2006.
- [106] F. Orderud, J. Hansgård, and S. I. Rabben. Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4791, pages 858–865, Brisbane, Australia, 2007.
- [107] D. Ormoneit, M. J. Black, T. Hastie, and H. Kjellström. Representing cyclic human motion using functional analysis. *Image and Vision Computing*, 23(14):1264–1276, 2005.
- [108] N. C. Oza and S. Russell. Online bagging and boosting. *Artificial Intelligence and Statistics*, pages 105–112.



- [109] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Trans. Medical Imaging*, 22(8):986–1004, 2003.
- [110] A. Rabinovich, S. Belongie, T. Lange, and J. M. Buhmann. Model order selection and cue combination for image segmentation. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1130–1137, New York, NY, 2006.
- [111] T. R. Raviv, N. Kiryati, and N. Sochen. Segmentation by level sets and symmetry. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1015–1022, New York, NY, 2006.
- [112] N. Ray, S. T. Acton, T. Altes, and E. E. de Lange. MRI ventilation analysis by merging parametric active contours. In *Proc. IEEE International Conference on Image Processing*, pages 861–864, Thessaloniki, Greece, 2001.
- [113] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 824–831, Beijing, China, 2005.
- [114] X. Ren, C. Fowlkes, and J. Malik. Cue integration for figure/ground labeling. In *Neural Information Processing Systems*, Whistler, Canada, 2005.
- [115] T. Rohlfing, C. R. Maurer, D. A. Bluemke, and M. A. Jacobs. Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint. 22:730–741, 2003.
- [116] J. Rosai. *Rosai and Ackerman's Surgical Pathology*. Mosby, nine edition, 2004.
- [117] S. T. Rowels and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [118] M. Rudemo. Empirical choice of histogram and kernel density estimators. *Scandinavian Journal of Statistics*, 1:65–78, 1982.
- [119] G. Sapiro and D. L. Ringach. Anisotropic diffusion on multivalued images with applications to color filtering. *IEEE Trans. Image Processing*, 5(11):1582–1586, 1996.
- [120] D. W. Scott. Remarks on fitting and interpreting mixture models. *Computer Science and Statistics*, 31:104–109, 1999.
- [121] D. W. Scott. Parametric statistical modeling by minimum integrated square error. *Technometrics*, 43:274–285, 2001.
- [122] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, first edition, 2004.
- [123] X. Shen and L. Yang. *Handbook of medical image analysis: Segmentation and registration models*. Kluwer, first edition, 2005.
- [124] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [125] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. European Conference on Computer Vision*, volume 1, pages 1–13, Graz, Austria, 2006.

- [126] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 370–377, Beijing, China, 2005.
- [127] J. S. Stahl and S. Wang. Convex grouping combining boundary and region information. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 946–953, Beijing, China, 2005.
- [128] J. S. Suri and R. M. Rangayyan. *Recent Advances in Breast Imaging, Mammography, and Computer-Aided Diagnosis of Breast Cancer*. SPIE, 2006.
- [129] R. Szeliski, R. Szeliski, J. Coughlan, and J. Coughlan. Hierarchical spline-based image registration. *International Journal of Computer Vision*, pages 194–201, 1994.
- [130] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [131] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.
- [132] D. Terzopoulos and R. Szeliski. *Tracking With Kalman Snakes*. MIT-Press, first edition, 1992.
- [133] J. P. Thirion. Image matching as a diffusion process: An analogy with Maxwell demons. *Medical Image Analysis*, 2(3):243–260, 1998.
- [134] T. Toyoda, K. Tagami, and O. Hasegawa. Integration of top-down and bottom-up information for image labeling. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1106–1113, New York, NY, 2006.
- [135] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 1589–1596, Beijing, China, 2005.
- [136] O. Tuzel, L. Yang, P. Meer, and D. J. Foran. Classification of hematologic malignancies using texton signatures. *Pattern Analysis and Applications*, 2007.
- [137] U. S. Cancer Statistics Working Group. United states cancer statistics: 2003 incidence and mortality (preliminary data). *National Vital Statistics*, 53(5), 2004.
- [138] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3D human body tracking. *Computer Vision and Image Understanding*, 104:157–177, 2006.
- [139] A. L. Varbanescu, A. S. van Amesfoort, T. Cornwell, A. Mattingly, B. G. Elmegreen4, R. van Nieuwpoort, G. van Diepen, and H. Sips. Radioastronomy image synthesis on the Cell/B.E. *Euro-Par 2008*, 5168:749–762, 2008.
- [140] M. Varma and A. Zisserman. Classifying images of materials: Achieving view-point and illumination independence. In *Proc. European Conference on Computer Vision*, volume 3, pages 255–271, Copenhagen, Denmark, 2002.
- [141] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4792, pages 319–326, Brisbane, Australia, 2007.

- [142] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, Kauai Island, HI, 2001.
- [143] J. Wang, P. Bhat1, A. Colburn, M. Agrawala, and M. Cohen. Interactive video cutout. In *Proc. ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques*, volume 1, pages 585–594, Los Angeles, CA, 2005.
- [144] S. K. Warfield, F. Jolesz, and R. Kikinis. A high performance approach to the registration of medical imaging data. *Parallel Computing*, 24(9):1345–1368, 1998.
- [145] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing: Image Understanding Archive*, 55(1):855–867, 1990.
- [146] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 1800–1807, Beijing, China, 2005.
- [147] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 37–44, New York, NY, 2006.
- [148] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, 2nd edition, 2000.
- [149] C. Xu and J. L. Prince. Snakes, shapes and gradient vector flow. *IEEE Trans. Image Processing*, 7(3):359–369, 1998.
- [150] L. Yang, W. Chen, P. Meer, G. Salaru, M. D. Feldman, and D. J. Foran. High throughput analysis of breast cancer specimens on the grid. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4791, pages 617–625, Brisbane, Australia, 2007.
- [151] L. Yang, W. Chen, P. Meer, G. Salaru, L. A. Goodell, V. Berstis, and D. J. Foran. Virtual microscopy and grid-enabled decision support for large scale analysis of imaged pathology specimens. *Accepted in IEEE Trans. Information Technology in Biomedicine*, 2008.
- [152] L. Yang and D. J. Foran. *Parametric and Geometric Deformable Models: An Application in Biomaterials and Medical Imaging*. Springer-Verlag New York, first edition, 2007.
- [153] L. Yang and D. J. Foran. A variational framework for partially occluded image segmentation using coarse to fine shape alignment and semi-parametric density approximation. In *Proc. IEEE International Conference on Image Processing*, volume 1, pages 37–40, San Antonio, TX, 2007.
- [154] L. Yang, B. Georgescu, Y. Zheng, D. J. Foran, and D. Comaniciu. A fast and accurate tracking algorithm of left ventricles in 3D echocardiography. In *Proc. International Symposium on Biomedical Imaging*, Paris, France, 2008.
- [155] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu. 3D ultrasound tracking of the left ventricles using one-step forward prediction and data fusion of collaborative trackers. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

- [156] L. Yang, J. Liu, L. R. Welch, and C. D. Cavanaugh. A  $L_2E$ -based QoS forecasting technique for dynamic, distributed real-time systems. In *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, 2003.
- [157] L. Yang, J. Liu, L. R. Welch, and C. D. Cavanaugh. A robust QoS forecasting technique for dynamic distributed real-time testbed. volume 1, pages 35–43, New Orleans, LA, 2003.
- [158] L. Yang, P. Meer, and D. Foran. Unsupervised segmentation based on robust estimation and color active contour models. *IEEE Trans. Information Technology in Biomedicine*, 9:475–486, 2005.
- [159] L. Yang, P. Meer, and D. J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.
- [160] L. Yang, O. Tuzel, W. Chen, P. Meer, G. Salaru, L. A. Goodell, and D. J. Foran. Pathminer: A web-based telemicroscopy, intelligent archiving and image analysis system. *Accepted in IEEE Trans. Information Technology in Biomedicine*, 2008.
- [161] L. Yang, O. Tuzel, P. Meer, and D. J. Foran. Touching cells segmentation in hematologic specimens using concave vertex graph. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 5241, pages 833–841, New York, NY, 2008.
- [162] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computer Survey*, 38(4):13.1–13.45, 2006.
- [163] S. D. Zenzo. *A Note on the Gradient of a Multi-Image*. Oxford University Press, 3rd edition, 1986.
- [164] H. Zha and Z. Zhang. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Science Computation*, 26(1):313–338, 2004.
- [165] T. Zhao and R. Nevatia. 3D tracking of human locomotion: A tracking as recognition approach. In *Proc. IEEE International Conference on Pattern Recognition*, pages 1054–1060, Quebec City, Canada, 2002.
- [166] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Fast automatic heart chamber segmentation from 3D CT data using marginal space learning and steerable features. In *Proc. IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.
- [167] Y. Zhu, X. Papademetris, A. Sinusas, and J. S. Duncan. Segmentation of myocardial volumes from real-time 3D echocardiography using an incompressibility constraint. In *Proc. International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 4791, pages 44–51, Brisbane, Australia, 2007.
- [168] B. Zitova and J. Flusser. Image registration methods: A survey. *Image Vision Computing*, 21(11):977–1000, 2003.

## Vita

### Lin Yang

- 1999** B. Sc., Department of Information and Communication Engineering, Xian Jiaotong University, P.R. China.
- 2002** M. Sc., Department of Information and Communication Engineering, Xian Jiaotong University, P.R. China.
- 2006** M. Sc., Department of Electrical and Computer Engineering, Rutgers – The State University of New Jersey.
- 2009** Ph. D., Department of Electrical and Computer Engineering, Rutgers – The State University of New Jersey.

### Publications

**Lin Yang**, Oncel Tuzel, Wenjin Chen, Peter Meer, Gratian Salaru, Lauri A. Goodell and David J. Foran, "PathMiner: A web-based telemicroscopy, intelligent archiving and image analysis system", *IEEE Transaction on Information Trachnology in Biomedicine*, 2008, In press.

**Lin Yang**, Wenjin Chen, Peter Meer, Gratian Salaru, Viktors Berstis and David J. Foran, "Virtual Microscopy and Grid-enabled Decision Support for Large Scale Analysis of Imaged Pathology Specimens", *IEEE Transaction on Informtion Trachnology in Biomedicine*, 2008, In press.

**Lin Yang**, Bogdan Georgescu, Yefeng Zheng, David J. Foran and Dorin Comaniciu, "A Fast and Accurate Tracking Algorithm of Left Ventricles In 3D Echocardiography", *Proc. International Symposium on Biomedical Imaging (ISBI)*, Paris, France, May, 2008.

**Lin Yang**, Oncel Tuzel, Peter Meer and David J. Foran, "Automatic Image Analysis of Histopathology Specimens Using Concave Vertex Graph", *The 11th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, New York, U. S. A., September 2008.

**Lin Yang**, Bogdan Georgescu, Yefeng Zheng, Peter Meer and Dorin Comaniciu "3D Ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers", *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, June 2008.

**Lin Yang**, Wenjin Chen, Peter Meer, Gratian Salaru, Michael D. Feldman and David J. Foran "High Throughput Breast Cancer Analysis on the Grid", *The 10th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Brisbane, Australia, November 2007.

**Lin Yang**, Peter Meer and David J. Foran, "Multiple Class Segmentation Using a Unified Framework over Mean-shift Patches", *The 2007 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, June 2007.

Oncel Tuzel, **Lin Yang**, Peter Meer and David J. Foran, "Classification of hematologic malignancies using texton signatures", *Pattern Analysis and Applications*, Feb. 2007.

**Lin Yang** and David J. Foran "A Variational Framework For Partially Occluded Image Segmentation Using Shape Priors and Semi-Parametric Density Approximation", *The 2007 IEEE International Conference on Image Processing (ICIP)*, San Antonio, Texas, September 2007.

**Lin Yang** and David J. Foran, *Parametric and Geometric Deformable Models: An Application in Biomaterials and Medical Imaging*. Suri/Farag, eds, Vol. 2, Chapter 4, pp. 95-122, Springer, ISBN: 0-387-31204-8, 2006.

**Lin Yang**, Peter Meer and David J. Foran. "A General Framework for Segmenting Imaged Pathology Specimens Using Level-set and Gaussian Hidden Markov Random Fields", *The 2006 Advance Practices, Instruction and Innovation through Informatics (APIII)*, Vancouver, PA, October, 2006.

**Lin Yang**, Peter Meer and David J. Foran, "Unsupervised Analysis of Imaged Pathology Specimens Based on Robust Estimation and Color Active Contour Models", *The IEEE Transaction on Information Technology in Biomedicine*, Vol. 9, pp. 475-486, 2005.

Xiaoping Shen and **Lin Yang**, *Handbook of Medical Image Analysis: Segmentation and Registration Models*, Suri/Wilson/Laxminarayan, eds, Vol. 1, Chapter 1, 47 pp., Kluwer, ISBN: 0306485508, 2005

Dazhang Gu, **Lin Yang** and Lonnie R. Welch. "A predictive, decentralized load balancing approach", *The 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Denver, Colorado, April 2005.

Mehmet Celenk and **Lin Yang**, "Tumor detection in VIVO NIRF imaging", *The SPIE International Symposium on medical imaging (SPIE Medical Imaging)*, February, San Diego, CA, 2004.

**Lin Yang**, Gratian Salaru, David J. Foran and Peter Meer, "Merging Robust Estimation with Color GVF snake : An unsupervised Image Guided Decision Support System", *The 2004 Advance Practices, Instruction and Innovation through Informatics (APIII)*, Pittsburgh, PA, October, 2004.

**Lin Yang**, Jundong Liu, Charles D. Cavanaugh and Lonnie R. Welch, "A  $L_2E$ -Based QoS Forecasting Technique for Dynamic, Distributed Real-time Systems", *The 2003 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, Las Vegas, U.S.A., June, 2003.

**Lin Yang**, Jundong Liu, Charles D. Cavanaugh and Lonnie R. Welch, "A Robust QoS Forecasting Algorithm for A Dynamic, Distributed Real-time Testbed", *The IEEE 2003 International Workshop on Computer Architectures for Machine Perception (CAMP)*, New Orleans, U.S.A., May, 2003