

2013

Towards More Desirable Segmentation via User Interactions

Yaoyao Zhu
Lehigh University

Follow this and additional works at: <http://preserve.lehigh.edu/etd>

Recommended Citation

Zhu, Yaoyao, "Towards More Desirable Segmentation via User Interactions" (2013). *Theses and Dissertations*. Paper 1374.

This Dissertation is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

Towards More Desirable Segmentation via User Interactions

by

Yaoyao Zhu

Presented to the Graduate and Research Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy

in

Computer Engineering



Lehigh University

January 2013

© 2013
Yaoyao Zhu
ALL RIGHTS RESERVED

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Date

Dr. XiaoLei Huang
Dissertation Director

Accepted Date

Committee Members:

Dr. Xiaolei Huang
Name of Committee Chair

Dr. Daniel Lopresti
Name of Committee Co-Chair

Dr. Henry Baird
Name of Committee Member

Dr. George Nagy
Name of Committee Member

Mr. Rodney Long
Name of Committee Member

To my family: Changwen Zhu, Qiaofeng Jiang, Linling Zhu and Yong Zhu

Acknowledgements

First and foremost, I am grateful to my two advisors Prof. Daniel Lopresti and Prof. Xiaolei Huang for their advice, encouragement and continuous support over my Ph.D. years at Lehigh. A thesis is the virtual act of connecting dots. They provided the dots.

Dr. Lopresti gave me the first idea to explore user interactions in image analysis systems. During all stages of my research, I learned a lot from his clear way of thinking with valuable comments and tough questions. Dr. Huang has been an excellent advisor and her guidance helps me in all the time of my research and writing this thesis. Without their help, this thesis would not be possible.

I would like to thank the rest of my doctoral committee: Prof. George Nagy, Prof. Henry Baird and Mr. Rodney Long, for their help and insightful comments and valuable questions regarding this thesis.

I thank Dr. Sameer Antani and Dr. George Thoma at the National Library of Medicine (NLM) for offering me the internship opportunities and valuable research experience at NLM.

My sincere thanks also go to Mr. P.P.S. Narayan , Ms. Chitra Phadke, and Dr. Viswanath Poosala of Bell Labs for hosting me as a summer intern and leading me to work on their Geopepper project.

I also would like to give special thanks to my labmates from the Image Data Emulation and Analysis Laboratory: Tian Shen, Hongsheng Li, Edward Kim, Wei Wang, Ting Xu and Sunhua Wan, for the stimulating discussions and for the days we worked together.

Last but not the least, thanks also go to my friends and my family, for all the support they have given to me.

Contents

List of Figures	xii
List of Tables	xvii
Abstract	1
Principal Contributions	3
1 Introduction	6
1.1 Desired Properties of Interactive Segmentation Methods	9
1.2 Review of Interactive Segmentation Methods	11
1.2.1 Active Contour Models	11
1.2.2 Graphical Models	19
1.2.3 Discussions	21
1.3 User interactions in Interactive Segmentation Methods	22
1.3.1 Setting or Changing Parameter Values	22
1.3.2 Pictorial Inputs on the Image Grid	23
1.3.3 Discussion	24
1.4 Proposed Interactive Segmentation Methods based on Different Models .	25

1.4.1	Motivation	25
1.4.2	Interactive Active Volume Models with Interactive Polygons	27
1.4.3	Interactive Localized Level Set	27
1.4.4	Interactive Segmentation through Label Propagations on Hierarchical Hypergraphs	28
1.4.5	Evaluation of Interactive Segmentation Methods based on Graphical Models	28
1.5	Thesis Outline	29
2	Interactive Polygons in Region-based Parametric Active Contour	30
2.1	Introduction	30
2.2	Active Volume Model	32
2.3	Motivation for Employing User Interaction	35
2.4	Integrating User Interaction to AVM	37
2.5	Experiments	41
2.6	Discussion	42
3	Interactive Level Set Method with User Scribbles	44
3.1	Introduction	44
3.2	Localized Region-based Level Set	47
3.3	Modeling Region Separation Energy for Nature Images	49
3.4	Integrating User Scribbles as Energy Function	51
3.4.1	Modeling User Scribbles as Observation	51
3.4.2	Integrating Scribbles into Level Set Framework	53
3.5	Experimental Results	55
3.6	Discussion	56

4	Interactive Segmentation Through Label Propagations on Hierarchical Hypergraph	58
4.1	Introduction	58
4.2	Literature Review	59
4.2.1	Usage of Scribbles and Limitations	59
4.2.2	Graph-based Segmentation Models	61
4.3	New Framework: Label Propagation on Hierarchical Hypergraphs	64
4.3.1	Hierarchical Hypergraph as Image Representation	65
4.3.2	Label Propagation on the Hierarchical Hypergraph	71
4.3.3	Overview of the Segmentation Framework	78
4.4	New Interaction for Topology Change: Connecting Dots	78
4.5	Scribble Recommendation	83
4.6	Experimental Results	85
4.6.1	Evaluation of Proposed Framework	85
4.6.2	Comparison with Other Methods	87
4.6.3	Visual Verification	88
4.6.4	Quantitative Verification	88
4.6.5	Evaluation of Scribble Recommendation System	89
4.7	Discussion	93
5	Evaluation of Interactive Segmentation Methods	94
5.1	Introduction	94
5.2	Literature Review	95
5.2.1	Quality Measures	95
5.2.2	Data Sets	99
5.2.3	Evaluation Methods	99

5.3	Data Set, Evaluation Criteria and Algorithms in Our New Framework	102
5.3.1	Data Set	102
5.3.2	Evaluation Criteria	102
5.3.3	Algorithms	104
5.4	User Studies in Our New Framework	105
5.4.1	Experiment Design	106
5.4.2	Questionnaire	106
5.5	Automatic Evaluation by Machine-generated Scribbles in Our New Frame- work	107
5.5.1	Generating Scribbles by Random Walks	108
5.5.2	Simulating User Interactions by Different Strategies	109
5.6	Evaluation Results and Analysis	110
5.6.1	Accuracy	110
5.6.2	Efficiency	112
5.6.3	Sensitivity to Scribbles	114
5.6.4	Results from User Questionnaires	116
5.7	Discussion	117
6	Conclusion and Future Work	119
6.1	Conclusion	119
6.2	Future Work	120
	Appendices	122
	A Image Dataset for Evaluation	122
	B Recruitment Screening Protocol for Evaluation of Interactive Image	

Segmentation Methods	124
C User Study Questionnaire	127
Curriculum Vita	146
References	135
Curriculum Vita	146

List of Figures

1.1	Main components of an interactive segmentation method [80].	8
1.2	(a)Contour; (b)Image; (c)Initial contour; (d)Final contour in parametric active models	14
1.3	(a)Level set; (b)Image and initial level set; (c)Evolving level set; (d)Final level set in geometric active models	16
1.4	(a)Graphical model; (b)Segmentation as graph cuts in graphical model	19
1.5	(a)Hard point constraints in active contours (white point); (b)Soft point constraints in active contours (blue points); (c)Scribbles of the object and background in graphical models (yellow scribbles for object and blue scribbles for background)	24

2.1 Synthetic example segmentation using AVM. (A) The model on the original image. (B) The binary map P_B estimated by intensity-based likelihood maps and applying the Bayesian Decision rule. (1) Initial model. (2) The model after 12 iterations. (3) Final result after 26 iterations without interactive polygons ($p = 0.74, q = 0.97$). (4) Final result with interactive polygons (merge polygon in yellow and split polygon in green) ($p = 0.99, q = 0.98$). (Note: all figures in the chapter are best viewed in color) 35

2.2 Synthetic example (a) The intensity overlap. (b) The intensity distribution of the foreground (green) and the background (red). 36

2.3 X-ray image segmentation using AVM. (A) The model on the original image. (B) The binary map P_B estimated by intensity-based likelihood maps and applying the Bayesian Decision rule. (1) Initial model. (2) The model after 23 iterations. (3) Final result after 84 iterations. . . . 37

2.4 Segmentation using AVM. (A)The X-ray image. (B) The MRI wrist image. (1) Segmentation result without interaction (X-ray: $p = 0.98, q = 0.98$; Wrist: $p = 0.97, q = 0.62$). (2) The initial contour and interactive polygons. (3) Final result (Spine: $p = 0.99, q = 0.99$; Wrist: $p = 0.98, q = 0.97$). 41

2.5 segmentation using AVM. (A)The X-ray foot image. (B) The binary ROI image. (1) Segmentation result without interaction ($p = 0.78, q = 0.95$). (2) The initial contour and interactive polygons. (3) Final result ($p = 0.97, q = 0.99$). 42

3.1	Synthetic image of a blob with heterogeneous intensity on a background of similar heterogeneous intensity. (a) Initial contour. (b) Unsuccessful result of region-based segmentation. (c) Successful result of edge-based segmentation technique [58]	45
3.2	Ball region is considered at each point along the contour, which is split by the contour into local interior and local exterior. (a) the local interior is the shaded part of the circle; (b)the shaded part of the circle is the local exterior [58]	48
3.3	(1)Initial contour for localized level sets; (2) Final segmentations.	55
3.4	(1)Initial contour for localized level sets; (2) Initial segmentations; (3) User scribbles for background (blue) or foreground (green); (4) New segmentation result.	56
4.1	Turbopixels	67
4.2	A simple graph (left) vs. a hypergraph (right)	68
4.3	Building a hierarchical hypergraph by grouping patches level by level.	69
4.4	(a) 4-connected grid graph; (b) building level-0 hypergraph	70
4.5	(a) Level-1 hypergraph; (b) Level-2 hypergraph;(c)Level-3 hypergraph.	71
4.6	basic belief propagation	73
4.7	(a)A figure of two leaves; (b)Original image with scribbles; (c)Segmentation result with the original initialization;(d)Segmentation result with the modified initialization.	77
4.8	(a)Connection dots (Green dot is the starting point and Blue dot is the ending point); (b)-(d) Front-propagation by fast marching; (e) Original Image and scribble; (f) The initial results; (g) Connection dots; (h) Final result.	81

4.9	(a)Construct hierarchical hypergraphs; (b)Label propagation on each level and between levels	82
4.10	(a)The original image with scribbles (b) The initial segmentation (c) Recommendation area is shown in the red window (d)Draw scribbles inside the recommendation window (e) The result in the second iteration	84
4.11	(a)The original image with scribbles (b) Segmentation result (c) The confusion map with recommendation scribble area	84
4.12	Convergences of (a) multilevel DBP comparing with standard DBP(b) multilevel GaBP comparing with standard GaBP	85
4.13	(1)Original image and scribbles; (2)Result of our method in DBP; (3) Result of our method in GaBP	86
4.14	Accuracy affected by random machine-generated scribbles in different length and different positions	87
4.15	Running time comparison between our framework, random walks, grow cut and graph cut	88
4.16	(1)Original image and scribbles; (2)Result of our method in DBP; (3) Result of our method in GaBP; (4)Result of random walks; (5)Result of grow cut	91
4.17	Error rates on 50 images in [89] for DBP	92
4.18	Error rates on 50 images in [89] for GaBP	92
4.19	Machine Experiments: comparing scribble recommendation system with nature upper bound and baseline	93
5.1	User interface of the interactive segmentation tool	105
5.2	Scribbles generated by random walk	109
5.3	Average accuracy in the first step for automatic evaluation and user study	111

5.4	Average final accuracy after 5 steps for automatic evaluation and user study	112
5.5	Number of iterations to reach 90% accuracy for automatic evaluation . .	113
5.6	Speed in each step for 5 iterations	114
5.7	Sensitivity analysis of segmentation by scribble perturbation.	115
5.8	Sensitivity analysis of segmentation by different quality of scribbles . . .	116

List of Tables

4.1	Region Growing Segmentation Methods	63
4.2	Graph Partitioning Segmentation Methods	63
4.3	Usage of User Scribbles in Interactive Segmentation Methods	64
4.4	Error rates for Grab Cut dataset	89
5.1	The number of iterations to reach satisfaction	113
5.2	Speed in one iteration in user study (sec)	114
A.1	Image dataset and its categories	123

Abstract

Interactive segmentation has been a classic approach to image segmentation. The user interactions used by most interactive segmentation methods, however, are not very intuitive.

In this thesis, we explore intuitive user interactions that can be incorporated into three popular segmentation methods so that these methods can achieve better results through user inputs. The three methods are parametric active contours, geometric active contours and graphical models:

- (1) We propose interactive polygons for Active Volume Models (AVM) which is an improved parametric active contour model. The two kinds of interaction polygons we apply are “merge polygons” and “split polygons” which identify the foreground and background, respectively. The proposed user interactions are intuitive and users can easily fix the segmentation results in the iterations by adding interactive polygons. The new interactive segmentation method can work very well on inhomogeneous medical images and the resulting segmentations have better sensitivity and specificity;
- (2) We propose a new interactive segmentation method which integrates user scribbles into the localized geometric active contour model or localized

level set, where scribbles are represented as a continuous energy term added into the level set functional. The resulting new interactive segmentation method can work very well on nature images.

(3) For graphical models, we propose a new interactive segmentation method which first models an image as a hierarchical hypergraph, then applies belief propagation to propagate the labels of seed pixels to the remaining unlabelled pixels. The resulting new interactive segmentation is more efficient and can achieve comparable accurate segmentation results for nature images. In addition, we propose a user interaction which allows users to change the connectivity of segmentation results. We also integrate a scribble recommend feature into the framework, which can intelligently recommend where the user should scribble next.

Finally, in order to evaluate interactive segmentation methods based on graphical models, we present a new framework that combines user study with user interaction simulated through machine-generated scribbles. We also propose evaluation measures to quantify the user experience and strategies for selecting where to put scribbles in the next iteration to mimic a real user. The proposed automatic evaluation framework is useful for preliminary evaluation of interactive segmentation methods which is confirmed with conducted user study. We also show the usefulness of the framework by evaluating several interactive segmentation methods including our proposed methods and the scribble recommendation feature.

This thesis is prepared under the supervision of Prof. Xiaolei Huang and Prof. Daniel Lopresti.

Principal Contributions

In this thesis, we explore user interactions that can be incorporated into three popular segmentation methods: parametric active contours, geometric active contours and graphical models. The corresponding interactive segmentation methods are augmented by easy, intuitive user inputs and shown to achieve better results.

The main contributions of this thesis are the following:

- With respect to interactive segmentation algorithms, the three interactive segmentation methods we propose for parametric active contours, geometric active contours and graphical models, respectively, are given below:
 1. new interactions called interactive polygons that are integrated into Active Volume Models (AVM). We propose two kinds of interactive polygons: “merge polygons” and “split polygons”, to identify the foreground and background, respectively. Users can easily correct segmentation results by iteratively adding these interactive polygons.
 2. a new interactive segmentation method which integrates user scrib-

bles into the localized geometric active contour model, where users can use scribbles to identify foreground and background pixels and correct erroneous initial segmentation results.

3. a new interactive segmentation method based on graphical models, which first models an image as a hierarchical hypergraph, then applies belief propagation to propagate the labels of seed pixels to the remaining unlabeled pixels. Two belief propagation algorithms are implemented: discrete belief propagation and Gaussian belief propagation in which the label variables are modeled as discrete and Gaussian variables respectively. In addition, a new user interaction is proposed which allows users to change the connectivity of segmentation results by adding points to identify thin line structures. A new feature is also integrated into the method to intelligently recommend where the user should add scribbles in the following iterations.
- With respect to evaluation of interactive segmentation methods, we propose a new framework to evaluate interactive segmentation methods based on graphical models. The framework combines user study with automatic evaluation by simulating user interactions with machine-generated scribbles. In addition, new evaluation measures are proposed to quantify the user experience; methods are proposed to generate scribbles by random walks; strategies are described for selecting where to put scribbles in the next iteration to mimic a real user; and user studies are conducted to validate the proposed automatic evaluation. The proposed evaluation framework is used to evaluate several

interactive segmentation methods including our proposed method and the proposed scribble recommendation feature.

- With respect to applications, the interactive segmentation methods proposed are useful for both medical image segmentation and nature image segmentation. The proposed user interactions are intuitive and their ease of use is validated by user studies. We believe interactive image segmentation holds great potential in robust, accurate and reproducible image segmentation, and the proposed new methods and evaluation mechanisms contribute toward better segmentation through user interactions.

Chapter 1

Introduction

Image segmentation is the process of identifying and separating relevant objects and structures in an image. Typically, it is defined as the partitioning of an image into non-overlapping, constituent and meaningful regions which are homogeneous with respect to some image characteristics, such as intensity value or texture feature.

Mathematically, given the image domain I , the segmentation problem is to calculate the sets $S_k \subset I$ whose union is the entire image I . Thus, the sets must satisfy

$$I = \bigcup_{k=1}^K S_k, \quad (1.1)$$

where $S_i \cap S_j = \emptyset$ for $i \neq j$ [84].

Image segmentation is a fundamental problem whose solution is essential to solving many vision problems where the quality of segmentation output largely influences the performance of the whole vision system. It remains an important research topic, and there have been many image segmentation algorithms which have achieved success in specific applications, such as object tracking [51, 57], object recognition [90, 115], image

reconstruction [16, 36].

Image segmentation methods have been applied to both 2D or 3D problems. In this thesis, we focus on 2D image segmentation methods where $I \in \mathbb{R}^2$. Also, we apply these segmentation on medical and nature images. We refer medical images as gray scale images while nature images as color or texture images. We only consider the color feature vector of nature images: $v = (r, g, b)$, where each component is an RGB color component in the range 0-255.

All these image segmentation methods largely fall into three categories: supervised, automatic/unsupervised and semi-supervised/interactive algorithms.

Supervised segmentation methods are pattern recognition techniques that partition the image into meaningful regions using training data with known labels. The term *supervised* refers to the fact that these methods require manually labeled training data, and they learn a mapping function from features to labels in order to classify or segment new data. We provide some examples of supervised classifiers here. The K-nearest-neighbor (kNN) is a simple nonparametric classifier since there is no underlying assumption about the statistical structure of the data. Maximum Likelihood (ML) is a widely-used parametric classifier, which assumes that the image features are sampled from probability distributions, such as a Mixture of Gaussian. Support Vector Machine (SVM) [29] is also a commonly-used supervised learning method that considers two-group classification problems. Artificial Neural Networks (ANNs) are a network of weighted nodes where the weights are iteratively adjusted until the network outputs are sufficiently close to desired values of training data [82, 86]. Besides these methods, Boosting [93] algorithm is a general method for improving the accuracy of any given learning algorithm. The advantage of supervised classifiers is that the integration of high-level prior knowledge can significantly improve segmentation accuracy. However,

the disadvantage is also apparent, since manual labeling of training data can be tedious and time-consuming. In some situations, manually labeled training data are not available.

Fully automatic, unsupervised segmentation of arbitrary images remains an unsolved problem. This is especially true for nature images in which there are multiple objects. Part of the reason lies in the definition of the segmentation problem as the task of identifying relevant objects in an image. But the notion of a relevant object is highly dependent on context and generally impossible to define based on the image data alone. The identification of relevant objects may require, for example, user experience, knowledge of the task at hand, and knowledge of the imaging process. These are qualities that humans possess, but that computers are notoriously lacking. To tackle this challenge, semi-supervised, or interactive segmentation methods employ human expert knowledge as additional input, thus making the segmentation problem more tractable. For example, in nature images, users can interactively identify the object of interest among many objects.

The goal of interactive segmentation methods is to minimize the required user interactions, while maintaining tight user control to guarantee the correctness of the results. These interactive methods vary in the models and the user interactions used.

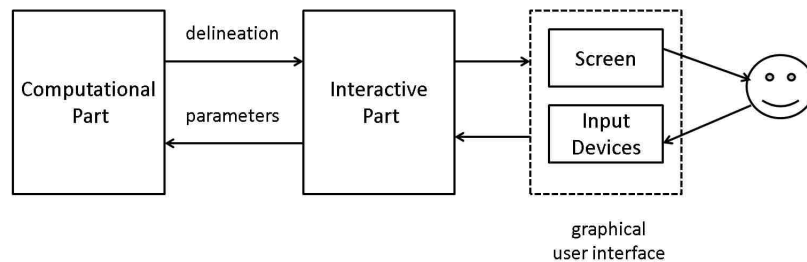


Figure 1.1: Main components of an interactive segmentation method [80].

The interactive segmentation process [80] is illustrated in Figure 1.1. It shows

1.1 Desired Properties of Interactive Segmentation Methods

three components of an interactive image segmentation: the computation part, the interaction part and the graphical user interface (GUI). The GUI is the component that receives the user's guidance for action, often via visual programming components such as brush for scribbles, sliders for parameter setting, or direct image clicks for pictorial input. The interaction part translates the input given via the GUI into parameters so that they can be used by the computation part. The computation part corresponds to some algorithm that uses a specific model to encode the information available in an image with given parameters, which are provided by the interaction part. Then it performs computation to generate segmentation results.

In the remainder of this chapter, we will discuss the desired properties of interactive segmentation methods followed by reviewing interactive segmentation methods based on different models. We then will review user interactions in these segmentation methods. Later, we will give an overview of our proposed three new interactive segmentation methods proposed which incorporate user interactions into different models. Finally, we will give an introduction to our proposed evaluation system which combines user study and automatic evaluation with machine-generated scribbles.

1.1 Desired Properties of Interactive Segmentation Methods

An interactive segmentation method takes an image, together with user input given in some form, and produces a segmentation of the image. For a good segmentation method, Grady [46] proposed the following properties:

1. Fast computation.
2. Fast editing.

1.1 Desired Properties of Interactive Segmentation Methods

3. An ability to produce, with sufficient interaction, an arbitrary segmentation.
4. Intuitive segmentation.

Grady did not specify the meanings of these properties. According to our understanding, the first two requirements are related to the speed of the computation part of the segmentation process. As shown in Figure 1.1, interactive segmentation is an iterative process. During this process, the segmentation results should be updated quickly when the user changes the inputs in an ideal situation. From one iteration to the next iteration, the changes in user inputs are relatively small, so it is possible to accelerate the computation of the current solution for the current input by reusing information from the previous solution. Fast editing can be achieved in this way.

The third requirement is related to user control in iterations. For a good interactive segmentation method, only modest user interactions are required to achieve a desired result. During the process, the user can apply interactions iteratively until a desired segmentation has been obtained. But still there will be cases when the segmentation method fails to obtain a desired result. So it is essential for a good interactive segmentation method that the user can correct the segmentation results and achieve an arbitrary segmentation.

In interactive segmentation, users can judge the correctness of the result and terminate the iterative process. Thus, the goal of a segmentation method is primarily to produce segmentations that capture the intent of the user instead of correct segmentation. This distinction is emphasized by the fourth requirement. This requirement is rather vague, thus hard to quantify. In most cases, people assume that the boundary of the desired segmentation should coincide with regions of high contrast, e.g., strong edges, in the image. For nature images, this usually not the case since the desired segmentation may contain several such regions. This property is also related to robustness

of the interactive segmentation method. The segmentation method should perform consistently and predictably on degraded images, e.g., images with noisy or missing data. Also, since interactive segmentation methods are subject to variations in user inputs, it is desirable for a segmentation to be robust with respect to small changes in user inputs in order to make segmentation results repeatable.

As we can see, these properties are vague and difficult to quantify and measure. We will put our efforts in Chapter 5 to explore quantitative measures of these properties and propose a new framework for evaluation of interactive segmentation methods.

1.2 Review of Interactive Segmentation Methods

In this section, we review interactive segmentation methods based on two different mathematical models: active contour models and graphical models.

1.2.1 Active Contour Models

Active contours are curves that deform within digital images to recover object boundaries. Typically, to extract an object boundary, an initialization contour needs to be placed on the image and allowed to deform, over multiple iterations, under the influence of internal and external forces. Internal forces keep the curve or surface smooth throughout the deformation. External forces are usually derived from image information and drive the curve or surface towards the desired object boundaries. Active contours are classified as either parametric active contours or geometric active contours, according to their representation and implementation.

Parametric Active Contour Models or Snakes

Parametric active contours are represented explicitly as parametrized curves in a Lagrangian formulation. Parametric active contours are also called “snakes”. In this thesis, we will use snakes and parametric active contours interchangeably.

All snake properties and behaviors are specified through an energy function. A partial differential equation controlling the snake causes it to deform while reducing its energy.

The first snake model was proposed by Kass [55] in 1988. The energy function which the snake was to minimize was defined as the following:

$$\begin{aligned}
 E_{snake} &= \int_0^1 e_{int}(\mathbf{v}(s))ds + \int_0^1 e_{ext}(\mathbf{v}(s))ds \\
 &= \alpha \int_0^1 |\mathbf{v}_s(s)|^2 ds + \beta \int_0^1 (|\mathbf{v}_{ss}(s)|^2) ds + \lambda \int_0^1 P(C(s))ds \quad (1.2)
 \end{aligned}$$

where the position of the snake on the image is represented parametrically by a planar curve $\mathbf{v}(s) = (x(s), y(s))$. The term $e_{int} = \alpha(|\mathbf{v}_s(s)|^2 + \beta|\mathbf{v}_{ss}(s)|^2)$ represents the internal energy of the curve due to bending, and controls the smoothness of the contours. The external energy $e_{ext} = \lambda \int_0^1 P(C(s))ds$ represents the image forces pushing the snake toward the desired object. $P(C(s))$ is a potential which is based on some image features, such as intensity, gradient and edges. α , β and λ are real positive constants to weight the smoothness constraints and image forces.

For the internal energy, $\mathbf{v}_s(s)$ is the first derivative and $\mathbf{v}_{ss}(s)$ is the second derivative of $\mathbf{v}(s)$ with respect to s . $\mathbf{v}_s(s)$ specifies the rate of change of length of the curve; the coefficient α allows the curve to have a smaller or larger degree of contraction. Therefore α can be interpreted as the elasticity coefficient. $\mathbf{v}_{ss}(s)$ specifies the rate of the curvature; the coefficient β regulates the rate of the change of the curve in the

direction normal to its boundary. This term is used to preserve the smoothness. If the value of β is high then the curve is hard and resists bending; when it is small, it allows the curve to develop a corner. These two coefficients can be adjusted so that the curve has the appropriate elasticity and can deform to the boundary of the object of interest.

For the external energy, the commonly used image energy for grey-scale images is

$$e_{ext} = -|\nabla I(x; y)|^2 \quad (1.3)$$

where the term $I(x; y)$ represents the image intensity function, which may be considered a “landscape”. The gradient operator calculates the edge map over the image. Low values of the negative edge map can be interpreted as a valley, the snake will go in the direction of the minima on the edge map in a manner resembles like rolling down to the valley, driven iteratively by the image force. For snakes, the initial curve must be interactively specified by the user; then the energy terms can deform the snake to the desired boundary of the object.

As noted above, snakes using edge terms rely on image gradient information to deform, so they are sensitive to noise and spurious edges. For this reason, they are subject to converging to stop at undesirable local minima in their corresponding energy functions. It has been shown that this model may be improved to more accurately estimate a variety of complex object boundaries. One improvement is to include region information. Because region-based approaches incorporate statistical information about larger areas of the image (as compared to edges only), they are less constrained by the initial position, and less sensitive to noise. For example, the Active Volume Model (AVM) uses region statistics information from both the interior and exterior of the contour. Details of AVM are discussed in Chapter 2. Figure 1.2 illustrates the parametric active models and (b) and (c) shows the initial contour and the converged

contour responding to the boundary of the object, respectively.

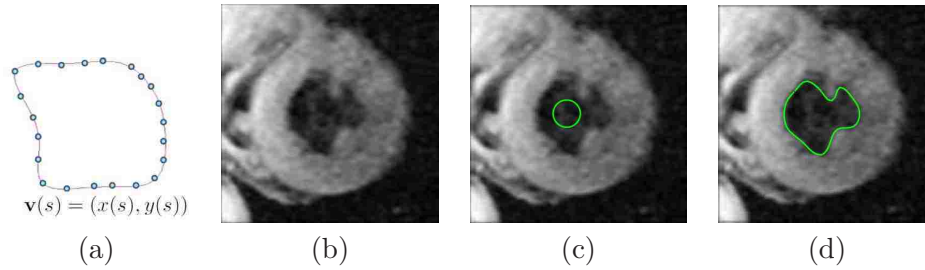


Figure 1.2: (a)Contour; (b)Image; (c)Initial contour; (d)Final contour in parametric active models

Geometric Active Contour Model or Level Set

Geometric active contours are represented implicitly as level sets of a scalar function of high-dimensional variables [30, 63]. The level set approach was first introduced by Osher and Sethian [81] in fluid dynamics. Applying it to image segmentation was simultaneously suggested by Casseles et al [22] and Malladi and Sethian [64].

The basic idea is to embed the planar curve in a scalar function $\phi(x, y, t)$ (Figure 1.3(a)), which defines a function of spatial variable (x, y) , evolving in time t , such that the contour is retrieved by intersecting the surface defined by ϕ with the xy plane. This contour is called the zero-level-set. The signed distance between the plane and the surface is defined over the rest of the xy plane, where the sign is positive for points outside the curve and negative for points inside. Instead of evolving the curve in the plane like snakes, the surface evolves in time with respect to the xy plane, thus the new contour (the zero-level-set of the function) can be obtained. For an image, $\phi(x, y, t)$ can also be seen as a distance map where pixels that lie on the object boundary will get 0 values, while all other pixels are assigned their signed distance from the curve.

Since the evolving front $\mathbf{C}(t)$ is a zero level set of the scalar function for every time

t , we obtain for the evolving contour:

$$\phi(\mathbf{C}(t), t) = 0 \quad (1.4)$$

Derivation with respect to t , using the chain rule we get

$$\phi_t + \nabla\phi \cdot \mathbf{C}_t = 0 \quad (1.5)$$

Let F be the speed in which the contour propagates in the direction normal to the curve. Hence

$$F = \mathbf{C}_t \cdot \mathbf{n} \quad (1.6)$$

Where

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad (1.7)$$

Therefore Equation 1.5 becomes:

$$\phi_t + F|\nabla\phi| = 0 \quad (1.8)$$

And we get a PDE on ϕ with an initial contour $\phi((C), t = 0)$. This equation can be solved using finite differences approximations for the spatial and temporal derivatives. Figure 1.3(b), (c) and (d) show the initial contour, the evolving contour and final contour in level set, respectively.

Just as for snakes, where we have to select the different energy terms for smoothness and convergence, for level sets we need to choose the propagation speed terms defining F . These should incorporate an internal regularization term and an external term to encourage stopping at edges.

Malladi and Sethian [65] suggested a regularization term which is a curvature de-

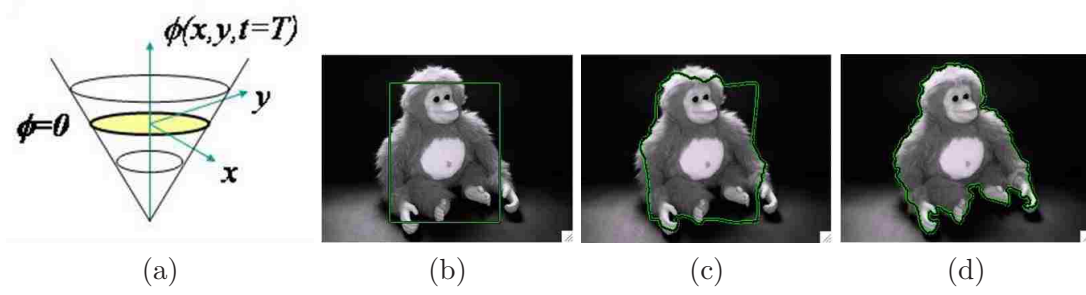


Figure 1.3: (a)Level set; (b)Image and initial level set; (c)Evolving level set; (d)Final level set in geometric active models

pendent speed $F = F(k)$, for example, $F = -k$, where concave points will move faster in the normal direction. In order to get an inflation force we can add a constant term F_0 :

$$F(k) = F_0 - k \quad (1.9)$$

Malladi and Sethian multiply the above speed by a function which serves as an edge-stop term:

$$g(x, y) = \frac{1}{1 + |\nabla G_\sigma * I(x, y)|} \quad (1.10)$$

This term behaves as an inverse of the image gradient so that it approaches zero near an edge and drives the velocity to a stop. To skip weak edges, we can use a Gaussian filter G for smoothing. Putting it all together in Equation 1.13, we obtain the following:

$$\phi_t + g(x, y)(F_0 - k)|\nabla\phi| = 0 \quad (1.11)$$

Now incorporating the classical equation for the curvature k :

$$k = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = \frac{\phi_{yy}\phi_x^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{xx}\phi_y^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \quad (1.12)$$

and changing the sign F_0 (can be positive or negative), the final level set flow is obtained as a PDE:

$$\phi_t = g(x, y)|\nabla\phi|(F_0 + \operatorname{div}(\frac{\nabla\phi}{|\nabla\phi|})) \quad (1.13)$$

In order to obtain faster convergence, the *Geodesic Active Contour* model (GAC) [23] was proposed; GAC combines explicit active contour models with implicit active contour models, whose energy function of the level set is

$$E(C(s)) = \int_0^1 g(|\nabla I(C(s))|)|C'(s)|ds, \text{ where } g(|\nabla I|) = \frac{1}{1 + |\nabla I|^2}, \quad (1.14)$$

Then the level set flow as a PDE is defined by:

$$\phi_t = g(x, y)|\nabla\phi|(F_0 + \operatorname{div}(\frac{\nabla\phi}{|\nabla\phi|})) + \nabla g \nabla\phi \quad (1.15)$$

where C represents the front (zero level set) curve of the evolving level set function. This model is different from Malladi and Sethian's, since it has the additional term $\nabla g \nabla\phi$. This term attracts the curve toward the boundaries of the object better than the $g(x, y)$ term in Equation 1.13. Moreover, in cases of high variation in gradient values, the Malladi and Sethian level set approach may fail to stop, since $g(x, y)$ reduces the energy to zero only for high gradient values. Low values, or gaps in the edge may create difficulties in stopping the curve. The new GAC term allows tracking of boundaries with high variation of gradients, including small gaps.

Just as for snakes, we can integrate region information into the level set formulation. A well-known example is the Mumford-Shah functional [76]. The segmentation problem, as formulated by Mumford and Shah, can be defined as follows: given an observed image u_0 , find a decomposition Ω_i of Ω , where $\Omega \subset R^2$, such that the new

“segmented” image u varies smoothly within each Ω_i , and discontinuously across the boundaries of Ω_i . The simplified case is obtained by restricting the segmented image u to be piecewise-constant (i.e. $u = \text{constant } c_i$ inside each component Ω_i) or piecewise-smooth functions. The problem is often called the “minimal partition problem”. Recently, solutions for several simplified cases of the Mumford-Shah functional have been proposed in the level set framework. In [24], the piecewise-constant function is minimized within the level set framework by Chan and Vese:

$$E(C) = \lambda_1 \int \int_{R_i(C)} |I(p) - c_1| dp + \lambda_2 \int \int_{R_o(C)} |I(p) - c_2| dp + \mu \text{Length}(C) + \nu \text{Area}(R_i(C)), \quad (1.16)$$

where λ_1 , λ_2 , μ and ν are positive constants to balance the contribution of each term, R_i and R_o correspond to the interior foreground and exterior background regions of the contour C , c_1 and c_2 represent the mean intensities of interior and exterior regions respectively, and $I(p)$ refers to the image intensity value at pixel p .

An alternative approach to including region information is [114], where segmented regions are characterized by statistics such as the mean intensity and variance. The approach given in [92] applies a multiphase level set to segmentation, assuming piecewise-constant intensity within one region. It is considered as solving a classification problem because it assumes the mean intensities of all region classes are known *a priori*, and only the set of boundaries between regions is unknown.

All of the above approaches assume the distributions within regions to be piecewise-constant, piecewise-smooth, Gaussian, or Mixture-of-Gaussian, which may limit their effectiveness in segmenting objects whose interiors have textured appearance and/or complex multi-modal intensity/color distributions. This is often the case in nature images. Furthermore, only integrating region information in level sets is not sufficient for

many nature images. In Chapter 3, we will introduce an improved level set formulation which only consider local region, rather than global region, color image statistics.

1.2.2 Graphical Models

Graphical segmentation methods model an image as a weighted, undirected graph (Figure 1.4(a)). Mathematically, let $G = (V, E)$ be a graph where $v_i \in V$ is a set of vertices corresponding to the image elements, pixels or regions in Euclidean space. E is a set of edges connecting certain pairs of neighboring vertices. Each edge $(v_i, v_j) \in E$ has a corresponding weight $w((v_i, v_j))$ which measures some property between the two vertices connected by that edge.

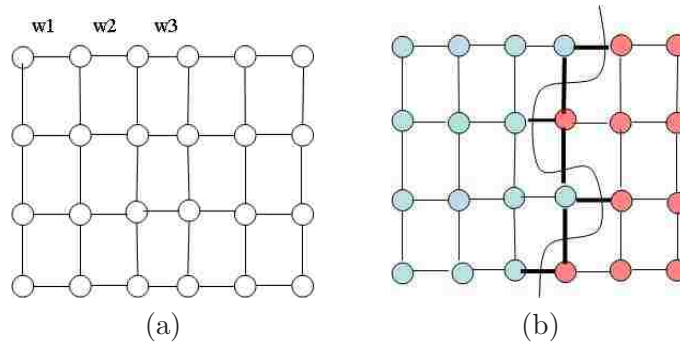


Figure 1.4: (a)Graphical model; (b)Segmentation as graph cuts in graphical model

In this thesis, we only consider bi-segmentation problems. For interactive image segmentation based on graphical models, the initialization is usually user-defined scribbles indicating some labeled pixels belonging to the object (foreground) or background. Then an image is bi-partitioned into two disjoint regions, such that each region A is a connected graph $G = (V, E)$, where $V' \subseteq V, E' \subseteq E$.

Like bi-partition of a graph, the segmentation of an image can be interpreted in the form of graph cuts (Figure 1.4(b)), and the cut value is usually defined as:

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v), \quad (1.17)$$

The exact solution to image segmentation is hard to achieve because of uncertainties of the understanding of image content. Therefore, it is more appropriate to solve this problem with optimization methods, where the segmentation problem is formulated as a minimization of some cost functions. We can find an exact or approximate solution for these optimization methods. In this case, the optimal bi-partitioning of a graph can be taken as the one which minimizes the cut value in Equation 1.17.

In graphical models, image segmentation is also formulated as a labeling problem, where a set of labels L is assigned to a set of elements in V . In binary segmentation, for example, the problem can be described as assigning a label f_i from the set $L = (\text{object}, \text{background})$ to elements $i \in V$ where the elements in V are the image pixels or regions. Labeling can achieve the same effect on image partitioning. Many methods can perform both segmentation and labeling simultaneously.

According to [4], graphical model based interactive segmentation methods are classified into five categories by their methods:

1. Minimal spanning tree based methods [38, 116]: segmentation methods based on the clustering of image pixels on the minimal spanning tree. The regions or sub-graphs can be obtained by removing edges from the whole graph so that the connection of vertices inside regions satisfies the minimal sum on edge weights.
2. Graph cut with cost functions [97, 112]: segmentation methods based on optimizing the cost function by graph cut. The cost function can be different with different cut criteria.
3. Graph cut on Markov random field models [49]: segmentation methods based

on optimization of the graph cut function in the MAP-MRF framework via mincut/max-flow algorithms or other alternatives. The graph cut function is combined with user scribbles and the regularization of the smoothness.

4. The shortest path based methods [34]: the boundary of the object is a set of shortest path between vertices.
5. Other methods: several efficient graph theoretic methods such as random walks [46] and dominant set based method [83] that do not belong to the above categories.

A comprehensive survey of graphical models can be found in [4].

1.2.3 Discussions

The main difference between parametric and geometric active contour models is that geometric active contour models do not need explicit parametrization; geometric active contour models can handle topology changes (split and merges) very naturally, and they are commonly used in segmenting multiple objects [35]. Due to their internal forces, parametric active contour models can not deal with protrusions and sharp/thin structures, while the level set approach may be able to satisfactorily segment such structures, for example, tubular structure in [100]. A detailed discussion on the relationship between the two approaches can be found in [113]. Both parametric and geometric active contour models need an initialization contour to bootstrap the evolution of the contours. The biggest drawback of active contours, however, is that the common formulations do not incorporate much user interaction in the process. Thus, it is difficult for users to correct segmentation results during iterations. Graphical models tend to be more flexible with respect to user interactions, where user scribbles can naturally

be integrated to improve results during iterations.

1.3 User interactions in Interactive Segmentation Methods

Two main types of inputs provided by the user during the interactive segmentation methods were identified in the reviewed methods: setting parameter values and pictorial input directly on the image grid.

1.3.1 Setting or Changing Parameter Values

There are two types of parameters: real and discrete. In the case of real parameter values $v \in [v_1, v_2]$, $[v_1, v_2] \subset \mathbb{R}$, v may take on continuous values from the domain $[v_1, v_2]$. Some examples of such parameters are: the threshold level for binarization [21]; the balance of weights in the cost function of a deformable model [20]; the scale used to compute image derivatives and locate image structure [62], and the parameters used to calculate feature similarity between pixels in graphical models such as Random walks [46];

In the case of a discrete domain, values $v \in [v_1, v_2] \subset \mathbb{Z}$ refer to parameters that specify some levels for the computation method. Some examples of such parameters are number of iterations [21] and the maximum size of the segmented region given by the number of pixels in region merging and splitting methods [99].

Parameter setting or changing can be done with inputs via a text field, a slider, a dial or a similar interactive technique, and the result updated with the new parameter configuration is then displayed on the screen for user evaluation.

1.3.2 Pictorial Inputs on the Image Grid

Pictorial input refers to user input provided directly on images, which are positions in the image grid $[x, y] \in R$ such as dots, lines or regions. They are spatial parameters for the interactive segmentation methods. There are three purposes of these spatial parameters: initial contours, constraint points for active contours, and scribbles for graphical models.

Initial contours in Active Contours

In some interactive segmentation methods such as active contour models [24, 55], pictorial inputs on the image grid identify initial segmentation of the object of interest. They specify a rough boundary of the segmentation which is used to bootstrap the computation part of interactive segmentation methods. Users can draw the initial contour freely on the image and the contour then deforms to the outline of the object. In these models, the contour is optimized on the basis of an energy function that balances the properties of contours and image information. Some methods also use this initial contour to learn the region information which can be incorporated into active contour models [95].

Constraint points in Active Contours

Constraint points are user-specified points commonly used in active contour models [41][71]. They serve as geometric point constraints. Hard points shown in Figure 1.5(a) are the points through which the segmentation contour is forced to pass. Soft points shown in Figure 1.5(b) are attraction points that pull or push the segmentation contours, which are incorporated as external forces into the model. Soft points are often used to pull the segmentation contour off spurious edges.

Constraint points are points provided directly on the image, using the mouse or some other pointing device.

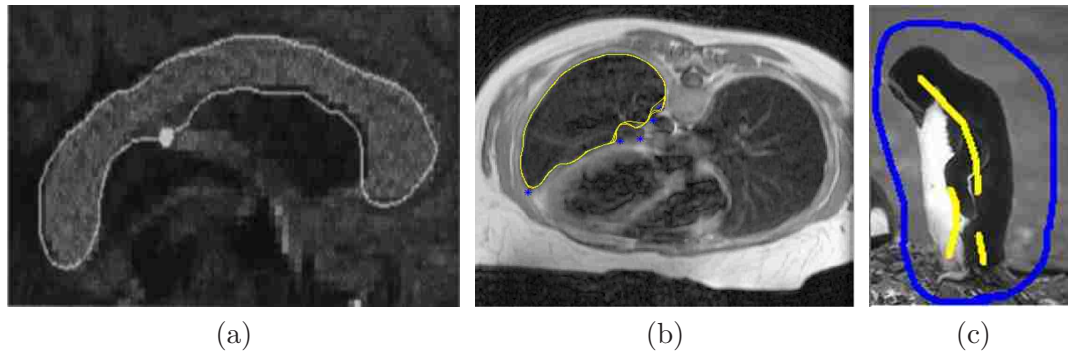


Figure 1.5: (a)Hard point constraints in active contours (white point); (b)Soft point constraints in active contours (blue points); (c)Scribbles of the object and background in graphical models (yellow scribbles for object and blue scribbles for background)

User Scribbles in Graphical Models

In graphical models, users can draw separated or continuous lines or dots on images to provide user inputs to the models. As shown in Figure 1.5(c), these scribbles are partial labeling for image pixels which identify the object and the background. They serve various functions for graphical models such as: (1) Identify Regions of Interest (ROI); (2) Provide seeds with pre-defined labels; (3) Control topology; (4) Correct segmentations. We will give a more detailed review in Chapter 4.

1.3.3 Discussion

Setting or changing parameters is simple to implement, but it requires user knowledge in how the parameter changes will affect the computation; otherwise it may lead to inefficient interaction or even failure to achieve desired results. This drawback may be somewhat alleviated by careful user training.

1.4 Proposed Interactive Segmentation Methods based on Different Models

Pictorial inputs are simpler and more intuitive for users. Initial curves for active contours, however, may be time consuming to draw, since some active contour models require fairly precise initial boundaries close to the edges. Although it is convenient to use constraint points, again, this interaction could be very tedious; for example, users may need to add many attraction points to make the segmentation curve deform to the correct edges. Also, adding hard points or moving the control points could make the segmentation contours no longer smooth. It is reasonable to hypothesize that scribbles may be the most intuitive, efficient and flexible interactions and a few mouse clicks may suffice. It is very natural to include user scribbles in graphical models and make the whole process iterative. The meaning of scribbles is clear to users since they only identify some background pixels and some foreground pixels. Users do not need extra knowledge or training in using interactions to complete the segmentation tasks. In this thesis, we only explore pictorial inputs as user interaction and incorporate these into the algorithms.

1.4 Proposed Interactive Segmentation Methods based on Different Models

1.4.1 Motivation

As mentioned above, one drawback of parametric and geometric active contour models is that the common formulations do not allow much user interactions during iteration. Upon initialization, boundaries evolve to the final segmentation without the user being able to guide or correct the segmentation. As we have mentioned, for nature images, users' guidance and knowledge is necessary for interactive segmentation methods to achieve desirable results. For active contours, we need to develop intuitive user inter-

1.4 Proposed Interactive Segmentation Methods based on Different Models

actions and design new active contour formulations where these user interaction can be incorporated.

For graphical models, user scribbles are easy to integrate into the algorithms, but we have found that more flexible interactions are required. For example, it is difficult for users to draw a very thin line to identify that it belongs to the foreground. It is also helpful to guide users to add more scribbles in the proper places during iteration to obtain better results.

For evaluation of interactive segmentation methods, user study is commonly used but it is time consuming and labor intensive. Automation of evaluation may be employed, however, as a preliminary evaluation step. At the same time, we need to quantify the user experience to create an evaluation measure.

Towards these goals, we propose three new interactive segmentation methods are proposed in this thesis based on parametric active contours, geometric active contours and graphical models.

As for evaluation, the first two active contour methods use contours as the initialization to boost the deformation while in the following iterations, the user interactions to correct segmentation are different. In our case, interactive polygons and scribbles are used instead for parametric and geometric active contours, respectively. For graphical models, scribbles are used both for initialization and correcting segmentation results in iterations. So evaluation of graphical model based interactive segmentation are much easier. In this thesis, we design a new framework for evaluation of interactive segmentation based on graphical models. Evaluation of active contour models can adopt the similar framework and the only difference is evaluation of initial use interaction and user interaction to correct segmentation results have to be evaluated differently.

1.4 Proposed Interactive Segmentation Methods based on Different Models

1.4.2 Interactive Active Volume Models with Interactive Polygons

We propose a new user interaction method called interactive polygons. These interactive polygons are designed for use with an improved parametric active contour model called Active Volume Model(AVM), which deforms with constraints from both Region Of Interest (ROI) and image gradient information. The two kinds of interactive polygons we apply are merge polygons and split polygons which identify the foreground and background, respectively. Users are allowed to draw these polygons to correct the original AVM segmentation results. These interactive polygons are used to update the region statistics in the original model and help the model deform to the desirable boundaries.

1.4.3 Interactive Localized Level Set

Like geometric active contour models, the current formulation of level sets does not allow much user interaction in the process except initialization of contours. To segment nature images, we modify a level set formulation to include color and texture image information. The new modified level set allows any region-based segmentation energy to be re-formulated in a local way. We consider local rather than global image statistics and evolve a contour based on local information. Thus it can deal with complex, colorful and challenging nature images. Based on this formulation, we further propose a probabilistic formulation for color image segmentation which integrates user scribbles. The user scribbles are represented in a continuous energy term, which is added into the new level set formulation. So the resulting algorithm can determine the most likely segmentation given the input image and the user input.

1.4 Proposed Interactive Segmentation Methods based on Different Models

1.4.4 Interactive Segmentation through Label Propagations on Hierarchical Hypergraphs

We also develop a new framework for interactive segmentation using graph-based models. In this framework, an image is modelled as a hypergraph with pixels as the nodes of the graph. Then belief propagation is used to propagate the labels of seed pixels to the remaining unlabeled pixels. We implemented two belief propagation algorithms: discrete belief propagation, and Gaussian belief propagation, in which the label variables are modeled as discrete and Gaussian variables, respectively. To make the algorithm more efficient, a hierarchical scheme for label propagation is also integrated in the algorithm. Experiments show that our new algorithms can achieve reasonable results in an efficient way. Based on our new framework, we propose new, intuitive user interactions which allow users to change the connectivity of segmentation results. Another contribution is that we also integrate a feature into the framework that intelligently recommends where the user should scribble next.

1.4.5 Evaluation of Interactive Segmentation Methods based on Graphical Models

In order to evaluate graphical model based interactive segmentation methods, we propose a new framework which combines user study with automatic evaluation by simulating user interactions with machine-generated scribbles. We propose evaluation measures to quantify the user experience. We generate scribbles by random walks in arbitrary length and shapes according to the ground truth. We develop strategies for selecting where to put scribbles in the next iteration to mimic a real user. Also we conduct a user study to validate the proposed automatic evaluation. Then we use the proposed framework to evaluate several interactive segmentation methods including our

proposed method and the proposed recommending scribble recommendation feature.

1.5 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, we introduce a new interactive segmentation method which combines new user interactions called interactive polygons with an improved parametric active contour model, i.e., AVM. In Chapter 3, we propose a new interactive segmentation method which integrates user scribbles into the localized geometric active contour model. In Chapter 4, we design a new interactive segmentation method based on graphical models which can effectively achieve good results by user scribbles and has new interactions to change topology of the segmentation result. In Chapter 5, we present a new evaluation framework for interactive segmentation methods which combines user study with automatic evaluation by simulating user interaction with machine-generated scribbles. We conclude the thesis and discuss future work in Chapter 6.

Chapter 2

Interactive Polygons in Region-based Parametric Active Contour

2.1 Introduction

Boundary extraction is an important task in medical image analysis. The main challenge is to retrieve high-level information from low-level image signals while minimizing the effect of noise, intensity inhomogeneity, and other factors. Model-based methods have been widely used with considerable success. We have proposed a novel deformation-based method, called Active Volume Model (AVM)[96] which deforms with constraints from both Region Of Interest (ROI) and image gradient information. The ROI, which represents the predicted object, is obtained from a classification of image features based on model-interior statistics. An approximation of the object appearance statistics, the model-interior statistics are learned adaptively during model evolution.

An advantage of the AVM model is that its formulation allows the ROI information to naturally become part of the snake's external forces; in this way, rapid model deformations can be derived by finding the solution of the Euler equations in a variational framework [54]. Experiments show that the AVM model performs better for most medical images than other gradient-based active contour models with respect to robustness and accuracy.

For some medical images, however, for example those with complex objects in cluttered backgrounds, the AVM model may fail due to similar appearance between the foreground and some background objects. In these cases, user interaction can help. For snake-like deformable models, the traditional user interactive controls are so-called soft points and hard points [40]. Soft points are attraction points which can incorporate the external forces into the model. Hard points are the points through which the segmentation contour is forced to pass. Some deformable models [72] even allow users to pull the controllable points to the correct positions. Although these interaction methods are convenient, the interaction could be very tedious, for example, users may need to add many attraction points to make the segmentation curve deform to the right edges. Adding hard points or moving the control points could make the segmentation contours no longer smooth.

As for user interaction in our AVM method, we can take advantage of the fact that AVM integrates the model-interior statistics and deforms with constraints from ROI. In this chapter, we explore a new user interactive method which is similar to the scribbles used in other interactive segmentation methods such as the random walker method [47] and many MRF based methods [13]. Users can draw polygons in the background or the foreground in addition to the original deforming contour. These polygons are called “merge polygons” if they are drawn in the foreground, and “split polygons” if

drawn in the background. At the same time, we can also call “merge polygons” as “foreground polygons” and “merge polygons” as “foreground polygons”. From these polygons, we calculate statistical information for ROI region or background region. Then we incorporate this information into the original deformable model. Thus, the performance of the deformable model can be improved. The process can be iterated until the desired contour is obtained.

We demonstrate that our interactive polygons can be very helpful in segmenting inhomogeneous images. Although there are some algorithms [60] which can automatically segment such images, these algorithms still have parameters which must be manually adjusted to get good results. These parameters are not very intuitive to users who are unfamiliar with segmentation methods, while our interactive polygons can be easily understood and manipulated until the desired segmentations are achieved.

2.2 Active Volume Model

An active volume model is a deforming solid that minimizes internal and external energy. The internal constraint ensures the model has a smooth boundary. The external constraints come from image data, priors, and/or user-defined features. Representing the model boundary parametrically, $\mathbf{v}(s) = (x(s), y(s))$, the internal energy term of AVM is defined similarly to active contour models.

$$E_{int} = \int_0^1 (\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2)ds \quad (2.1)$$

The external energy function consists of two terms: the gradient term E_g and the

region term E_R . So the overall energy function is:

$$E = E_{int} + E_{ext} = E_{int} + k \cdot (E_g + k_{ext} \cdot E_R) \quad (2.2)$$

where k is a constant that balances the internal and external forces. k_{ext} is a constant that balances the contributions of the gradient term and the region term.

The gradient data term can be defined using the gradient map, edge distance map, or a combination of both. The region data term is a novel aspect of the active volume model, in which it learns the appearance statistics of the object of interest dynamically and the model's deformation is driven by the predicted object-region boundary. This is also the focus of our new interaction method.

Let us assume that each constraint corresponds to a probabilistic boundary prediction module. Then, suppose we have n independent external constraints, the feature used in the k th constraint is f_k , and $L(\mathbf{v})$ denotes the label of a pixel \mathbf{v} . Our approach to combining the multiple independent modules uses the Bayes rule in order to evaluate the final confidence rate:

$$\begin{aligned} Pr(L(\mathbf{v})|f_1, \dots, f_n) &= \frac{Pr(f_1, \dots, f_n|L(\mathbf{v}))Pr(L(\mathbf{v}))}{Pr(f_1, \dots, f_n)} \\ &\propto Pr(f_1|L(\mathbf{v})) \dots Pr(f_n|L(\mathbf{v}))Pr(L(\mathbf{v})) \end{aligned} \quad (2.3)$$

For each independent module, the probability $Pr(f_k|L(\mathbf{v}))$ is estimated based on the active volume model's interior statistics. Considering a module using intensity statistics, the object region can be predicted according to the current model-interior intensity distribution. For instance, for a pixel \mathbf{v} with intensity feature value $I(\mathbf{v}) = i$ where i

ranges from 0 to 255, we have:

$$\begin{aligned}
 Pr(i|I) &= Pr(i, obj|I) + Pr(i, non_obj|I) \\
 &= Pr(i|obj, I)Pr(obj|I) \\
 &+ Pr(i|non_obj, I)Pr(non_obj|I)
 \end{aligned} \tag{2.4}$$

In the equation, the intensity distribution over the entire image I , $Pr(i|I)$, is known, and we estimate the object-interior distribution $Pr(i|obj, I)$ by the current model-interior intensity distribution. Therefore, the background distribution can be derived:

$$Pr(i|non_obj, I) = \frac{Pr(i|I) - Pr(i|obj, I)Pr(obj|I)}{Pr(non_obj|I)} \tag{2.5}$$

Assuming a uniform prior, $Pr(obj|I) = Pr(L(\mathbf{v}) = obj) = 0.5$ and $Pr(non_obj|I) = Pr(L(\mathbf{v}) = non_obj) = 0.5$, in Eqn. 2.5, we are able to compute the background probability $Pr(i|non_obj, I)$. Applying the Bayesian Decision rule, we can obtain a binary map P_B that represents the predicted object region; that is, $P_B(\mathbf{v}) = 1$ if $Pr(i|obj, I) \geq Pr(i|non_obj, I)$, and $P_B(\mathbf{v}) = 0$ otherwise. We then apply a connected component analysis algorithm on P_B to retrieve the connected component that overlaps the current model. This connected region is considered as the current ROI. Let us denote the signed distance transform of the current model's shape as Φ_M , and the signed distance transform of the ROI shape as Φ_R . Then the region-based external energy term is:

$$E_R = \int_0^1 \Phi_M(\mathbf{v}(s))\Phi_R(\mathbf{v}(s))ds \tag{2.6}$$

The multiplicative term provides two-way balloon forces that deform the model toward the predicted ROI boundary.

2.3 Motivation for Employing User Interaction

In this section, we discuss some failure cases using AVM and give an analysis as to why AVM fails. Figure 2.1 shows a synthetic example for which AVM fails. As can be seen, the left part of the polygon is missing in the projected ROI.

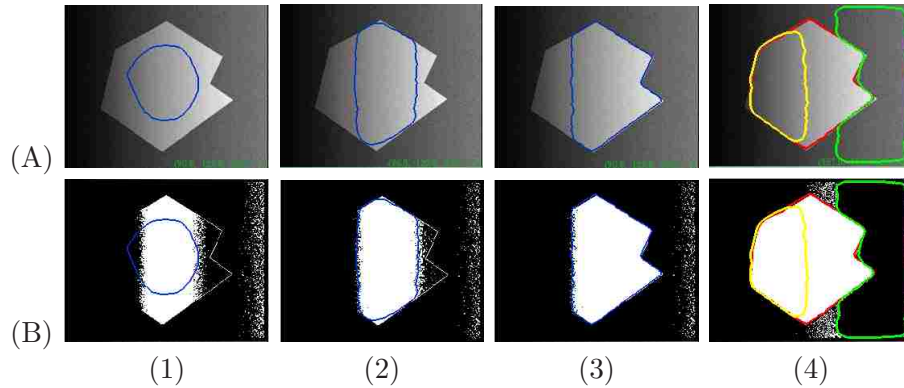


Figure 2.1: Synthetic example segmentation using AVM. (A) The model on the original image. (B) The binary map P_B estimated by intensity-based likelihood maps and applying the Bayesian Decision rule. (1) Initial model. (2) The model after 12 iterations. (3) Final result after 26 iterations without interactive polygons ($p = 0.74$, $q = 0.97$). (4) Final result with interactive polygons (merge polygon in yellow and split polygon in green) ($p = 0.99$, $q = 0.98$). (Note: all figures in the chapter are best viewed in color)

In the current AVM, we only calculate the intensity distribution of the foreground and background and use them to obtain a ROI binary map. The idea works fine for many medical images where the assumption that the intensity is homogeneous within the object holds. For certain non-homogeneous images, AVM may be confused when pixels with a certain intensity in one area of the image belong to the foreground while other pixels with the same intensity belong to the background in a different part of the image. Using the image in Figure 2.2 as an example, pixels in the area A have the same intensity as pixels in the area B while they belong to the foreground and the background, respectively. We plot the intensity distribution of the foreground and the background in Figure 2.2(b). As can be seen, the distribution of the background shown

in the blue box dominates the distribution of the foreground, thus we treat foreground pixels with intensity values in the blue box as the background ones, and the left portion of the polygon is mistakenly marked as the background in the ROI binary map. The segmentation contour could not deform to the left part of the polygon because of the external forces coming from the region term.

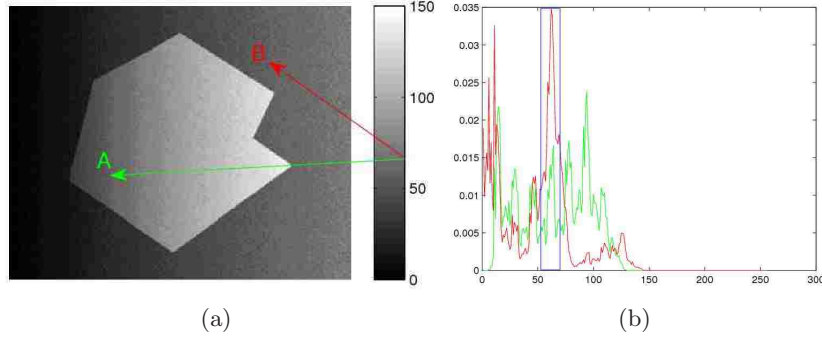


Figure 2.2: Synthetic example (a) The intensity overlap. (b) The intensity distribution of the foreground (green) and the background (red).

Similar situations arise in some medical images. One example is magnetic resonance (MR) images, which exhibit intensity inhomogeneity due to the bias field. The bias can cause serious misclassification when intensity-based segmentation algorithms are used. Essentially, the misclassification is due to an overlap of the intensity ranges of different tissues introduced by the bias, so that voxels in different tissues are not separable by their intensities. Intensity inhomogeneities also often occur in images of other modalities, such as X-ray and computed tomography images [60]. Figure 2.3 shows a failure case in an X-ray image where the part in the red circle is mistakenly left out.

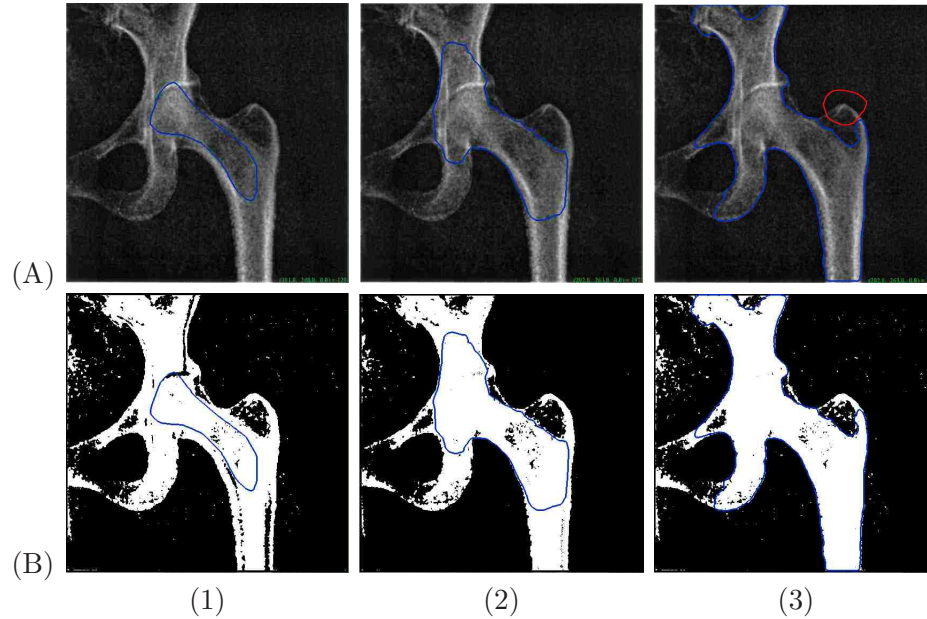


Figure 2.3: X-ray image segmentation using AVM. (A) The model on the original image. (B) The binary map P_B estimated by intensity-based likelihood maps and applying the Bayesian Decision rule. (1) Initial model. (2) The model after 23 iterations. (3) Final result after 84 iterations.

2.4 Integrating User Interaction to AVM

The analysis in the previous section suggests that intensity information itself is not sufficient for segmentation of inhomogeneous images. User interaction can help in these cases since human eyes are normally good at distinguishing the background and the foreground with high reliability. The interaction we introduce here consists of polygons that the user draws in the background or in the foreground to help the active contour deform to the desired boundary. The polygons in the background, i.e., split polygons, are there to prevent the active contour from leaking to the background. The polygons in the foreground, i.e., merge polygons, are there to help the active contour flow into the foreground.

With merge/split polygons, we cannot assume uniform priors anymore in Eqn. 2.5.

Instead, we calculate the priors for each pixel according to its distances to the merge or split polygons. We use the following notations to describe how to calculate binary ROI map P_B using the priors of $Pr(obj|I)$, $Pr(non_obj|I)$.

1. Let $s(n, m)$ be a pixel on image Ω . s 's position is (n, m) , $n = 1, \dots, N$, $m = 1, \dots, M$. s 's intensity is $i = I(n, m)$;
2. $P_k, k = 1, \dots, K$: K merge polygons drawn by users including the original deformable contour which is viewed as foreground;
3. $Q_l, l = 1, \dots, L$: L split polygons drawn by users;
4. $P_B(n, m)$: binary ROI image;

We incorporate the spatial information into the distribution function by adding priors which are functions of the distance to interactive polygons. The original contour and interactive polygons can deform in parallel or separately by users' choices. In our interactive model, merge/split polygons can also be deformed using AVM. The region statistics of the original contour are incorporated in a similar way since we treat the original contour as a special case of a merge polygon. If a pixel is in a merge or split polygon, then the corresponding pixel in the ROI binary map is set to 255 or 0 respectively. Otherwise, if the intensity value of a background pixel in the original contour appears in both the background and foreground (for instance, pixels in the blue box in Figure 2.2(b)) we then calculate its priors according to its distances to the split polygon and the merge polygon, if they exist, as shown in Eqn. 2.7.

$$\begin{aligned}
 Pr(obj|I) &= \frac{d(s, P)^{-1}}{d(s, P)^{-1} + d(s, Q)^{-1}} \\
 Pr(non_obj|I) &= \frac{d(s, Q)^{-1}}{d(s, P)^{-1} + d(s, Q)^{-1}}
 \end{aligned}
 \tag{2.7}$$

Where $d(s, P_k)$ is defined as the distance between pixel s and polygon P_k , so $d(s, P) = \min_k d(s, P_k)$ and $d(s, Q) = \min_l d(s, Q_l)$.

Then we use Equation 2.5 to calculate $Pr(i|non_obj, I)$. Algorithm 1. summarizes our algorithm.

Algorithm 1 Calculate binary ROI map P_B

```

for  $n = 1$  to  $N$  do
  for  $m = 1$  to  $M$  do
    if  $s(n, m)$  inside any  $P_k, k = 1, \dots, K$  then
       $P_B(n, m) \leftarrow 255$ 
    else if  $s(n, m)$  inside any  $Q_l, l = 1, \dots, L$  then
       $P_B(n, m) \leftarrow 0$ 
    else
      if  $Pr(obj|i)$  for any  $P_k, k = 1, \dots, K \neq 0$  and  $Pr(obj|i)$  for any  $Q_l, l = 1, \dots, L$ 
       $\neq 0$  for current background pixels then
         $Pr(obj|I) \leftarrow \frac{d(s,P)^{-1}}{d(s,P)^{-1}+d(s,Q)^{-1}}$ 
         $Pr(non\_obj|I) \leftarrow \frac{d(s,Q)^{-1}}{d(s,P)^{-1}+d(s,Q)^{-1}}$ 
         $Pr(i|non\_obj, I) \leftarrow \frac{Pr(i|I)-Pr(i|obj,I)Pr(obj|I)}{Pr(non\_obj|I)}$ 
      else
         $Pr(obj|I) \leftarrow 0.5$ 
         $Pr(non\_obj|I) \leftarrow 0.5$ 
         $Pr(i|non\_obj, I) \leftarrow \frac{Pr(i|I)-Pr(i|obj,I)Pr(obj|I)}{Pr(non\_obj|I)}$ 
      end if
      if  $Pr(i|obj, I) \geq Pr(i|non\_obj, I)$  then
         $P_B(n, m) \leftarrow 255$ 
      else
         $P_B(n, m) \leftarrow 0$ 
      end if
    end if
  end for
end for

```

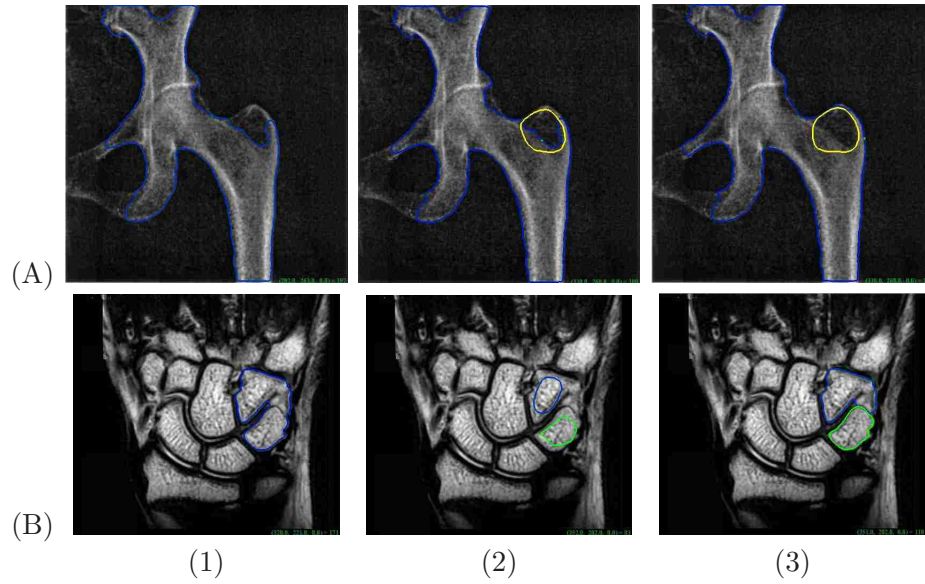


Figure 2.4: Segmentation using AVM. (A) The X-ray image. (B) The MRI wrist image. (1) Segmentation result without interaction (X-ray: $p = 0.98$, $q = 0.98$; Wrist: $p = 0.97$, $q = 0.62$). (2) The initial contour and interactive polygons. (3) Final result (Spine: $p = 0.99$, $q = 0.99$; Wrist: $p = 0.98$, $q = 0.97$).

2.5 Experiments

Figure 2.4(A) shows our experiment with only merge polygons for the X-ray image. Figure 2.4(B) shows the experiment with only split polygons for the image. Figure 2.5 shows the result with both merge polygon and split polygon for an X-ray foot image. These experiments show that AVM can perform reasonably well for some complicated images with convenient user interaction. We use sensitivity (p) and specificity (q) to measure the accuracy between segmentations with and without interactive polygons.

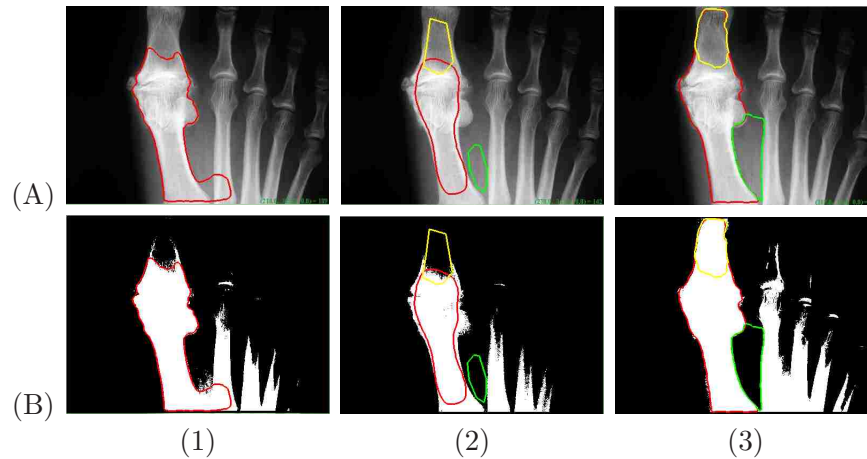


Figure 2.5: segmentation using AVM. (A) The X-ray foot image. (B) The binary ROI image. (1) Segmentation result without interaction ($p = 0.78$, $q = 0.95$). (2) The initial contour and interactive polygons. (3) Final result ($p = 0.97$, $q = 0.99$).

2.6 Discussion

In this chapter, we introduce a new user interaction method called interactive polygons to help with the segmentation of difficult images such as inhomogeneous medical images. The novelty of this interaction method is to integrate region statistics for interactive polygons with the original parametric active contour model. Compared to the conventional parametric active contour model, it has the capability of user interaction with the initial segmentation results thus the model can deform to the accurate boundary even in some difficult cases. Anyone can use our model to achieve desirable segmentation results even they may have no knowledge of segmentation models. Although we illustrated our model on gray scale medical images, the proposed model can be easily extended to color images or texture images by adding color or texture features. Also our model can be extended to 3D images by finite element method. The interactive polygons can be placed on original 3D segmentation. The 3D segmentation process can be carried out using the same methodology, with 3D rather than 2D coordinate system.

This is a subject of a future research.

Chapter 3

Interactive Level Set Method with User Scribbles

3.1 Introduction

In this chapter, we will explore a new interactive segmentation based on level sets. First, we need a proper level set formulation which can segment nature color images.

Like snakes [55], the desired segmentation is achieved by deforming a contour to minimize a given energy functional in level set formulation [63, 101], where a contour is presented as a level set of a scalar function of high-dimensional variables. Level sets can be classified into two categories: edge-based and region-based. Edge-based active contour models use image gradients in order to obtain object boundaries [23] while the region-base active contour models of Zhu and Yuille [118] use region statistics such as distributions, intensity histograms, texture maps, or structure tensors. Image gradients are highly localized image information; for this reason the edge-based level sets are very sensitive to image noise and highly dependent on initial curve placements.

But one advantage of edge-base level sets is that there are no global constraints on the image. Therefore, correct segmentation can still be achieved where regions of object and background are heterogeneous. In contrast to edge-based level sets, the region-based level sets are robust to initial curve placement and exhibit less sensitivity to image noise. Since most region-based level sets use the global region statistics, they will fail in many cases for nature images where the object to be segmented cannot be easily distinguished by global statistics. The objects in nature images may consist of many heterogeneous regions.

Reference [58] shows a situation where the foreground and background are heterogeneous and share nearly the same statistical model; we show this in Figure 3.1. This synthetic image is segmented improperly by a standard region-based algorithm [24], but correctly by an edge-based algorithm [23].

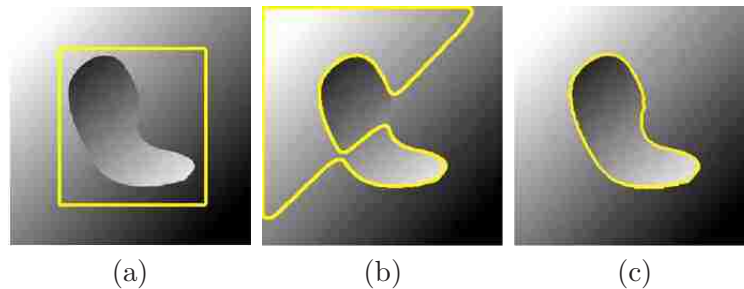


Figure 3.1: Synthetic image of a blob with heterogeneous intensity on a background of similar heterogeneous intensity. (a) Initial contour. (b) Unsuccessful result of region-based segmentation. (c) Successful result of edge-based segmentation technique [58]

To accurately segment the objects in nature images, a new class of active contour energies should be considered which use local information, but also incorporate the benefits of region-based techniques including distinguishing regions by color and texture. The localized region-based level set proposed by [58] is such a level set formulation; it uses local rather than global image statistics, and evolves a contour based on this local information. This approach, however, cannot work on nature images since it does

not integrate color or texture information into the formulation. So in this chapter, in order to segment objects in nature images, in our first step we modify the localized region-base level set framework to model regions by statistics of energy in color and texture information.

For nature images, user interactions are necessary either to identify the correct ROI or correct segmentation results in iterations. As a generic segmentation tool, the level sets suffer from an important drawback: current formulations do not allow much user interaction. Upon initialization, boundaries deform to the final segmentation without the user being able to guide or correct the segmentation. So for the second step in this chapter, we will incorporate user interactions into the modified framework and allow users to fix topologies and segmentation results easily by simply adding background and foreground scribbles. The whole process is iterative. In every iteration, given the image and the user scribbles, our new framework can determine the most likely segmentation.

In our framework, a user scribble is treated as an independent observation in the Bayesian framework and considered equally with the image features. The user scribbles are seen as a soft constraint rather than a hard constraint since user scribbles are just another source of information beside image features. When a location is scribbled as object by the user then areas in its vicinity are given a certain probability of being part of the object. The contour deforms driven by a competition of the image features and the user scribbles in the resulting framework.

We now briefly summarize the contents of the remainder of this chapter. In the following section, we introduce the general framework for localizing region-based flows based on [58]. In Section 3.3, we introduce our new energies for color and texture implemented in this framework. In Section 3.4, we discuss how to model user scribbles and integrate user scribbles into the framework. We go on to show some experiments

in Section 5.6. Here, we show how user scribbles are used to improve segmentation results on challenging images. In Section 5.7, we make concluding remarks.

3.2 Localized Region-based Level Set

In this section, we give a brief introduction to local region-based framework for guiding active contours; details can be found in [58]. In most region-based level sets, the foreground and background regions are modeled with global statistics. For most nature images, however, global statistics for the foreground and background are not separable. The foreground and background are only separable in small local regions. This is the motivation for the localized region-base level set.

In the localized region-base level set, the foreground and background are described in terms of small local regions. Local energies can be computed at each point along the curve where local region is split into local interior and exterior by the curve. Each point of the curve evolves separately to minimize or maximize the energy in its own local region.

We let I denote a given image defined on the domain Ω , and let C be a closed contour represented as the zero level set of a signed distance function ϕ . The interior of C can be defined by the following approximation of the smoothed Heaviside function:

$$\mathcal{H}\phi(x) = \begin{cases} 1, & \phi(x) < -\epsilon \\ 0, & \phi(x) > \epsilon \\ \frac{1}{2}\{1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi\phi(x)}{\epsilon})\}, & \text{otherwise} \end{cases}$$

Then the exterior of C is defined as $(1 - \mathcal{H}\phi(x))$. The area around the curve can be specified by the derivative of $\mathcal{H}\phi(x)$, a smoothed version of the Dirac delta:

$$\delta H\phi(x) = \begin{cases} 1, & \phi(x) = 0 \\ 0, & |\phi(x)| < \epsilon \\ \frac{1}{2\epsilon}\{1 + \cos(\frac{\pi\phi(x)}{\epsilon})\}, & \text{otherwise} \end{cases}$$

If x and y are denoted as independent spatial variables while each representing a point in Ω , a mask function in terms of a radius parameter r is defined:

$$\mathcal{B}(x, y) = \begin{cases} 1, & \|x - y\| < r \\ 0, & \text{otherwise} \end{cases}$$

$\mathcal{B}(x, y)$ will be 1 when the point y is within a ball of radius r centered at x , and 0 otherwise. It is used to mask the region shown in Figure 3.2. Using $\mathcal{B}(x, y)$, we now define an energy functional in terms of a generic internal energy, F :

$$E(\phi) = \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} \mathcal{B}(x, y) \cdot F((I_y), \phi(y)) dy dx \quad (3.1)$$

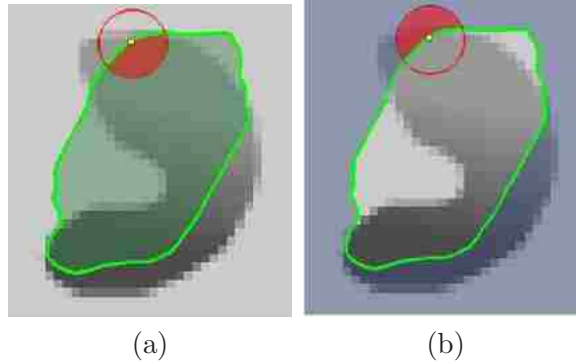


Figure 3.2: Ball region is considered at each point along the contour, which is split by the contour into local interior and local exterior. (a) the local interior is the shaded part of the circle; (b) the shaded part of the circle is the local exterior [58]

In this framework, we only consider contributions from the points near the contour

while ignoring inhomogeneity in the region far away. Thus the framework can deal with many objects in nature images. In Equation 3.2, every point x selected by $\delta\phi(x)$ is masked with $\mathcal{B}(x, y)$ to ensure that F operates only on local image information about x . This term of the energy is just the sum of F values for every $\mathcal{B}(x, y)$ neighborhood along the zero level set. In the end, a regularization term for smoothness is added to penalize the arclength of the curve with a weight of α . The final energy is as follows:

$$E(\phi) = \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} \mathcal{B}(x, y) \cdot F((I_y), \phi(y)) dy dx + \alpha \int_{\Omega_x} \delta\phi(x) \|\nabla\phi(x)\| dx \quad (3.2)$$

3.3 Modeling Region Separation Energy for Nature Images

Equation 3.2 formulates the framework in terms of a generic internal energy F . We now introduce histogram separation energy in term of color information from nature images to replace F .

Let $P_{in}(z)$ and $P_{out}(z)$ to be two smoothed feature histograms computed from the global interior and exterior regions of a segmented image I using z feature bins, where z can be intensity for gray-scale images or feature vectors in feature space Z for color images.

After calculating feature histograms from interior and exterior regions, we use the Bhattacharyya coefficient [11] to compare probability density functions and obtain a scalar corresponding to the similarity of the two histograms.

To make interior and exterior regions to be as different as possible, we adopt an image segmentation energy proposed by Michailovich et al. [74] to minimize this measure. This energy is called the histogram separation energy, defined as follows

$$E_{HS} = \int_z \sqrt{P_{in}(z)P_{out}(z)} dz \quad (3.3)$$

In the localized case, $P_{in,x}(z)$ and $P_{out,x}(z)$ indicate the feature histograms in the local image regions $B(x, y) \cdot \mathcal{H}\phi(y)$ and $B(x, y) \cdot (1 - \mathcal{H}\phi(y))$, respectively. By substituting the local equivalents for $P_{in}(z)$ and $P_{out}(z)$, Equation 3.3 becomes:

$$F_{HS} = \int_z \sqrt{P_{in,x}(z)P_{out,x}(z)} dz \quad (3.4)$$

By substituting F into Equation 3.2, the localized version of this flow is as follows

$$E(\phi) = \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} \mathcal{B}(x, y) \cdot \int_z \sqrt{P_{in,x}(z)P_{out,x}(z)} dz dy dx + \alpha \int_{\Omega_x} \delta\phi(x) \|\nabla\phi(x)\| dx \quad (3.5)$$

where K is a Gaussian kernel.

With the Bhattacharyya measure, the global version of this flow can segment the object only where the feature profile of the entire object and the entire background are separable, while the localized version can effectively separate locally heterogeneous regions.

Unfortunately, if using color as the segmentation features for nature images, 3-D histograms must be used. This may be computationally expensive for large images. We adopt the Fast ICA algorithm [2] to reduce the dimensionality of the feature space from 3 to 2. This algorithm projects the colors of pixel x onto the subspace spanned by their two independent components.

3.4 Integrating User Scribbles as Energy Function

Now we have the localized region-base level sets framework which can work on nature images. In this section, we will discuss how to model user scribbles as energy functions and integrate them into the framework.

3.4.1 Modeling User Scribbles as Observation

Let us assume we are given some scribbles from a user who marked certain image locations as “object” or as “background”. How should such additional information be integrated in the framework? We believe that user scribbles should be treated as another observation in addition to the input image. This will be detailed in the following for the segmentation of a single object of interest in the image plane Ω . The object of interest may be multiply-connected and the segmentation will be defined in terms of a level set function $\phi : \Omega \rightarrow \mathbb{R}$, where image locations x with $\phi(x) \geq 0$ denote parts of the object and locations x with $\phi(x) < 0$ denote the background.

Let us assume we are given an image $\phi : \Omega \rightarrow \mathbb{R}$ and user scribbles $L : \Omega \rightarrow \{-1, 0, +1\}$ where the label values reflect the user scribbles, namely:

$$L(x) = \begin{cases} +1, & x \text{ marked as 'object'} \\ -1, & x \text{ marked as 'background'} \\ 0, & x \text{ not marked} \end{cases}$$

Maximizing the a posteriori probability, the segmentation is computed by the following equation in Bayesian approach:

$$P(\phi|I, L) = \frac{P(I, L|\phi)P(\phi)}{P(I, L)} \quad (3.6)$$

with respect to ϕ .

The denominator in the above equation can be ignored since it does not depend on the segmentation. We make the assumption that the input image and the user scribbles are independent in order to obtain a simple solution. Thus, we have

$$P(I, L|\phi) = P(I|\phi)P(L|\phi) \quad (3.7)$$

This assumption can be removed by using more accurate modeling of the dependency between image and user scribbles. We leave this to future work. Now we move on to model the likelihood $P(L|\phi)$ for user labeling L from scribbles, given a segmentation ϕ . We have the following equation:

$$P(L|\phi) = \prod_{x \in \Omega} \prod_{y \in \Omega} P(L(x), x|\phi(y), y) dx dy \quad (3.8)$$

With the additional assumptions that label values at different locations are independent (a product over all x) and the values of ϕ at different locations are also independent (a product over all y).

The term $P(L(x), x|\phi(y), y)$ specifies the likelihood of a label $L(x)$ at location x given the segmentation $\phi(y)$ at location y . Based on the observation that if the level set function at a location y is positive, i.e. $sign(\phi(y)) = 1$, then a point x in its vicinity is more likely to be labeled object and less likely to be labeled background, we will model this as an exponential function:

$$P(L(x), x|\phi(y), y) \propto exp(\lambda L(x) sign(\phi(y)) k_\sigma(x, y)) \quad (3.9)$$

Where $k_\sigma(x, y)$ is Gaussian kernel:

$$k_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) \quad (3.10)$$

The weight parameter λ specifies how strong the effects of the user scribbles are on the segmentation. The parameter σ defines the spatial range where a point labeled by users will affect the segmentation. For the localized version, it is masked by $\mathcal{B}(x, y)$ so its effect is limited to the local region. More sophisticated models for the user interaction than the Gaussian kernel can be developed, which we leave to future work.

3.4.2 Integrating Scribbles into Level Set Framework

Putting all together, we have an equation of total energy:

$$E(\phi) = E_{region}(\phi) + \alpha E_{smooth}(\phi) + \lambda E_{scribbles}(\phi) \quad (3.11)$$

where the region separation energy is given by Equation 3.12:

$$E_{region}(\phi) = \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} \mathcal{B}(x, y) \cdot \int_z \sqrt{P_{in,x}(z)P_{out,x}(z)} dz dy dx \quad (3.12)$$

The smoothing term is given by:

$$E_{smooth}(\phi) = \int_{\Omega_x} \delta\phi(x) \|\nabla\phi(x)\| dx \quad (3.13)$$

The energy associated with the user scribbles is given by:

$$\begin{aligned} E_{scribbles}(\phi) &= -(1/2) \int \int \mathcal{B}(x, y) L(x) \text{sign}(\phi(y)) k_\sigma(x, y) dx dy \\ &= -(1/2) \int L_\sigma(y) \text{sign}(\phi(y)) dy \end{aligned} \quad (3.14)$$

with the localized Gaussian-smoothed label function

$$L_\sigma(y) = \int \mathcal{B}(x, y)L(x)k_\sigma(x, y)dx \quad (3.15)$$

Since the sign function can be expressed by the Heaviside function as $sign(\phi) = 2\mathcal{H}\phi(x)-1$, we can combine the three energies to obtain a total energy of the form:

$$\begin{aligned} E(\phi) &= \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} \mathcal{B}(x, y) \cdot \int_z \sqrt{P_{in,x}(z)P_{out,x}(z)} dz dy dx \\ &+ \alpha \int_{\Omega_x} \delta\phi(x) \|\nabla\phi(x)\| dx \\ &+ \lambda \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} L_\sigma(y) dy \end{aligned} \quad (3.16)$$

Minimization of this energy can be done by evolving the following descent equation:

$$\begin{aligned} \frac{\partial\phi}{\partial t}(x) &= \delta\phi(x) \int_{\Omega_y} \frac{\mathcal{B}(x, y)\delta\phi(y)}{2} \\ &\times [F_{HS}(\frac{1}{A_{out}} - \frac{1}{A_{in}}) + \int K(z - I(y)) \\ &\times (\frac{1}{A_{out}} \sqrt{\frac{P_{out,x}(z)}{P_{in,x}(z)}} - \frac{1}{A_{in}} \sqrt{\frac{P_{in,x}(z)}{P_{out,x}(z)}}) dz] dy \\ &+ \alpha \delta\phi(x) div(\frac{\nabla\phi(x)}{|\nabla\phi(x)|}) \\ &+ \lambda \delta\phi(x) L_\sigma \end{aligned} \quad (3.17)$$

The goal of the first term (weighted by α) is to minimize the length of the segmenting contour. The second term drives the contour to separate the two feature distributions in the object and background regions. The last term (weighted by λ) specifies the effect of the user scribbles, driving the contour to favor the segmentation of object and background as indicated by the user scribbles.

3.5 Experimental Results

First, we illustrate the segmentation results for the modified localized region-based level set. Figure 5.3 shows two nature images where object or background consists of heterogeneous regions with different color and texture. Our modified localized level set can work very well on these nature images.

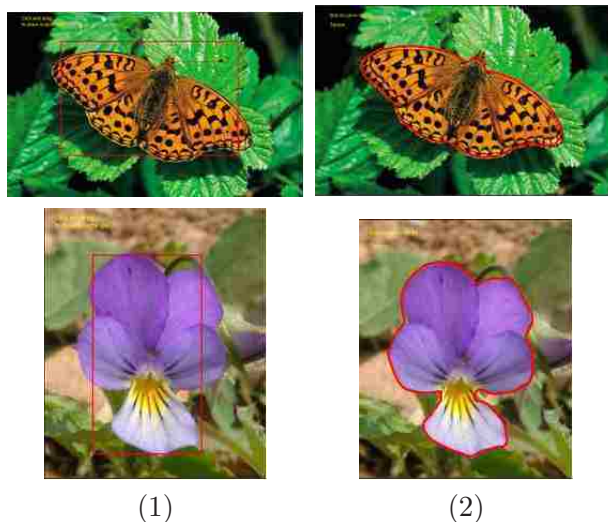


Figure 3.3: (1)Initial contour for localized level sets; (2) Final segmentations.

In practice, the segmentation of an image with user scribbles is obtained in the following manner:

1. The user selects areas of object by drawing the initial contour with the mouse
2. The algorithm evolves the initial contour by localized region-based level set
3. The user corrects errors in the segmentation by adding additional foreground or background scribbles
4. The algorithm updates the contour, using the scribbles provided by the user

The whole process can be iterated if necessary. In Figures 3.4, we present respective steps from the applications of our interactive level set method to the segmentation of nature images. In the figure, Figures 3.4(1) shows the initial contour for localized region-based level set. Figures 3.4(2) shows the initial segmentation result after the first iteration. Figures 3.4(3) shows the user scribble to mark background or foreground. The final segmentation result is shown in Figures 3.4(4).

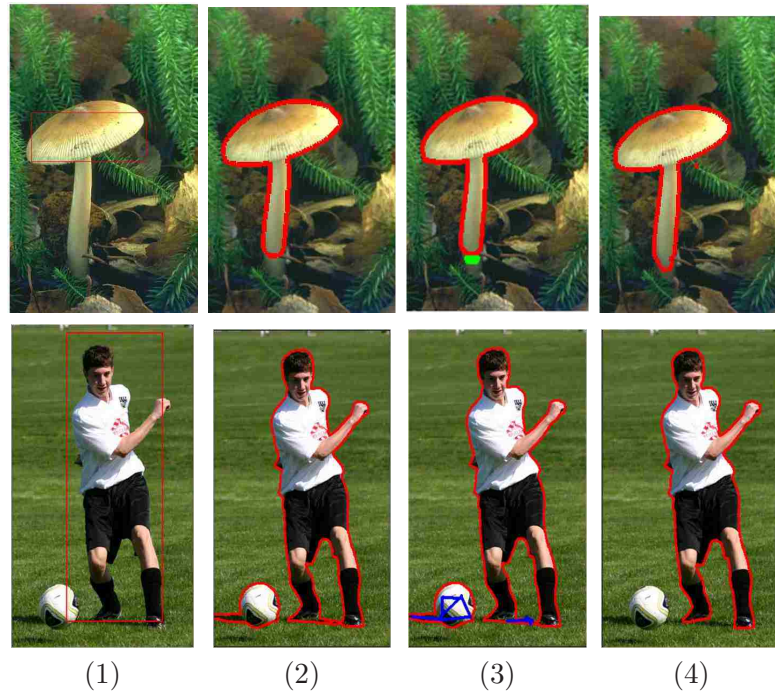


Figure 3.4: (1)Initial contour for localized level sets; (2) Initial segmentations; (3) User scribbles for background (blue) or foreground (green); (4) New segmentation result.

3.6 Discussion

In this chapter, we present a Bayesian formulation of level set segmentation which allows the integration of user scribbles with the input image. In order to work on nature

images, we first modify the localized region-based level set with Bhattacharyya region separation energy where we only consider local features of the image. The user scribbles are treated as independent observations of an underlying scene. The segmentation process computes the most probable segmentation given the image features and the labeling specified by user scribbles which identify foreground or background. The effect of the user scribbles is characterized by a Gaussian kernel in a local region. The user can easily interact with the segmentation process with scribbles in the iterations to correct the segmentation result. We believe that our approach constitutes a first step toward a more sophisticated probabilistic modeling of user input in the context of image segmentation. We can design a complicated model for user scribbles depending on the features of images. The other future works is to implement the model on GPU to improve speed for a smooth interaction and fast feedback from the segmentation process. One issue of interactive level set is that the initial contour has to be a closed contour while the following correcting user interactions are scribbles that are not necessarily closed contours. So we can adopt scribbles as the initial user interactions instead to add more flexibility to the model.

Chapter 4

Interactive Segmentation Through Label Propagations on Hierarchical Hypergraph

4.1 Introduction

In this chapter, we develop a new framework using graph-based models for interactive segmentation. In this framework, an image is modeled as nodes of a hypergraph. Belief propagation is then used to propagate the labels of seed pixels to the remaining unlabeled pixels. We implemented three belief propagation algorithms: discrete belief propagation; and Gaussian belief propagation in which the label variables are modeled as discrete and Gaussian variables, respectively. To make the algorithm more efficient, we also integrated a hierarchical scheme for label propagation into our framework. Experiments show that our new algorithm can achieve comparable results with state-of-the-art graph-based interactive methods [14, 31, 48], while being more computationally

efficient. Another contribution is that we integrated a new user interaction into the framework which allows users to change the connectivity of segmentation results.

The remainder of this chapter is organized as follows. Section 4.2 reviews the usage of scribbles in the graph-based models. Section 4.3 presents our framework. A new user interaction for connectivity of segmentations is presented in Section 4.4. Section 4.5 proposes a scribble recommendation system integrated into the framework which recommends where to add scribbles during iteration. We illustrate experimental results of our framework in Section 4.6. We summarize this chapter in Section 4.7.

4.2 Literature Review

In this section, we introduce the usages of user scribbles in graph-based models. Then we give a brief overview of segmentation methods using graph-based models and how scribbles are used in some of these interactive segmentation methods.

4.2.1 Usage of Scribbles and Limitations

There are multiple advantages of using scribbles. One is flexibility. It is easy to include user scribbles in graph-based segmentations and make the whole process iterative. The second is intuitiveness. The meaning of scribbles is clear to users so it is easy for users to learn and apply scribbles in the segmentation process. The main usages of scribbles are listed below.

1. Identify Regions of Interest (ROI)

For segmentation of nature images, one goal is to find “meaningful” segmentations. The “meaningfulness” here often refers to their semantic context. For example, in many situations, users are more likely to look for certain objects

such as a “person” or an “animal”. These objects could consist of several inhomogeneous regions. Since many automatic segmentation algorithms only work on the low level features of images, there is a so-called “semantic gap” between the outputs of these algorithms and semantically “meaningful” segmentations. In this case, the “semantic gap” can be narrowed by engaging humans in the segmentation process. Interactive segmentations exploit user interactions to roughly identify the Regions of Interest (ROI). Users can put dots or lines on the objects they want to extract. Good interactive segmentations may potentially obtain accurate object boundaries.

2. Provide seeds with pre-defined labels

Another purpose of user scribbles is to provide seed pixels with pre-defined labels. In some classification based interactive segmentations, these seeds with pre-defined labels along with unlabelled pixels are viewed as a training dataset in a semi-supervised learning process [32][117]. The classification process can take these labelled and unlabelled data points to train a classifier. The other usage of these seeds is to build models for each label (e.g., object or background label) such as the MRF model or GMMRF model [14][18]. In some region growing like interactive segmentations, these labelled pixels are also used as growing seeds [107][78]. In addition, some segmentation algorithms use scribbles to design complicated weight functions for graph-based models [85][6].

3. Control topology

User scribbles are used as a way to control the topologies of segmentations by merging several inhomogeneous regions or splitting homogeneous ones. For instance, users can put a long scribble through the whole body of a person which

indicated that this person's head, neck, torsos and legs should be connected in the segmentation. Also, there are recent research results in using scribbles to construct shape priors or connectivity priors to control topology [105][108].

4. Correct segmentations

Scribbles give users a tool in the correction process of segmentations if needed. Users can make corrections both on labels or on the erroneously segmented regions. Interactive segmentation with scribbles can update models or focus on the erroneous regions to refine results.

Although scribbles are flexible and intuitive to users, one disadvantage is that they could be tedious, since in every step users have to make a decision such as whether to assign more pixels to their correct labels.

4.2.2 Graph-based Segmentation Models

We can classify interactive graph-based segmentation models into two categories based on whether a model adopts a bottom-up or a top-down strategy: (1) Region Growing or merging models; (2) Graph Partitioning models, respectively. It should be noted that both kinds of methods may generate results that are either too coarse, or over-segmented.

Region Growing or Merging Models

Region Growing models use a bottom-up strategy. In this approach, neighboring pixels are examined and added to a region class if no edges are detected. This process is iterated for each boundary pixel in the region. If adjacent regions are found, a region-merging algorithm is used in which weak edges are dissolved and strong edges are left intact. We discuss several specific strengths of region growing here. Unlike gradient

and Laplacian methods, the borders of regions found by region growing are perfectly thin (since we only add pixels to the exterior of our region) and connected. The algorithm is also very stable with respect to noise. Most importantly, membership in a region can be based on multiple criteria. It can simultaneously take advantage of several image properties, such as gradient or gray level intensity value. There are, however, several disadvantages to region growing. First and foremost, it can be very computationally expensive. For this reason, careful design is required to implement an efficient algorithm if carefully designed. Table 4.1 lists some examples of region growing based segmentation methods in the order of the year published.

Graph Partitioning Models

In contrast to Region Growing models, Graph Partitioning methods use a top-down strategy. They view an image as a graph and try to find good partitioning of the graph. Traditional clustering methods such as K-means and, mean-shift are some of the methods used.

Though recent methods tend to be more complex, all graph partition methods have the goal of finding a partition that corresponds to a segmentation such that the boundary between segments has a small number of (weighted) edges cut, i.e., the boundary threads between as many dissimilar pixels as possible. In practice, the partition minimizes the cost function of a constrained optimization problem. One disadvantage is this constrained minimization problem is often difficult, or completely impractical. This happens in particular with Normalized Cut[98]. A number of graph partitioning methods are based on graph Laplacians. Min cuts [17] and random walks [48] are two examples. The underlying assumption in these methods is that the points are generated by a probability distribution with support on a submanifold of the Euclidean

space. Thus these methods can be viewed as spectral clustering methods. The other choice is to use some heuristic optimization methods, such as an MRF modeled graph. Examples are Simulated Annealing (SA) and Iterated Conditional Modes(ICM). Table 4.2 lists examples of graph partitioning based segmentation methods in the order of the year published.

Table 4.1: Region Growing Segmentation Methods

References	Year Published	Interactive
Multiscale Aggregation [43]	2003	No
Region Growing with Bezier Model [61]	2004	Yes
Grouping with Bias [78]	2004	Yes
Local Variations [39]	2004	No
Grow-Cut [107]	2005	Yes
Segment Extraction [5]	2008	Yes

Table 4.2: Graph Partitioning Segmentation Methods

Optimization Method	References	Year Published	Interactive
Simple Clustering	Mean Shift [28]	2002	No
	K-means [91]	2009	No
Spatial Clustering	Min Cut [17]	2001	Yes
	Normalized Cut [98]	2004	Yes
	Random Walk [48]	2006	Yes
	Transduction Segmentation [33]	2008	Yes
Heuristic	Simulated Annealing [56]	1989	Yes
	Modified Metropolis Dynamics [10]	1996	No
	Iterated Conditional Modes [42]	1996	No

User Scribbles in Graph-based Segmentation Models

We can also categorize graph-based segmentation methods based on how they use user scribbles. In Section 5.2.3, we described several usages of scribbles. Table 4.3 categorizes some published interactive segmentation methods by their usage of scribbles. For example, in [6] and [85], the segmentation and alpha matte are obtained from the fast, linear complexity, computation of weighted distances to the user-provided scribbles. [107] and [78] use scribbles as seeds to grow or group the partitions. In semi-supervised graph partitioning models, scribbles are used as labeled seeds to minimize

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

the cost functions. [18] and [14] build MRF model or GMMRF model for user labels. Fuzzy connectedness models [104][26] define how the image partitions hang together from user scribbles.

Table 4.3: Usage of User Scribbles in Interactive Segmentation Methods

Usage	References
Modulating weight function	Distance Cut [6] Adaptive Weighted Distance [85]
Providing labeled seeds for growing	Grow Cut [107] Grouping with Bias [78]
Providing labeled seeds in semi-supervised graph partitioning	Min Cut [17] Normalized Cut [98] Random Walk [48] Label Propagation [109] Transduction Segmentation [33]
Specifying training regions for learning parameters of MRF or GMMRF model	MRF with Graph Cut [18] GMMRF model [14]
Specifying connectedness	Fuzzy Connectedness [104] Iterative Fuzzy Connectedness [26]

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

The goal of image segmentation is to partition an image into several groups, i.e. assign a proper group/class label to each image pixel. In interactive image segmentation, a user pre-labels some of the pixels (i.e. assigning hard labels) and our goal is to label the rest of the pixels (i.e. computing soft labels). The intuition behind our label propagation method is a simple, iterative procedure. In every iteration, each pixel considers some label information from its spatial neighborhood and updates its own label. That is, the more similar a pixel x_i is to its neighbor pixel x_j , the more label information x_i will learn from x_j . This procedure iterates until the label of each pixel does not change, i.e., converges.

The spatial neighborhood is captured in our hypergraph model. Since the labels of the initially unlabeled pixels are updated locally and in parallel, it usually takes many

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

iterations for the hard labels to be propagated over long distances on the hypergraph to reach all unlabeled pixels. Inspired by [37], we adopt a hierarchical hypergraph so that label propagation can be performed in a coarse-to-fine manner. Thus the long range interactions between pixels can be captured by shorter ones in coarse hypergraphs. Our method uses such a hierarchy to initialize labels of the unlabeled pixels at successively finer levels. Since the more similar a pixel x_i is to pixel x_j in the feature space, the more likely the two pixels have similar labels, we can construct the hierarchical graph by grouping based on pixel similarity. The grouping predicate is described in detail in Section 4.3.1.

Given the above motivation, we propose a new interactive segmentation framework based on our hierarchical hypergraph model. It consists of two main steps: constructing a hierarchical hypergraph as image representation and propagating labels on the hierarchical hypergraph.

4.3.1 Hierarchical Hypergraph as Image Representation

In this step, we build hierarchical hypergraphs as representations of images.

Superpixel

In order to build hierarchical hypergraphs, we draw on the concept of superpixels from the computer vision literature. Superpixels are formed in a preprocessing stage to group pixels into patches. The motivation for using superpixels is that: (1) they make computation more efficient. The complexity of an image can be reduced from hundreds of thousands of pixels to only a few hundred superpixels. Classifying only superpixels reduces the degrees of freedom of the image model. Although some structures may get lost when we use superpixels, they are usually minor details. (2) They provide a

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

compact representation. Pairwise constraints, which on the original pixel grid apply only to adjacent pixels, can model much longer-range interactions between superpixels. (3) They are perceptually meaningful. Each superpixel is a perceptually consistent unit, i.e. all pixels in a superpixel are most likely uniform with respect to, for instance, color and texture, therefore they can act as a region of support for a feature vector.

The superpixel technique we use in our hypergraph model is called “TurboPixels” [59]. This is a geometric-flow based algorithm and produces segments that on one hand respect local image boundaries, while on the other hand limit under-segmentation through a compactness constraint. The method is very fast, with a complexity that is approximately linear in image size. TurboPixels have the following properties that meet our needs in building a hierarchical hypergraph: (1) uniform size and coverage. The resulting superpixels are approximately uniform in size and shape. (2) Connectivity. Each superpixel represents a simply connected set of pixels. (3) Compactness. In the absence of local edge information, superpixels remain compact. (4) Smooth, edge-preserving flow. Superpixel boundaries coincide with image edges. (5) No superpixel overlap. Any pixel is only assigned to one superpixel. Figure 4.1 shows one example of TurboPixels.

Hierarchical Hypergraph

A hypergraph is an extension of a graph in which edges are allowed to connect arbitrary, non-empty sets of vertices. Similar to graphs, hypergraphs can be used to represent images [88] [79]. A comparison between a simple graph and a hypergraph is illustrated in Figure 4.2. It is worthwhile highlighting the several desirable aspects of hypergraph based image segmentation: (1) Unlike commonly used pixel-wise similarity, hypergraphs consider patch based homogeneity, which is arguably more meaningful; (2)



Figure 4.1: Turbopixels

Hypergraphs enable a user to directly specify soft constraints to be considered as edges (called hyperedges); (3) Hypergraphs are suited to represent the spatial relationship among superpixels, which are not defined on a uniform grid. Combining superpixels generated by the TurboPixels method with hypergraphs leads to increased computational efficiency and enables implementation of our hierarchical segmentation scheme.

Like simple graphs, hierarchical hypergraphs can be built by grouping level by level. Figure 4.3 shows such an example. The bottom level hypergraph is called the level-0 hypergraph. The upper level hypergraphs are called level-1, level-2 hypergraph, and so on.

Building Hierarchical Hypergraph

We first build level-0 hypergraph starting from the grid graph. Fig. 4.4(a) illustrates a 4-connected grid image graph $G(V, E)$. Each edge is associated with a similarity weight, which is defined in Equation 4.1.

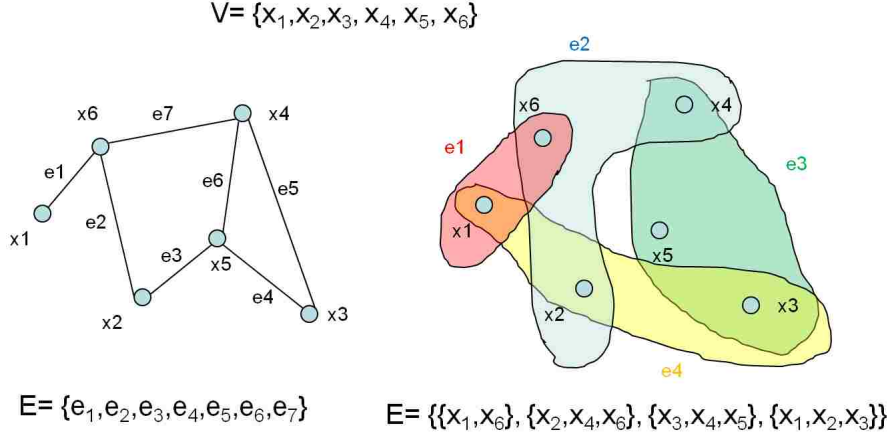


Figure 4.2: A simple graph (left) vs. a hypergraph (right)

$$w_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma}} & x_j \in N_i \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

where $\sigma > 0$ is a free parameter and N_i are the neighboring pixels of pixel x_i . Equation 4.1 has widely been used in many graph-based methods for calculating the edge weights [48]. To make the similarity insensitive to σ , we normalize each w_{ij} as $w_{ij} = \frac{w_{ij}}{\max w_{ij}}$ so that all the similarities fall within the range $[0, 1]$.

In Figure 4.4(b), the blue pixels in this grid graph belong to cluster α and the red pixels to cluster β . On level-0, clusters are superpixels. The hyperedge between cluster α and β consists of the neighboring pixels which connect cluster α and β as shown in Figure 4.4(b).

After construction of level-0 hypergraph, we group the clusters level by level. The grouping predicate compares the inter-cluster differences to the intra-cluster differences, which is similar to [39]. We define the internal difference of a cluster α to be the largest weight in the minimum spanning tree of the cluster, $MST(\alpha, E)$ (as shown in red lines

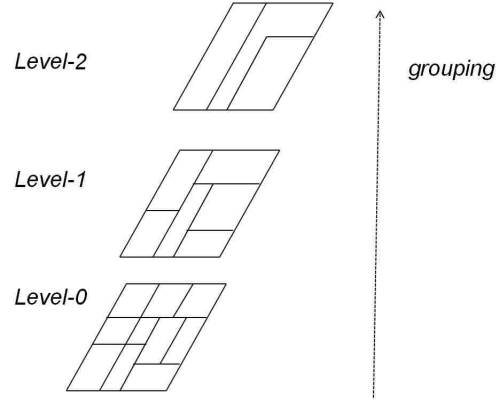


Figure 4.3: Building a hierarchical hypergraph by grouping patches level by level.

in Figure 4.4(b)). That is,

$$I(\alpha) = \max_{e \in MST(\alpha, E)} w(e) \quad (4.2)$$

We also define the difference between two clusters α , β to be the minimum weight on the simple edge in the hyperedge connecting these two clusters. That is,

$$D(\alpha, \beta) = \min_{e \in E'(\alpha, \beta)} w(e) \quad (4.3)$$

where $E'(\alpha, \beta)$ is the set of hyperedges between cluster α and cluster β .

If there is no hyperedge connecting cluster α and β we let $D(\alpha, \beta) = \infty$. In the example shown in Figure 4.4(b), we have $I(\beta) = \max(w4, w5, w6, w7, w8)$ and $D(\alpha, \beta) = \min(w1, w2, w3)$.

Finally, we define the minimum inter-cluster difference, MI :

$$MI(\alpha, \beta) = \min(I(\alpha) + \tau(\alpha), I(\beta) + \tau(\beta)) \quad (4.4)$$

where τ is a normalization factor which is used to reduce the effect of cluster size. It is

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

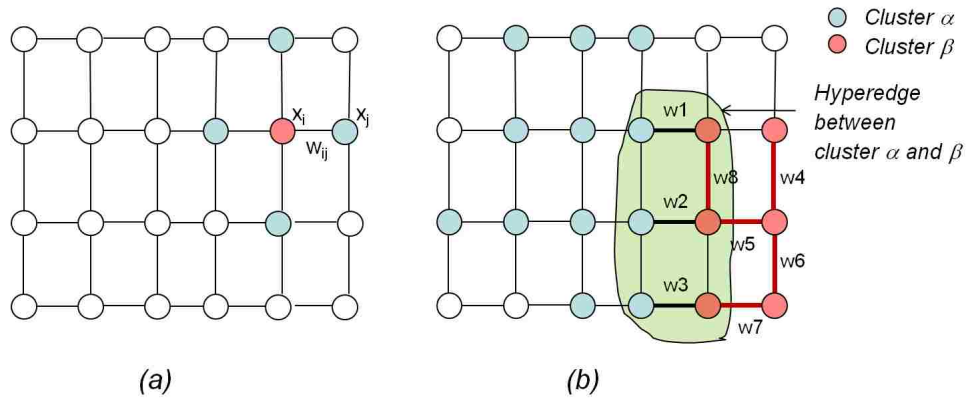


Figure 4.4: (a) 4-connected grid graph; (b) building level-0 hypergraph

defined as

$$\tau(\alpha) = k/|\alpha| \quad (4.5)$$

where k is a constant which controls the scale. The larger k causes a preference for larger components.

We now compare the inter-cluster difference $D(\alpha, \beta)$ with the minimum intra-cluster difference $MI(\alpha, \beta)$. If $D(\alpha, \beta) < MI(\alpha, \beta)$, we group them. Otherwise, we keep them separated.

In order to build hierarchical hypergraphs, we set the upper limit on how many superpixels can be merged on each level. The higher the level is, the more superpixels can be grouped into one cluster. In practice, the limit is 2^i for level i . Figure 4.5 shows level-1, level-2 and level-3 hypergraphs for the image in Figure 4.1 (it shows the level-0 hypergraph).

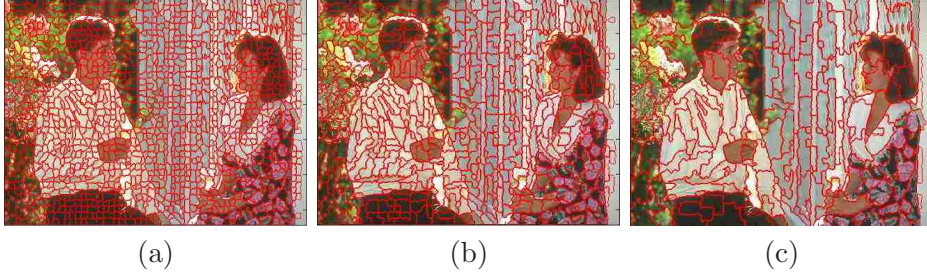


Figure 4.5: (a) Level-1 hypergraph; (b) Level-2 hypergraph; (c) Level-3 hypergraph.

4.3.2 Label Propagation on the Hierarchical Hypergraph

In the second step, we model the label variable for each pixel as a discrete or Gaussian variable. For these models, we generate labels based on discrete or Gaussian belief propagation, respectively.

Interactive Image Segmentation and MRF Construction

The task of interactive image segmentation is to assign unique labels to the unlabeled pixels, using labeled pixels to guide the process. This problem can be formulated in terms of energy minimization on a Markov Random Field (MRF), where each pixel is treated as a node and each node is considered to be connected to its spatial neighbors. In this formulation, the random variables are the labels of each pixel. Let $\mathcal{Y} = \{Y_i\}_{i=1}^N$ be the collection of the label variables. Then the joint event $\mathbf{y} = \{Y_1 = y_1, \dots, Y_N = y_N\}$ is called a configuration of \mathcal{Y} . Each configuration \mathbf{y} is associated with a posterior probability of the MRF $P(\mathbf{y})$. By the Hammersley-Clifford theorem,

$$P(\mathbf{y}) = \frac{1}{Z} \prod_i \exp(-\phi_i(y_i)) \prod_{i,j} \exp(-\psi_{ij}(y_i, y_j)), \quad (4.6)$$

where Z is a normalization constant, $\exp(-\phi_i(y_i))$ is the likelihood of pixel i having a label y_i and $\exp(-\psi_{ij}(y_i, y_j))$ is the prior for two neighboring pixels having labels y_i and y_j . The goal of MRF learning is to find the configuration $\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y}) =$

$\arg \min_{\mathbf{y}} E(\mathbf{y})$, where

$$E(\mathbf{y}) = \sum_i \phi_i(y_i) + \sum_{i,j} \psi_{ij}(y_i, y_j), \quad (4.7)$$

In the above equation of energy E , $\psi_{ij}(y_i, y_j)$ is the cost of assigning y_i and y_j to the neighbors i and j ; we assume that labels should be piecewise smooth. $\phi_i(y_i)$ is the likelihood cost since we assume the labeling should also fit to the learned models. So for the interactive image segmentation problem, we should define the likelihood cost as

$$\phi_i(y_i) = \begin{cases} \infty & y_i = l_i, i \in \mathcal{X}_L \\ 0 & otherwise \end{cases} \quad (4.8)$$

where l_i is the predefined label of pixel i , and \mathcal{X}_L is the labeled pixel set which is assigned by users in the interactive segmentation methods.

Eq. 4.25 is an optimization problem which may be NP hard. We can relax it as a probabilistic inference problem and use the belief propagation (BP) approach to solve it. For a graph without cycles (a chain or tree graph), BP computes the MAP configuration efficiently by updating a measure of the marginal distribution at each node, conditioned on information from its neighbors (passing messages from one node to another). The marginal probability is termed “belief”. The belief propagation can be run in an iterative way. At each iteration, each node not only receives a message from its neighbors, but sends a message to each of them.

Discrete Label Propagation

In discrete spaces, messages and compatibility functions can be expressed as arrays. There are two different message update rules: the sum-product rule and the max-product rule. In our research, we use the max-product rule [111]. Figure 4.6 illustrates

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

an example about how the belief propagation scheme works on an example simple graph.

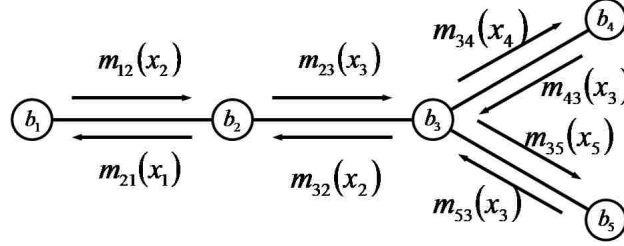


Figure 4.6: basic belief propagation

We denote by $m_{i \rightarrow j}^t$ the message that node i sends to node j at time t . The max-product rule of the message update is as follows:

$$m_{i \rightarrow j}^t(y_j) \leftarrow \kappa \max_{l_i} \phi_i(y_i) \psi_{ij}(y_i, y_j) \prod_{s \in N(i) \setminus j} m_{s \rightarrow i}^{t-1}(y_i) \quad (4.9)$$

After T iterations, the belief defined on node i can be calculated by

$$b_i(y_i) = \kappa \phi_{y_i} \prod_{s \in N(i) \setminus j} m_{s \rightarrow i}^T(y_i) \quad (4.10)$$

With negative log on each side of the above two equations, we can get the equivalent computations as follows:

$$m_{i \rightarrow j}^t(y_j) \leftarrow \min_{l_i} (\phi_i(y_i) + \psi_{ij}(y_i, y_j) + \sum_{s \in N(i) \setminus j} m_{s \rightarrow i}^{t-1}(y_i)) \quad (4.11)$$

$$b_i(y_i) = \phi_{y_i} + \sum_{s \in N(i) \setminus j} m_{s \rightarrow i}^T(y_i) \quad (4.12)$$

After T iterations, the pixel i is assigned the label y_i , so that $b_i(y_i)$ is minimum.

Now we define $\phi_i(y_i)$ for each pixel at each level. To give a simple example, we use

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

two labels, that is $Y = \{O, B\}$ or $Y = \{1, -1\}$, where O is object and B is background.

For pixels predefined as object,

$$\phi_i(y_i) = \begin{cases} 0 & y_i = O \\ \infty & y_i = B \end{cases} \quad (4.13)$$

For pixels predefined as background,

$$\phi_i(y_i) = \begin{cases} \infty & y_i = O \\ 0 & y_i = B \end{cases} \quad (4.14)$$

In our implementation, we replace ∞ with a finite value such as 2.

We also need to define $\phi_i(y_i)$ for cluster α at level l since we use clusters to represent pixels at the higher levels. Our method is as follows. If the cluster α only includes object labels, then use Eq. 4.13. If the cluster α only includes background labels, then use Eq. 4.24. If the cluster α includes both labels, let N_L denote the total number of labeled pixels; N_O , the number of pixels pre-labelled as object; and N_B , the number of pixels pre-labelled as background. We then use this assignment rule:

$$\phi_\alpha(y_\alpha) = \begin{cases} \frac{N_B}{N_L} & y_i = O \\ \frac{N_O}{N_L} & y_i = B \end{cases} \quad (4.15)$$

If the cluster α doesn't include any labels, then

$$\phi_\alpha(y_\alpha) = \begin{cases} 0 & y_i = O \\ 0 & y_i = B \end{cases} \quad (4.16)$$

The compatibility function $\psi_{ij}(y_i, y_j)$ for pixel i and j is defined as

$$\psi_{ij}(y_i, y_j) = \begin{cases} 0 & y_i = y_j \\ w_{ij} & otherwise \end{cases} \quad (4.17)$$

The compatibility function $\psi_{\alpha\beta}(y_\alpha, y_\beta)$ for cluster α and cluster β is defined as

$$\psi_{\alpha\beta}(y_\alpha, y_\beta) = \begin{cases} 0 & y_\alpha = y_\beta \\ W_{\alpha\beta} = \frac{1}{Z} \sum_{i \in \alpha, j \in \beta} w_{ij} & otherwise \end{cases} \quad (4.18)$$

where Z is a normalization constant. The initial messages are set to 0.

One of the issues with discrete label propagation is that we need to calculate an array of messages for all possible labels. To avoid it, we can model label variables as random continuous variables then use different thresholds to assign different labels.

Gaussian Label Propagation

Gaussian Belief Propagation (GaBP) is a special case of continuous BP, where the underlying label variable has a Gaussian distribution. The algorithm is based on the paper by Weiss et al. [110]. Now the messages $m_{ij} = N(u_{ij}, P_{ij})$ and the likelihood energy $\phi_i = N(u_0, P_0)$ are both Gaussian. We define the compatibility function ψ_{ij} for pixel i and j as

$$\psi_{ij} = \begin{pmatrix} a_{ij} & b_{ij} \\ b_{ij}^T & c_{ij} \end{pmatrix} = \begin{pmatrix} w_{ij} & 1 - w_{ij} \\ 1 - w_{ij} & w_{ij} \end{pmatrix} \quad (4.19)$$

We define the message $m_{ij} = N(u_{ij}, P_{ij})$ from node i to j as

$$P_{ij} \leftarrow c_{ij} - b_{ij}^T (a_{ij} + P_0)^{-1} b_{ij}, u_{ij} \leftarrow P_{ij} b_{ij}^T (a_{ij} + P_0)^{-1} P_0 u_0 \quad (4.20)$$

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

If the pixel i is prelabeled, then $\psi_i = N(u_0, P_0)$ is defined as

$$P_0 = 0, u_0 = +1, \text{ or, } -1 \quad (4.21)$$

If the pixel is unlabeled without labeled neighbors, then

$$P_0 = \sum_{s \in N(i) \setminus j} P_{si}, u_0 = P_0^{-1} \sum_{s \in N(i) \setminus j} P_{si} u_{si} \quad (4.22)$$

If the pixel is unlabeled with labeled neighbor k , then

$$P_0 = a_{ik} + \sum_{s \in N(i) \setminus j} P_{si}, u_0 = P_0^{-1} (-b_{ik} u_{ik} + \sum_{s \in N(i) \setminus j} P_{si} u_{si}) \quad (4.23)$$

For clusters α and β , the compatibility function $\psi_{\alpha\beta}$ is similar to Eq. 4.19 using $W_{\alpha\beta}$ instead. The likelihood energy ϕ_α can be defined similarly by replacing i with α and j with β .

When the message converges, the belief for pixel i can be obtained as

$$P_i \leftarrow \sum_{j \in N(i)} P_{ji}, u_i \leftarrow P_i^{-1} \left(\sum_{j \in N(i)} P_{ji} u_{ji} \right) \quad (4.24)$$

To assign two labels O or B , we set the threshold 0 for u_i .

After messages converge at a higher level hypergraph, we propagate the messages down to the lower level hypergraph. The rule is the messages of the higher level cluster will equally deliver to its children in the lower level cluster. Then the messages can be propagated in a coarse-to-fine manner. In this way, we can speed up the label propagation convergence because (1) these messages are already initialized close to the fixed points by running label propagation at one level of resolution to get an estimation of the messages of the next finer resolution; (2) by our grouping predicate, the similar

4.3 New Framework: Label Propagation on Hierarchical Hypergraphs

superpixels are merged into bigger clusters level by level. These pixel inside clusters has similar features so that these pixels on each level intend to have similar messages which is reflected by our propagating rule between levels.

Initialization in Belief Propagation

Sometimes in one image, the object is made of several separated parts. Figure 4.7 (a) shows a figure which has two separated leaves. In order to segment two leaves, we need two separate foreground scribbles. To simplify the user interaction, we can modify our method in the initialization step so that only one foreground scribble on one leaf is needed.

We calculate the mean colors of the clusters for each level. Even if the unscribbled clusters are far way from the scribbled clusters but they are similar in the mean colors, thus we assign them the label of the closest scribbled clusters in ϕ . Such initialization will make Belief Propagation more efficient since the ϕ of these clusters are closer to their final ϕ .

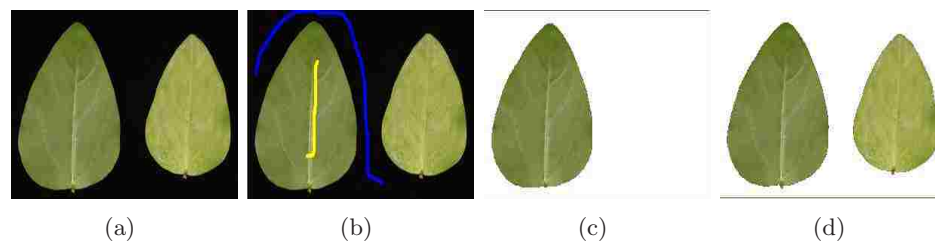


Figure 4.7: (a)A figure of two leaves; (b)Original image with scribbles; (c)Segmentation result with the original initialization;(d)Segmentation result with the modified initialization.

4.3.3 Overview of the Segmentation Framework

The overview of our framework is illustrated in Figure 4.9. In the first step, we construct a hierarchical hypergraph using the grouping predicate. This is a bottom up procedure. In the second step, we propagate labels on each level's hypergraph iteratively by DBP or GaBP. Initial messages on lower level hypergraphs are obtained from the higher level hypergraphs. We have found that with this multilevel approach it is sufficient to run label propagation for a small number of iterations at each level.

4.4 New Interaction for Topology Change: Connecting Dots

Inspired by [108], we develop a new interaction for topology change based on our framework, which is especially useful for solving the task of segmenting challenging objects with very thin, elongated parts. Our method, however, is different from [108], which merges the Dijkstra algorithm and graph cut. In our framework, users can first put two connecting dots on the two objects respectively. We then use fast marching method to find the center line which connects two points. We set the points on the center line as foreground then the described label propagation method is applied to propagate the labels.

We follow the fast marching method described in [27] to extract the center line (shortest path) between the starting point p_0 and the ending point p_1 . In this section, we present the basic idea to find the global minimum of the active contour energy using minimal paths (see details in [27]). The energy to minimize is similar to classical deformable models where it combines terms for smoothing and for attraction to image

features (Potential):

$$E(C) = \int_{\Omega} w_1 \|C'(s)\|^2 + w_2 \|C''(s)\|^2 + P(C(s)) ds \quad (4.25)$$

where $C(s)$ represents a curve drawn on a 2D image, Ω is its domain of definition $[0, L]$, and L is the length of the curve. A simplified energy model without a second derivative term leads to the expression

$$E(C) = \int_{\Omega} w + P(C(s)) ds \quad (4.26)$$

We now have an expression in which the internal forces are included in the external potential. The regularization is now achieved by the constant $w > 0$.

Given a potential $P > 0$ that takes smaller values near desired features, we look for paths along which the integral of $\tilde{P} = P + w$ is minimal. We can define the surface of minimal action U , as the minimal energy integrated along a path between a starting point p_0 and any point p :

$$U(p) = \inf_{A_{p_0,p}} E(C) = \inf_{A_{p_0,p}} \int_{\Omega} \tilde{P}(C(s)) ds \quad (4.27)$$

where $A_{p_0,p}$ is the set of all paths between p_0 and p . The minimal path between p_0 and the ending point p_1 in the image can be easily deduced from this action map. Assuming that potential P is always positive, the action map will have only one local minimum which is the starting point, and the minimal path will be found by a simple back propagation on the energy map.

To calculate this map, a front-propagation equation related to equation (4.27) is solved:

$$\frac{\partial C}{\partial t} = \frac{1}{\tilde{P}} \vec{n} \quad (4.28)$$

It evolves a front starting from an infinitesimal circle shape around until each point inside the image domain is assigned a value for U . The value of $U(p)$ is the time at which the front passes over the point p . Then it finds the shortest path energy to reach the start point from any point in the image. The fast marching technique, introduced by Sethian [94], was used by Cohen and Kimmel; in [27] the authors note that the map U satisfies the Eikonal equation:

$$\|\nabla U\| = \tilde{P} \quad (4.29)$$

After calculating the minimal action map U using the fast marching method. The gradient of U is orthogonal to the propagating fronts since these are its level sets. Therefore, the minimal action path between any point p and the start point p_0 is found by sliding back the map U until it converges to p_0 . This can be done with a simple steepest gradient descent, with a predefined descent step, on the minimal action map U , choosing $p_{n+1} = p_n - step \times \nabla U p_n$.

Figure 4.8(a-d) shows the process of extracting the center line of two points. Figure 4.8(e-h) shows how we use the connecting dots to obtain better segmentation result by connecting very thin, elongated parts (legs of the crane in our case).

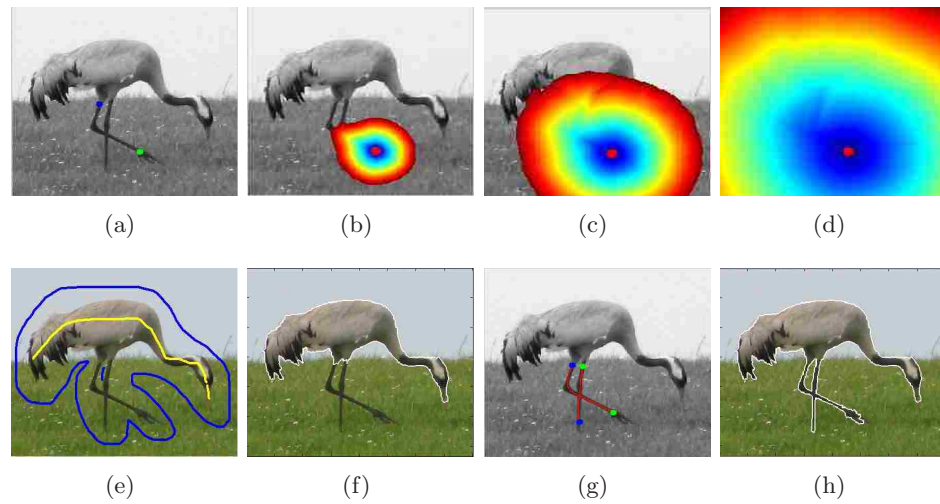


Figure 4.8: (a) Connection dots (Green dot is the starting point and Blue dot is the ending point); (b)-(d) Front-propagation by fast marching; (e) Original Image and scribble; (f) The initial results; (g) Connection dots; (h) Final result.

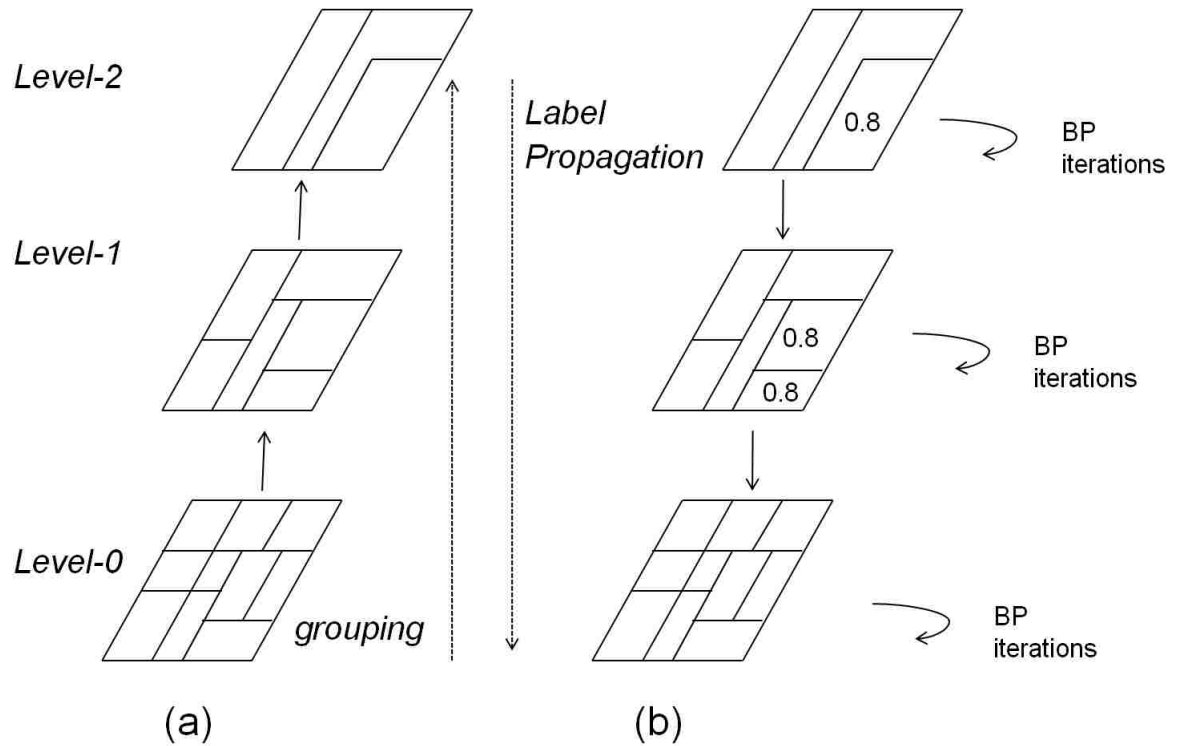


Figure 4.9: (a)Construct hierarchical hypergraphs; (b)Label propagation on each level and between levels

4.5 Scribble Recommendation

Inspired by the idea in [7] of automatic recommendation system that intelligently recommends where the user should scribble next, we integrated such a capability into our framework. This is actually a by-product of our algorithm. In our framework, the outputs of our methods (for example, DBP) are two beliefs for each pixel. One is the belief assigning the pixel as a background label. The other is the belief assigning the pixel a foreground label. We finally assign the pixel with the label where its belief is minimum. Note that the belief we minimize is in a sense the opposite of the intuitive belief that we are maximizing. This results from applying the negative log operator to our optimization equations in Equation 4.11 and 4.12. So the belief can be seen as a cost when assigned as a background or foreground pixel. The closer the two costs are, the more difficult it is for the algorithm to judge if it should be assigned as background or foreground. We can use a map of the difference of these two costs to measure the degree of difficulty in assigning a label. We call this a confusion map. Significantly, the areas in the map which correspond the great “confusion” are the areas where we need user iterations.

The confusion value of a pixel i in the confusion map can be calculated by the equation: $abs(belief(foreground) - belief(background))$. We then search a window (50 pixel by 50 pixel) in the image after the initial segmentation such that the total confusion value of pixels inside are minimum. Figure 4.11(c) shows one example of the confusion map while the original image with scribbles shown in Figure 4.11(a) and the segmentation result is shown in Figure 4.11(b). The darker in the confusion map, the more confusing for the algorithm. From the confusion map, we can recommend the area for the user’s next scribbles (red rectangle in Figure 4.11 (c)).

So, in our framework, we keep the users “in the loop”, that is, users can add

more scribbles and refine the segmentation results in each iteration. With our scribble recommendation system, users can easily put scribbles within a window recommended by our algorithm. One scenario is illustrated in Figure 4.10.

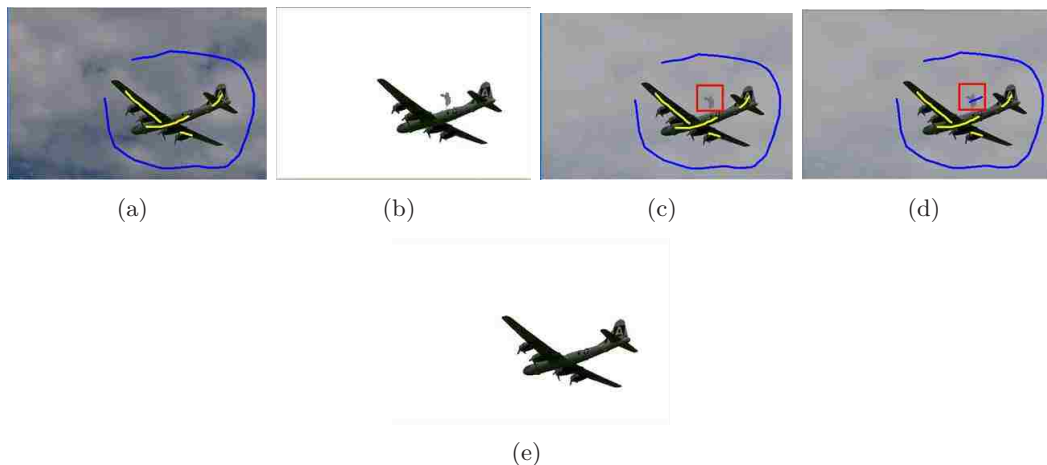


Figure 4.10: (a)The original image with scribbles (b) The initial segmentation (c) Recommendation area is shown in the red window (d)Draw scribbles inside the recommendation window (e) The result in the second iteration

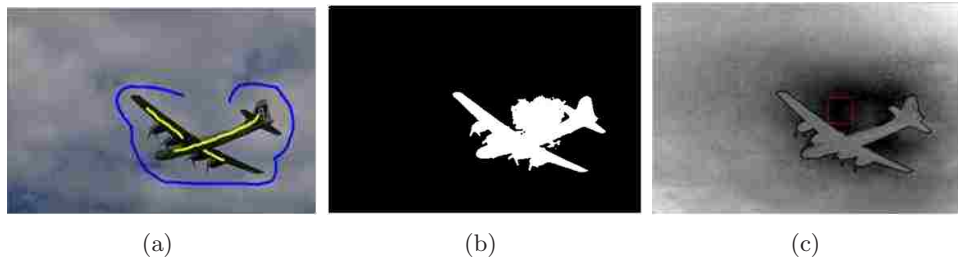
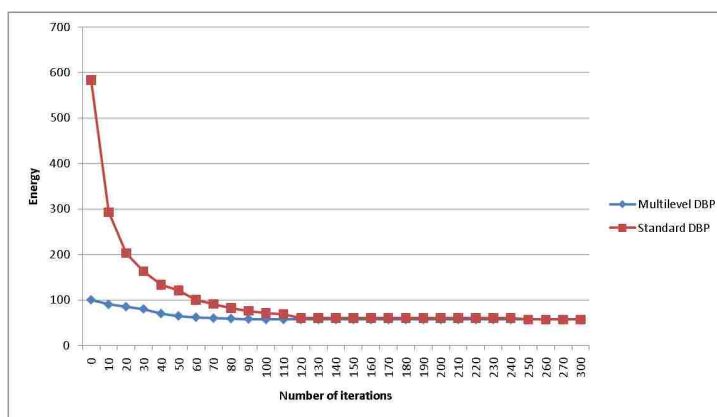
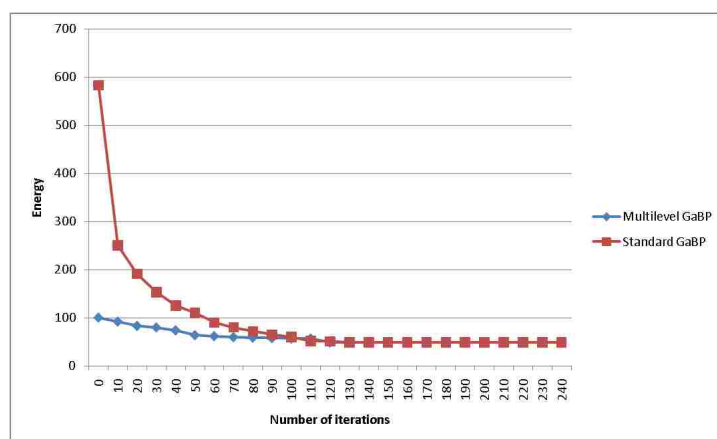


Figure 4.11: (a)The original image with scribbles (b) Segmentation result (c) The confusion map with recommendation scribble area



(a)



(b)

Figure 4.12: Convergences of (a) multilevel DBP comparing with standard DBP (b) multilevel GaBP comparing with standard GaBP

4.6 Experimental Results

4.6.1 Evaluation of Proposed Framework

In this section, we evaluate the proposed segmentation framework with respect to convergence and “sensitivity to scribbles”. By sensitivity to scribbles, we mean the sensitivity of the segmentation results to variations in scribble placement and length. As mentioned before, belief propagation can be seen as a method of computing a fast solu-

tion for minimizing the energy of an MRF. Figure 4.12 illustrates the energy minimizing as the iterations proceed. The energy is calculated by Eq. 4.25 for both standard and multilevel versions. As we can see, the multilevel method computes a low energy solution in significantly fewer iterations per level, as compared to the standard algorithm in both DBP and GaBP cases.

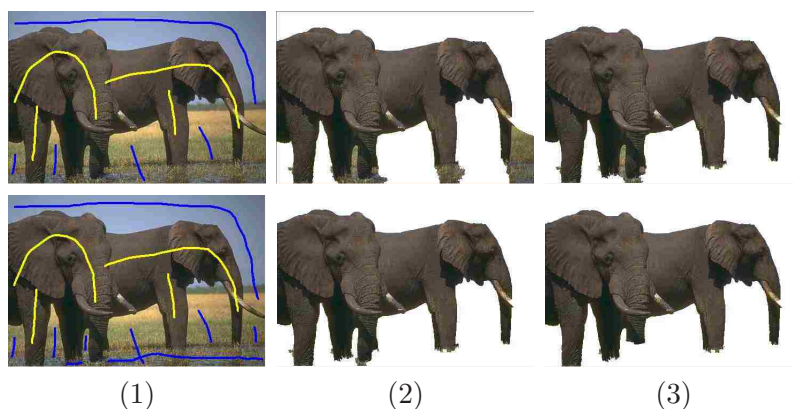


Figure 4.13: (1)Original image and scribbles; (2)Result of our method in DBP; (3) Result of our method in GaBP

Sensitivity to scribbles is illustrated visually in Figure 4.13. We can see that the results of segmentations are dependent on the quality of scribbles in both DBP and GaBP cases. Generally with more scribbles we can achieve more satisfactory results. We also used random machine-generated scribbles (see details in Section 4.6.5) to evaluate the segmentation accuracy [44] quantitatively for DBP. We increased the length of both background and foreground scribbles in a fixed-size (=30 pixels) step and changed the positions of background and foreground scribbles randomly in each step. Figure 4.14 shows how the length and different positions can effect the segmentation accuracy. We will discuss the problem about how to evaluate segmentation as a function of variations in scribble position in Chapter 5.

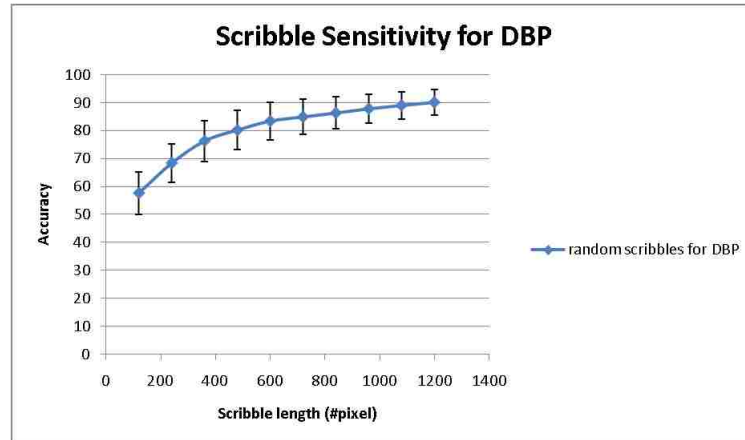


Figure 4.14: Accuracy affected by random machine-generated scribbles in different length and different positions

4.6.2 Comparison with Other Methods

In this section, we evaluate our segmentation results by comparing with random walks, grow cut, grab cut and graph cut with respect to speed, visual verification and quantitative verification. We used the Berkeley nature image segmentation dataset [67] for visual evaluation; this is distributed with a manually segmented ground truth. For quantitative evaluation, we used the Microsoft GrabCut dataset [89].

For all the experiments, we used $\sigma = 600$ in Eq. 4.1 and $k = 500$ in Eq. 4.5. In all cases we built four-level hypergraph.

Speed

We compared the running time of our framework with the standard random walker, grow cut and graph cut methods on segmenting the images of different sizes. For each size, we used 5 images and then calculated the average time. The times reported were obtained on a 3.0 Ghz Intel Core 2 Duo computer with 3.25 GB RAM. All of the algorithms were implemented in Matlab 9.0. For a fair comparison, we did not use

superpixels in our DBP and GaBP cases. The results are shown in Figure 4.15, from which we can see that our method performs competitively for the smaller images and significantly faster for the largest image tested.

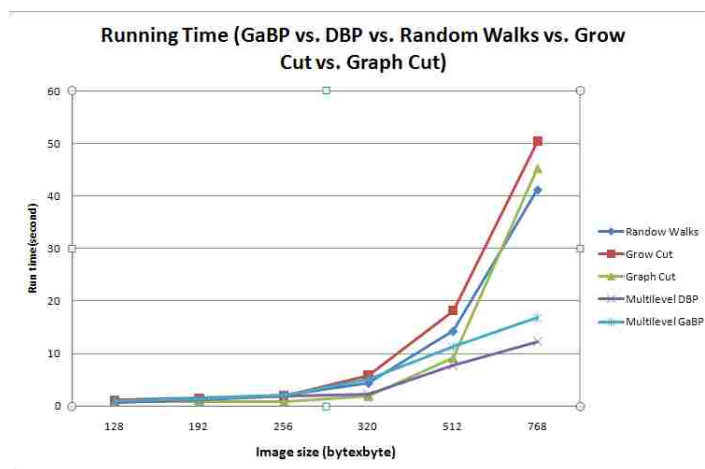


Figure 4.15: Running time comparison between our framework, random walks, grow cut and graph cut

4.6.3 Visual Verification

Representative segmentation results of our results compared with random walks and grow cut are shown in Figure 4.16. Figure 4.16 shows our algorithms (DBP and GaBP) both generate more details at the object boundaries. Compared with DBP and GaBP, GaBP appears better than DBP; this is possibly due to the more complex data modeling in GaBP. The trade-off, though, is that GaBP is less efficient than DBP.

4.6.4 Quantitative Verification

The Microsoft Grab Cut dataset provides seed regions that facilitated our experiments. Our average error rate was 7.2% in GaBP and 7.6% in DBP. We compare these to the performance of grab cut [14], grow cut, random walks and learning on hypergraph[31]

Table 4.4: Error rates for Grab Cut dataset

Methods	Error rates
DBP	7.6%
GaBP	7.2%
Grab Cut	7.9%
Grow Cut	8.3%
Random Walks	7.7%
Learning on Hypergraphs	7.3%

in Table 4.4. The error rates for all images for our algorithms are shown in Figure 4.17 and Figure 4.18.

4.6.5 Evaluation of Scribble Recommendation System

We also conducted evaluation of our scribble recommendation system using machine experiments without interactions from human beings. To mimic human scribbles, we developed a mechanism to generate automatic scribbles. We modeled these synthetic scribbles as (smooth) random walks. Both synthetic background and foreground scribbles were positioned in the background and foreground regions, respectively, in the ground truth. Our scribble generation technique consists of begins with randomly sampling a starting point in the image. A direction angle is then randomly sampled such that it is highly correlated (for smoothness) with the previous direction angle for the scribble, and a fixed-size (30 pixels) step is taken along this direction to extend the scribble (as long as it does not cross object boundaries, as indicated by the ground truth segmentation of the image). To mimic user scribbles given a recommendation box, the initial as well as subsequent points on the scribble are picked by considering the recommendation box to be a distribution. Using synthetic scribbles allows us to control the length of scribbles and observe the behavior of the algorithm with increasing information being fed to it.

We start by generating a foreground and background scribble on a random image

in a group of test images. We then compute the recommendation box according to our algorithm. Then we generate the next synthetic scribbles as guided by this recommendation box. We repeat this until we have scribbled about 1000 pixels across the group, and compute the average segmentation accuracy of the test images. For comparison purposes, we compute one baseline. That is, we use a uniform recommendation map on all test images, which essentially means randomly scribbling on the images.

Figure 4.19 shows the performance of our recommendation system with increasing scribble length, as compared to the baseline. We see that our proposed recommendation system does in fact provide meaningful guidance for regions to be scribbled on next (as compared to the baseline). A meaningful upper bound would be the segmentation accuracy that could be achieved if an oracle told us where the segmentations were incorrect, and subsequent scribbles were provided only in these erroneous regions. As seen in Figure 4.19, our recommendation system performs very close to this upper bound.

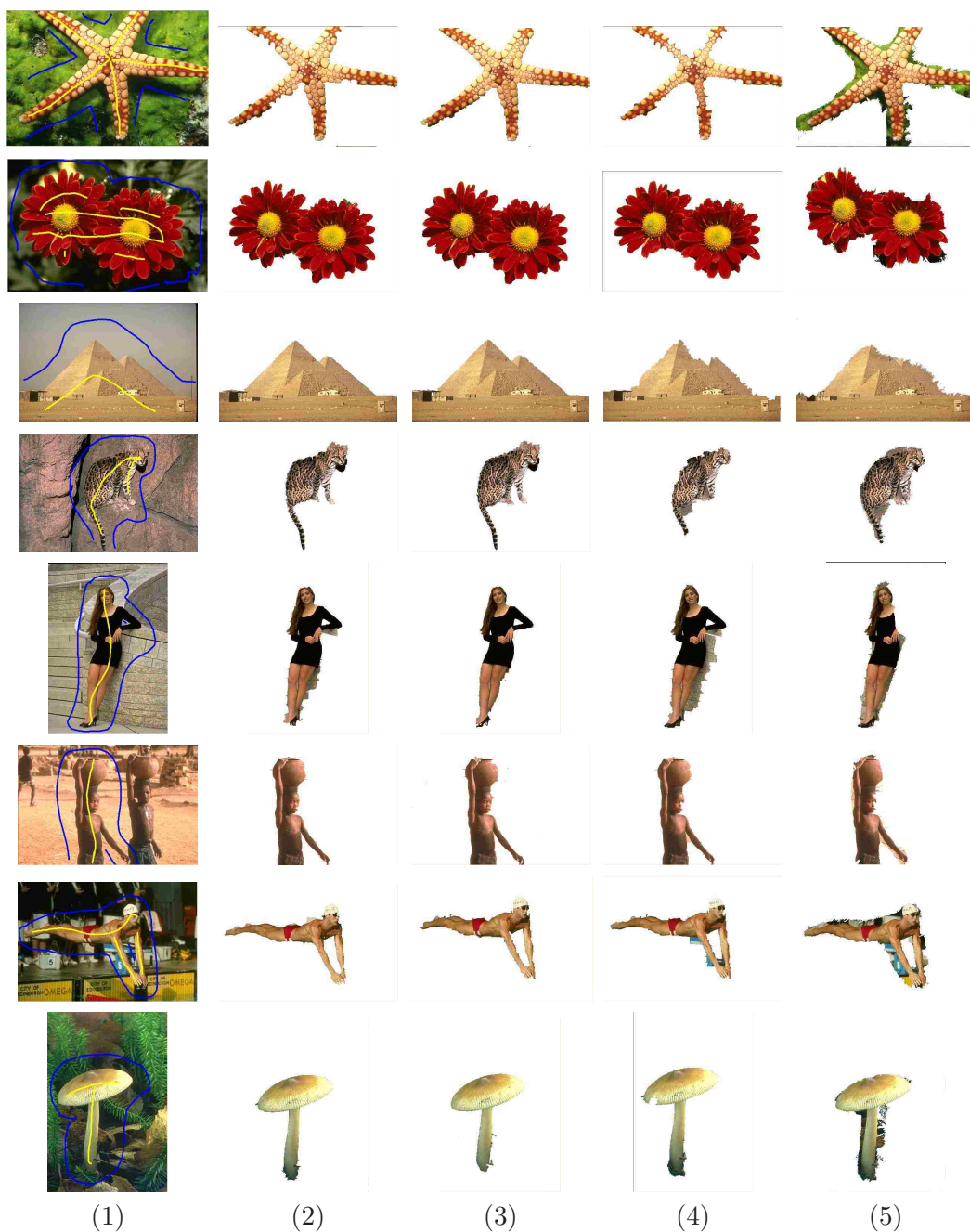


Figure 4.16: (1)Original image and scribbles; (2)Result of our method in DBP; (3) Result of our method in GaBP; (4)Result of random walks; (5)Result of grow cut

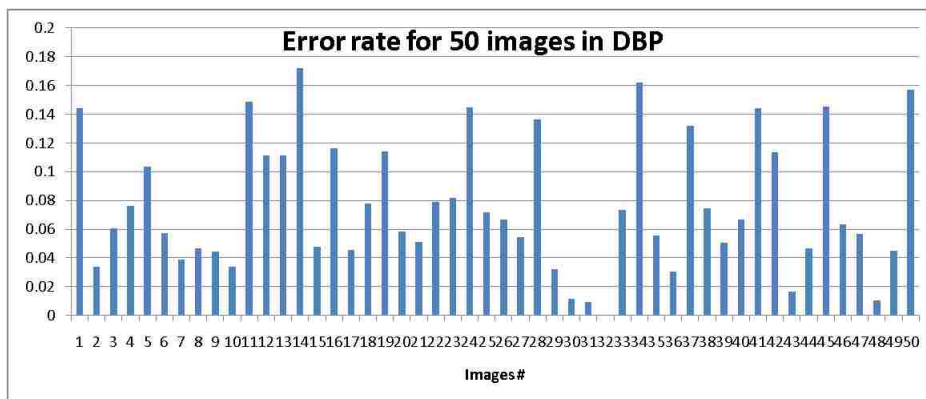


Figure 4.17: Error rates on 50 images in [89] for DBP

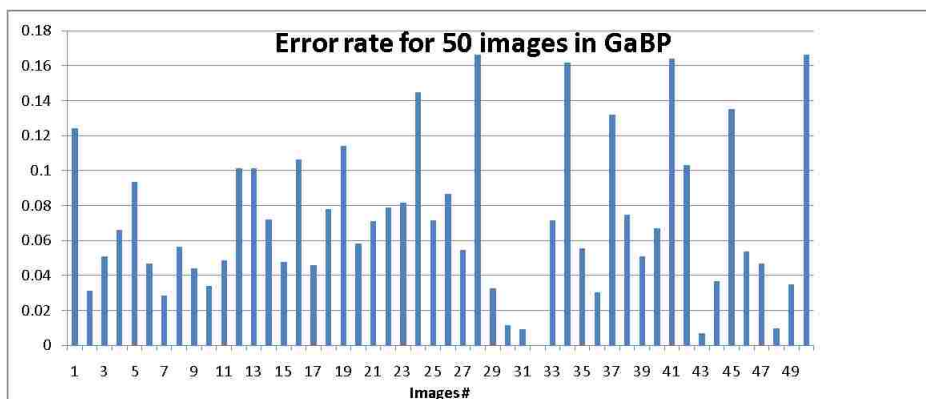


Figure 4.18: Error rates on 50 images in [89] for GaBP

4.7 Discussion

In this research, we have presented an efficient label propagation algorithm based on hierarchical hypergraphs for interactive image segmentation. Our method yields results of comparable accuracy to other algorithms and adds the advantage of speed. The experiments on natural image segmentation are provided in both quality and quantity to show the effectiveness of our method. Also from our framework, we add two new features: connection dots and scribble recommendation, which can assist users in obtaining more accurate segmentation. Our framework is flexible that we can add more user interactions in the future. For example, instead of using scribbles, users can draw a rectangle around the erroneous segmentation regions while the model can only refine segmentation results in this area thus to improve the speed of the model. Another improvement is to model user intention into the framework. Some aspects of user intention can be utilized. For example, the relative position between the scribble and the previous result indicates the segmentation error. The framework can work on these areas to fix the segmentation errors thus the framework may require fewer iterations.

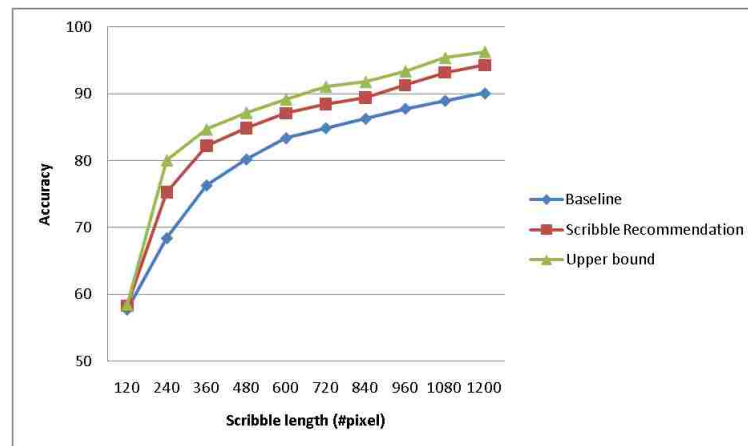


Figure 4.19: Machine Experiments: comparing scribble recommendation system with nature upper bound and baseline

Chapter 5

Evaluation of Interactive Segmentation Methods

5.1 Introduction

This chapter aims to develop a framework for evaluating of interactive segmentation algorithms that combine user studies and automatic evaluation with machine-generated scribbles. In order to compare popular interactive segmentation methods with our methods and to study what factors are important for better user experience, we focus on the issue of elimination of fluctuation in the results due to variations in the scribbles; reducing the number of required iterations; and reducing the number of required scribbles in an iterations. We seek to verify our ideas about how to guide users in adding scribbles so that more accurate and faster results can be achieved. Toward these goals we propose several quantitative measures to quantify the user experience. To implement measures, we develop a method for automatic evaluation which replaces the human operator with an algorithmic process designed to emulate human behavior

as closely as possible. To achieve this, we propose generating scribbles by random walks and driving the segmentation by automatically generating scribbles based on the current segmentation error and the ground truth data. This chapter explores two different strategies for generating these interactions. One is initializing the scribbles in random positions and iteratively adding segments to a scribble which extend its length and path. The other strategy is initializing the scribbles in the center of the largest erroneous region, and then iterating to extend the length and path. Finally, we evaluate several algorithms using each of these automatic strategies, and compare the results with the user studies.

The remainder of this chapter is organized as follows. Section 5.2 reviews the evaluation data set, criteria and methods. Section 5.3, 5.4 and 5.5 present our evaluation framework. Evaluation results are presented in Section 5.6. Section 5.7 provides the summary.

5.2 Literature Review

In this section, we review current evaluation methods. We identify three components for evaluating of interactive segmentations: (1) quality measures for segmentation results and user experience; (2) data sets on which the interactive methods will be tested; (3) methods used to evaluate the algorithms.

5.2.1 Quality Measures

The most commonly used criteria for interactive segmentation methods are as follows [80, 102]:

1. Accuracy: the degree to which the boundary of the object matches the ground

truth;

2. Efficiency: the amount of time or effort required to obtain the desired segmentation;
3. Repeatability or stability: the extent to which the same segmentation would be obtained in different times when the user has the same intention.

Accuracy

Accuracy measures how well a segmentation matches the object in ground truth. It can be classified into two categories: object accuracy and boundary accuracy. The object accuracy benchmark measures how well the entire region matches the ground truth while the boundary accuracy benchmark measures how well the boundary of the segmented object corresponds to the ground truth object boundary.

Let \mathbf{v} be any pixel inside the ground truth object and the coordinates of \mathbf{v} are in \mathbb{Z}^2 , and $G_O = \{\mathbf{v}\}$ be the set of all of these pixels. Similarly, define M_O to be the set of all pixels in the machine-segmented object. Let G_B and M_B denote the complements of these sets. Denote “true positive” pixels as TP , “false positive” pixels as FP , “true negative” pixels as TN , and “false negative” pixels as FN . We can assign these categories by comparing the machine-segmented object to the ground truth object.

The commonly used object accuracy measures are as follows:

1. Sensitivity p and specificity q :

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (5.1)$$

$$\text{specificity} = \frac{TN}{TN + FP} \quad (5.2)$$

Sensitivity is the fraction of pixels correctly included in the segmentation result out of all pixels in the segmentation result, while specificity is the fraction of pixels correctly excluded from the segmentation result out of all pixels outside of the ground truth.

2. Jaccard index: this is used to measure the overlap between the machine segmented object and the ground truth object:

$$J = \frac{|G_O \cap M_O|}{|G_O \cup M_O|} = \frac{TP}{TP + FP + FN} \quad (5.3)$$

We also refer to this index as “accuracy rate”. This index has been used by various authors for comparing segmentations [45, 52, 102]. The complement of accuracy rate is error rate, which is defined as $1 - J$.

3. Dice coefficient: this gives the “similarity level” of segmentation to ground truth. The Dice coefficient is defined as

$$DCS = \frac{2|G_O \cap M_O|}{|G_O| + |M_O|} = \frac{2TP}{2TP + FP + FN} \quad (5.4)$$

Other region accuracy measures include Hamming distance, Rand index and consistency error as described in [25], and normalized measures in [73].

To compute the boundary accuracy, [70] defines two sets: B_G and B_M as the

internal border pixels for the ground-truth object and for the machine-segmentation, respectively. Let $N_{\mathbf{x}}$ be the standard set of eight-neighbors of any $\mathbf{x} \in \mathbb{Z}^2$, they are defined as:

$$B_G = \{\mathbf{x} : \mathbf{x} \in G_O \wedge N_{\mathbf{x}} \cap G_B \neq \emptyset\} \quad (5.5)$$

$$B_M = \{\mathbf{x} : \mathbf{x} \in M_O \wedge N_{\mathbf{x}} \cap M_B \neq \emptyset\} \quad (5.6)$$

Similarly, boundary accuracy B_A is defined by substituting the sets B_G and B_M in place of G_O and M_O in Equation 5.3:

$$B_A = \frac{|B_G \cap B_M|}{|B_G \cup B_M|} \quad (5.7)$$

Unfortunately, as noted in [70], this measure is extremely sensitive to small errors near the boundary of the object. Generally, humans are less sensitive to small errors near the object boundary than to large holes in the object. For the case when there are many small errors near the boundary, the values given by Equation 5.7 will almost always be excessively low. To handle this, [70] redefines boundary accuracy by introducing an error tolerance near the border pixels. Another reported measure is called the “cut discrepancy” [73], which measures the distances between cuts for evaluating automatic segmentations

Both object accuracy and boundary accuracy measures generate values within the range $[0, 1]$, where larger values imply more accurate segmentation results.

Efficiency

Efficiency is commonly measured as computation time. This can be the total computation time during iterations or the computation time in each step.

Repeatability

In [70], repeatability is measured implicitly in user studies where multiple segmentations for the same object are obtained by different users working on the same image with the same segmentation algorithm. To our understanding, this repeatability can also be measured as robustness to variation of scribbles length and placements. In this chapter, we will measure this quality explicitly through automatic evaluation, which is discussed in Section 5.3.

5.2.2 Data Sets

In order to measure accuracy, efficiency and repeatability, we need data set and ground truth for evaluation of interactive segmentation methods. We use two widely available data sets. One is the GrabCut dataset [3] with known ground truth segmentations and scribbles which are called static trimaps. The other is the Berkeley segmentation dataset [68], which contain images with manually-segmented ground truth.

5.2.3 Evaluation Methods

There are three commonly used methods for evaluating of interactive segmentation methods.

Static User Interaction

In this method, researchers draw scribbles by themselves and create image trimaps. Then these fixed set of user scribbles are used as inputs to evaluate interactive segmentation methods [12, 15] and quality measures are computed between the segmentation and ground truth. As remarked above, the GrabCut data set [3] is distributed with such trimaps. One problem of this method is that it does not consider user interactions across iterations. The evaluation can only be done for one iteration only. But different users may react very differently to segmentation results produced during successive iterations. The static user interaction model can not simulate user interactions during the iteration process. Also some potentially desirable features for interactive segmentation methods can not be measured in this static model. For example, some iterative algorithms can accelerate the updating speed for computing the current solution, given the current inputs by reusing information from the previous solution. As another example, some algorithms can be more receptive to user interactions which means they can improve results quickly after certain iteration even after starting with worse segmentations in the first iteration.

User Studies

A user study involves a group of participants to finish the required segmentation tasks. They may also be asked to fill out a questionnaire after completing the tasks. The evaluation system for user studies must be easy to use and record the important measurement data. [69], [50] and [8] conducted user studies to evaluate their systems. Some of the problems of user studies are clear. They are expensive in both time and labor. Also, the results could be biased since it is difficult to consider all of the factors that may affect the results, such as the form and wording of the questionnaire, and

the experience and background of the participants. A new trend in user studies is to use incentive schemes called crowdsourcing for collecting training data from users on the web. Mechanical Turk [1] is a good example. This system has the potential to be a good platform to conduct user experiments for interactive vision systems since it is easier to recruit more people and collect user data in a short time. But it still suffers some disadvantages such as possibly biased results of user studies.

Automatic Evaluation by Simulating User Interactions

To conduct automatic evaluation, we need to replace the human user with an algorithmic process designed to simulate user interactions. Some system evaluations have incorporated this technique. For example, [87] evaluates the usability of the systems by simulation of users' navigational paths to specific commands from graphical user interfaces. [53] evaluates a spoken dialog systems by user simulation.

In the image world, several automatic evaluation methods have been proposed for medical image segmentation algorithm. [66] accepts a single seed point and evaluates how sensitive the algorithm is to various single seed placements. [75] extends [66]'s work by accepting multiple seeds as inputs and evaluating a 3D segmentation algorithm. More recently, [77] proposed a simulated model of a human user and then used it to evaluate and learn parameters of a state-of-the-art interactive segmentation system; [70] proposed a new evaluation framework for interactive segmentation methods of natural images by automating user interactions. The first two methods focus on medical imaging and the latter two can work on nature images. However, [77] and [70] both miss some important quality measures: [77] only measures the error rate by different robot users during iterations; [70] does not measure repeatability such as robustness to scribbles with different lengths and placements. The strategies [70] used do not include

5.3 Data Set, Evaluation Criteria and Algorithms in Our New Framework

random scribble placements, which we think are important to use as a baseline.

5.3 Data Set, Evaluation Criteria and Algorithms in Our New Framework

5.3.1 Data Set

The data set we are using is the Berkeley nature image segmentation dataset [67]. We chose 10 images that contained objects that could be concisely and unambiguously described to the experiment participants. The segmentation of each object was modified from the provided ground truth. The ten images are classified into five categories: uniform background with uniform foreground, uniform background with complex foreground, complex background with uniform foreground, complex background with complex foreground, and texture images. We also add two cervigram images with acetowhite regions, which is caused by whitening of potentially malignant regions of the cervix epithelium, following application of acetic acid to the cervix surface. These two cervigram are from National library of Medicine at NIH. The total of 12 images are listed in Appendix A.

5.3.2 Evaluation Criteria

As mentioned before, the criteria commonly used are accuracy and efficiency. We use Jaccard index or accuracy rate for accuracy. Sometimes we also use the error rate. Additionally, we also want to quantify the user experience by using the following controllability measures:

1. Robustness against variation in user inputs

5.3 Data Set, Evaluation Criteria and Algorithms in Our New Framework

Good interactive segmentations should be robust against small variations in user scribbles, which means the segmentation should be the same or nearly the same with some perturbed scribbles. These perturbations can be input into each step of iteration. In automatic evaluation, we use two methods: 1) by perturbation and 2) by different quality of scribbles. Both are described in detail in Section 5.6.3.

2. Number of iterations

We expect that too many iterations will make users frustrated, so good interactive segmentations should use as few iterations as possible. This argues for integrating the scribble recommendation system into the segmentation algorithms. As a quality measure, we calculate the number of iterations needed to achieve a specified level of accuracy in automatic evaluation. In user studies, we record the number of iterations required to achieve “satisfaction” as judged by the user.

3. Accuracy in each step

Good interactive segmentations should progressively decrease the error rate in each iteration. This is one measurement for fluctuation also. We calculate accuracy rate in each iteration.

4. Fluctuation in each step

Fluctuation measures whether the results are faithful to user intention. If not, users may choose to return to previous results. A good interactive segmentation method should avoid having this happened too often. This criteria will measure the number of times users return to the previous results.

5. Speed in each step

5.3 Data Set, Evaluation Criteria and Algorithms in Our New Framework

Good segmentation may take longer in the initial step, but should decrease the computation time in subsequent steps, since the corrections will work on smaller areas only.

For efficiency, we calculate several measures that embeds with the above controllability and details can be found in Section 5.6.2.

5.3.3 Algorithms

We focus on evaluating several interactive segmentation methods including our method (Chapter 4). They are all appropriated for object extraction from nature images. The final set of six algorithms is listed here:

1. DBP: label propagation on hypergraph via belief propagation, which is our method.
2. Fuzzy: interactive segmentation with fuzzy connectedness [103].
3. GraphCut: interactive graph cuts [19].
4. GrowCut: grow cut method [106].
5. MRF: interactive segmentation method based on MRF [9].
6. RW: random walks method [46].

We selected these graphical model algorithms so as to provide good coverage of the published algorithmic approaches designed for object extraction from natural scenes. We selected only algorithms whose input can be modeled by pictorial input on an image grid, allowing transparent integration into our scribble-driven segmentation tool. We did not consider active contour algorithms, as they require a different interaction model, and tend to be better suited to medical image segmentation applications.

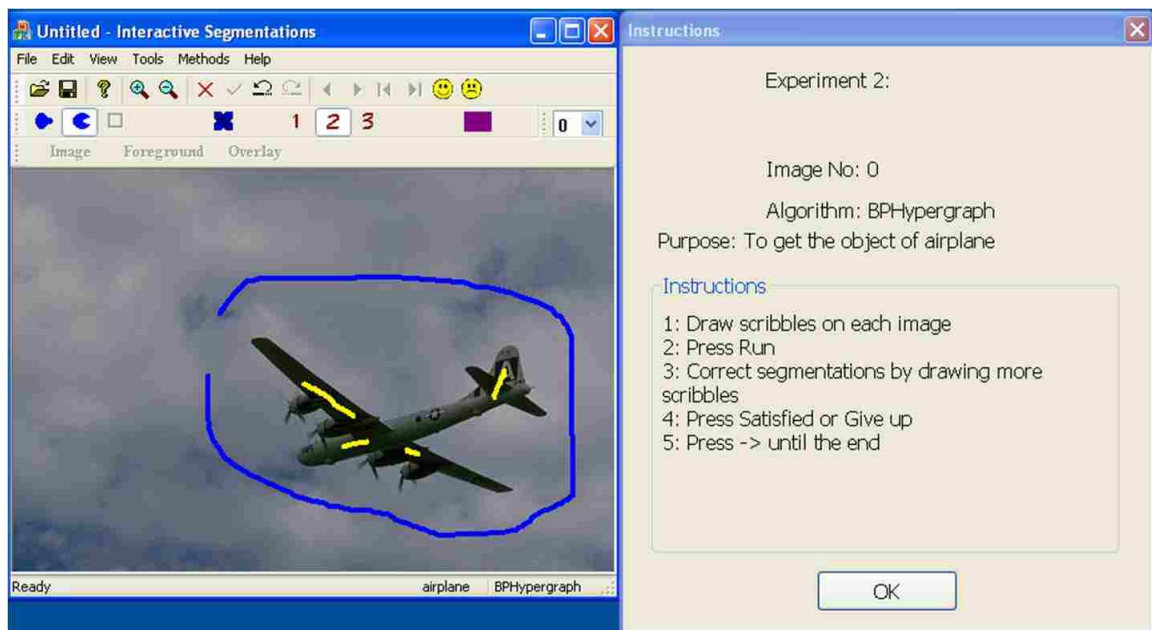


Figure 5.1: User interface of the interactive segmentation tool

5.4 User Studies in Our New Framework

A total of 40 volunteers participated in the user experiments. Each participant was first asked to complete a set of screening questions (see Appendix B). Some of the user characteristics are as follows: 25 were female, 15 were male; 37 of them had previously used some image manipulation program, such as Photoshop or GIMP; 30 of them had an engineering background.

Before the experiments, we showed demos and provided instructions about how to complete the required tasks. Then each participant was asked to extract objects in three experiments. No time limit was imposed for each experiment. To facilitate the experiments and host the segmentation algorithms, we developed an interactive segmentation tool (see Figure 5.1). This application was developed as a general purpose interactive segmentation tool, but includes a pop-up window to provide instructions.

During the experiments, users were presented with a series of object extraction tasks for images. As the user draws scribbles on the image with the mouse, the segmentation is updated. Then the user can add more scribbles to correct the results in several iterations until they are satisfied with the result or give up due to too many iterations. The tool records the time spent and the user scribbles in XML files. The segmentation results of images are saved in the form of binary images.

We discuss the analysis of these results in Section 5.6, and compare it with the results gathered from the automated experiments.

5.4.1 Experiment Design

We carried out three experiments. The total length of time taken for the three experiments was roughly one hour per user. The first experiment used one of the six methods (randomly chosen) to segment 12 images. The second experiment was to segment two images using all six methods. The methods are used in the alphabetical order that are listed in Section 5.3.3. The bias caused by method is minimized since most of the participants are not aware of these methods by their names. The third experiment was to use the scribble recommendation system with our method.

5.4.2 Questionnaire

After the experiments, we asked users several questions related to their experience, questions such as ranking the methods, ranking the properties of interactive segmentation methods, and whether the scribble recommendation system was useful. See Appendix C for details.

5.5 Automatic Evaluation by Machine-generated Scribbles in Our New Framework

The idea of automatic evaluation is to design an algorithmic process that can simulate, in some reasonable way, all the actions usually performed by a human user and we hope it can replace or augment expensive user studies. In interactive segmentation methods, only the first set of foreground and background scribbles supplied by the user in the first iteration can be used repeatedly; all other interactions are reactive, since the user is always attempting to correct the current segmentation errors in the following iterations. The errors themselves vary in iterations depending on the algorithm and user interactions from previous iterations.

Automating the evaluation requires us to identify each step that is made by the user, so that it can be replaced by an automated step. We identify the following user steps to carry out an interactive segmentation:

1. Identify the object to be extracted from the task description
2. Draw initial foreground and background scribbles
3. Correct errors in the segmentation by adding additional foreground or background scribbles
4. Decide after each interaction if the segmentation is satisfactory

We now discuss our approach to automatically simulating these step. The first step, identifying the object to be extracted from the task description can be bypassed since we are performing a supervised evaluation, so we already know the object in the ground truth. For the second step, we generate the initial scribbles in the object and background regions of the ground truth, using the random walks method described in

Section 5.5.1. After the initial segmentation, we compare this segmentation against the ground truth and find the set of mislabeled pixels. For the third step, we generate additional object or background scribbles from this set of mislabeled pixels. For the fourth step, the algorithm applies a straightforward accuracy rate criterion on the accuracy rate computed by and compare the segmentation to the ground truth. If the criterion is satisfied, the process terminates.

In this process, we select the additional scribbles from mislabeled pixel sets and assume object scribbles are always inside the ground truth object while background scribbles are outside the ground truth object. This means that the automated algorithm always makes the correct decision, i.e., it always correctly labels pixels. However, our experiments through user studies have shown that is not the case since sometimes during iterations users make mistakes.

The details of generating the initial scribbles are given in Section 5.5.1. The details for adding more scribbles during iterations are given in Section 5.5.2.

5.5.1 Generating Scribbles by Random Walks

For automatic evaluation, we first need to generate scribbles that mimic the ones drawn by the actual users. We developed a mechanism to generate synthetic scribbles which we modeled as (smooth) random walks. We positioned both the background and foreground synthetic scribbles in the background and foreground regions, respectively, of the ground truth. We initialize our scribble generation by sampling a starting point in the image uniformly at random. We then randomly sample a direction angle such that it is highly correlated with the previous direction sample (for smoothness). We extend the scribble along this direction as long as it does not cross object boundaries, as indicated by the ground truth. This approach generates scribbles of arbitrary length



Figure 5.2: Scribbles generated by random walk

and direction. One example of such background and foreground scribbles is shown in Figure 5.2.

5.5.2 Simulating User Interactions by Different Strategies

We explored two alternative strategies for initializing scribbles and for adding more scribbles in the iterations.

The basis of our strategies is the observation that users tend to begin extracting objects by marking as foreground some pixels in the approximate middle of the object, and marking as background some pixels well outside the object. During the iterations, they refine the segmentation by adding scribbles that lie inside large areas of misclassified pixels. To emulate this behavior, our first strategy is to initialize the segmentation by selecting pixels that are near the center of the ground truth object as object seeds, and selecting pixels that are near the center of the background region in the ground truth as background seeds. Similarly, during iterations, we add more scribbles near the center of the largest groups of misclassified pixels. We refer to this strategy as the

“Largest” strategy.

Our second strategy is to simply initialize at a random pixel location in the ground truth object, and at a random pixel location in the background region. We refer to this strategy as the “Random” strategy.

5.6 Evaluation Results and Analysis

We report accuracy, efficiency and controllability measures in this section. The objective of our analysis is to compare results from automatic evaluation with those from user study.

5.6.1 Accuracy

Here we examine the average accuracy over all images in the first step and final accuracy after the first 5 iterations.

Accuracy in first step

We calculated accuracy in the first step from automatic evaluation with two scribble initialization strategies (Section 5.5.2) and from the user study. For automatic evaluation, we generated scribbles with length of 400 pixels. In the user study, we averaged all results from all users from the second experiment.

Figures 5.3 show average accuracy in the first step from automatic evaluation and user study. Note that in Figure 5.3, the two scribble initialization strategies are denoted “Largest” and “Random”.

As expected, the largest region strategy is much better than the random strategy. Both strategy profiles indicate that DBP and random walks algorithms are comparable, and both perform better than the other algorithms. The user study shows results similar

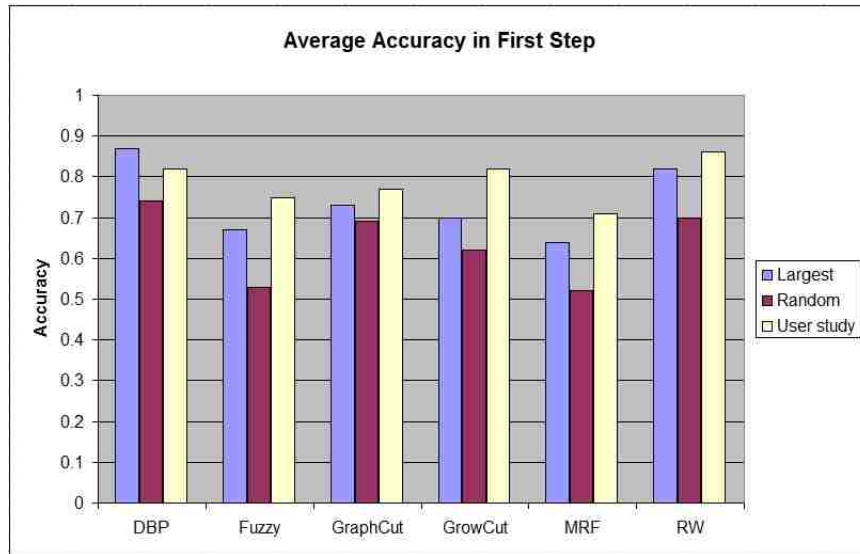


Figure 5.3: Average accuracy in the first step for automatic evaluation and user study

to the automatic evaluation, although the GrowCut algorithm also performs about as well as DBP and random walks.

Final accuracy in certain number of iterations

For the automated evaluation, we computed final accuracy after 5 iterations. We started with scribble length 600 pixels and then extended scribbles by 10 pixels after each iteration; for the user study, we computed final accuracy after the iteration which gave segmentation results which satisfied the user, but limited the iteration to 5. So, for the user study, 5 or fewer iterations were used in the averaging. The results are shown in Figure 5.4 for automatic evaluation and user study.

By the profiles of first accuracy and final accuracy we see that the DBP algorithm initially performs better than the random walks algorithm, but after 5 iterations the performance of random walks surpasses the DBP algorithm. Note that this observation holds for both the automated evaluation and for the user study. We interpret

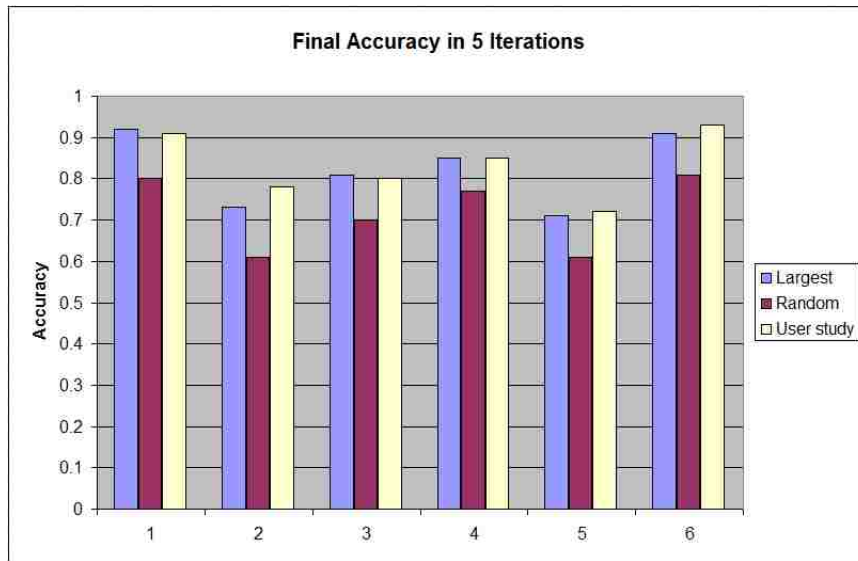


Figure 5.4: Average final accuracy after 5 steps for automatic evaluation and user study

this behavior as indicating that the random walks algorithm is more receptive to user interactions, in some sense.

5.6.2 Efficiency

Efficiency is calculated by two measures: the number of iterations and speed in the each step during the whole process.

Number of Iterations

Next, we wanted to characterize the algorithms by the amount of computation required to reach a specified level of accuracy. For the automated evaluation, we started with scribble length 50 pixels and extended scribble by 5 pixels in each step to achieve 90% accuracy. In the user study, we calculated the number of iterations when users reported satisfaction with the results.

Figure 5.5 shows the average number of steps to the accuracy of 0.9 in the au-

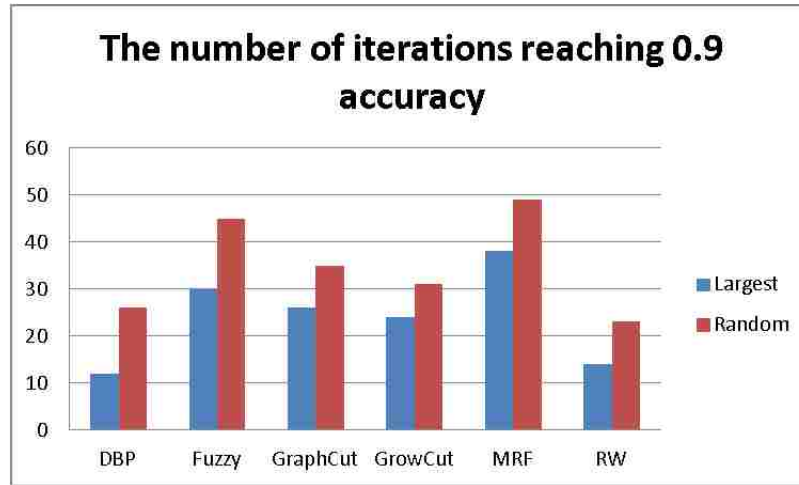


Figure 5.5: Number of iterations to reach 90% accuracy for automatic evaluation

automatic evaluation. From Figure 5.3, 5.4 and 5.5, it is clear that the largest region strategy, which is deterministically selecting the center of the largest candidate pixel region for each interaction, is an effective interaction pattern for the algorithms tested; on average it obtains a more accurate segmentation with fewer interactions than the random strategies. Table 5.1 shows the number of iterations to reach satisfaction in the user study. It approximately agrees with the results from automatic evaluation, where random walks and DBP can reach reasonable accuracy in fewer iterations while fuzzy and MRF perform the worst in terms of number of iterations.

Table 5.1: The number of iterations to reach satisfaction

DBP	Fuzzy	GraphCut	GrowCut	MRF	RW
3.5	5.6	4.9	4.3	5.7	3.6

Speed in the Whole Process

We calculated the speed (execution time) in the first 5 iterations in automatic segmentation and show these results in Figure 5.6. Table 5.2 shows the average time for

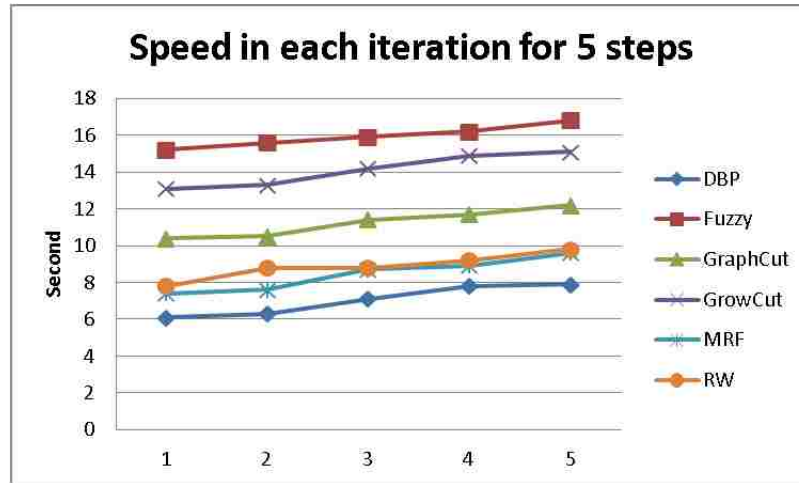


Figure 5.6: Speed in each step for 5 iterations

one iteration for the different methods in the user study. From Figure 5.6, we can see that all algorithms increase execution time during iterations which is not desirable in interactive segmentation. Our user study matches the automatic evaluation results in the sense that the fuzzy algorithm performs the worst, and that other algorithms are comparable in terms of speed.

Table 5.2: Speed in one iteration in user study (sec)

DBP	Fuzzy	GraphCut	GrowCut	MRF	RW
5.7	15.2	6.3	6.7	6.1	6.8

5.6.3 Sensitivity to Scribbles

Scribbles with varying lengths and positions affect segmentation accuracy. Generally speaking, the longer the length of scribbles, the more information is input to the segmentation methods, so we expect better segmentation results. Segmentation results are also affected by the positions of the scribbles. We used a perturbation method to analyze the sensitivity of the segmentations to scribble position.

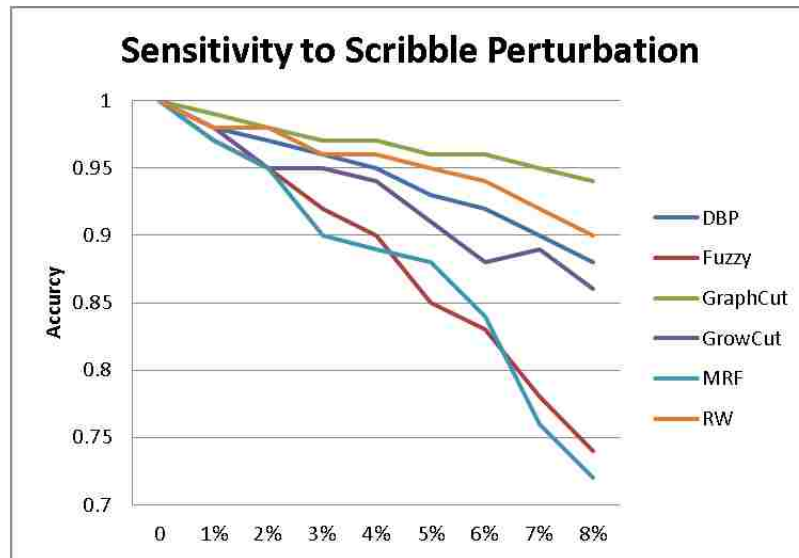


Figure 5.7: Sensitivity analysis of segmentation by scribble perturbation.

Sensitivity by Scribble Perturbation

Given the initial segmentation, the locations of the scribble were shifted as a group in a random direction with random magnitude. The variance of the magnitude ranged from 1% to 8% of the image size. Any sample in which the perturbation caused the foreground (background) scribbles to enter the background (foreground) was rejected and then retried until a valid perturbation was obtained.

Figure 5.7 shows that in this test, graph cuts demonstrates the greatest robustness to scribble placement within an object. This result is not surprising, since one might expect that the cost of the object boundary is generally lower than the cost of internal boundaries and therefore the location of the source/sink within the object should not have any effect. In contrast, the MRF algorithm exhibits the least robustness to seed placement within an object. This effect may be explained by the inherent dependence of model training on the given scribbles.

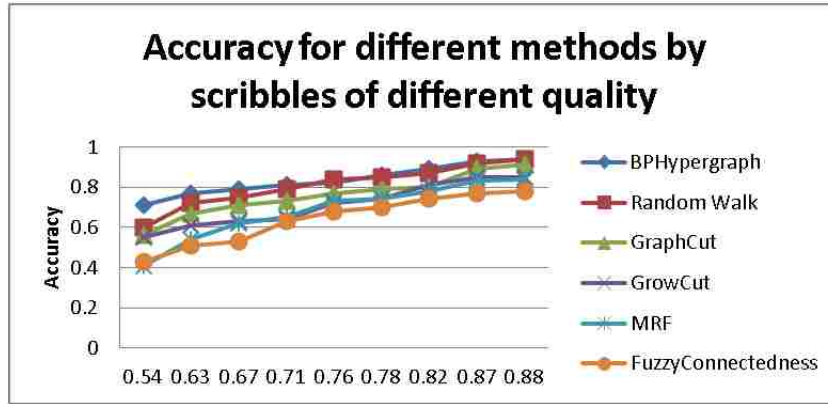


Figure 5.8: Sensitivity analysis of segmentation by different quality of scribbles

Sensitivity by Scribble Quality

In order to compare our method to the other methods with respect to “quality” of input scribbles, we need to define what this quality means. Since different methods lead to different segmentation results using the same scribbles, our solution was to use the same scribbles across all the methods, then use average accuracy as the measure of the quality of those scribbles. A higher average accuracy corresponds to higher quality scribble. The average accuracy was calculated by leave-one-out method.

Then we used scribbles generated by machine with different qualities (as defined above) to evaluate the segmentation accuracy, comparing our methods with other methods. Figure 5.8 shows the evaluation result. From the evaluation, we can see that our method (DBP) and random walks are among the top two methods with respect to accuracy obtained by a scribble of given quality.

5.6.4 Results from User Questionnaires

For the subjective evaluation of the methods by users, 36 of 40 people ranked our method among the top two methods along with random walks. The two most important

properties of interactive segmentation methods that users chose were accuracy in the first iteration and fewer iterations towards satisfaction.

5.7 Discussion

When introducing a new interactive segmentation algorithm it is important to be able to compare its performance with the state-of-the-art. In this chapter we develop a set of measurements and software for supervised evaluation of interactive segmentation using user study. Carrying out these user experiments is, however, a time-consuming and labor-intensive exercise, often prohibitively so. Thus in this chapter we also propose automatic evaluation by machine-generated scribbles. To this end, we investigated two strategies for automating the evaluation of interactive segmentation algorithms. The objective of these strategies is to simulate interactions that would normally be provided by a human operator using the ground truth and current segmentation error.

The first of two strategies is a simple, deterministic strategy: it always produces the same set of interactions given the same segmentation algorithm and input. The other strategy had a random component, aiming to more closely approximate the interactions usually produced by humans.

The experiments demonstrated that the results of the automated experiments are very similar to those of the user study. Evaluating the six segmentation algorithms using two strategies produced similar conclusions about the evaluated algorithms, and these conclusions agreed with the user study.

The automation strategies that we discussed in this chapter are perhaps the most useful when used as a preliminary step in an evaluation process. They allow algorithm developers to experiment with different variants of an algorithm to determine the most effective ones, without having to re-conduct user studies each time. Automated eval-

uation also provides a means for researchers to determine if a particular approach to interactive segmentation has practical merit, if it requires further consideration, if it needs modification, or if it should be abandoned, before expensive user experiments are undertaken. If feasible, user study should be performed for the final evaluation of an algorithm when comparing it against the state-of-the-art.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we proposed three interactive segmentation methods based on parametric, geometric active contour and graphical models.

In Chapter 2, we proposed interactive polygons for Active Volume Models (AVM) which is an improved parametric active contour model. The two kinds of interaction polygons we apply are “merge polygons” and “split polygons” which identify the foreground and background, respectively. Users can easily correct segmentation results in the iterations by adding interactive polygons.

In Chapter 3, we proposed a new interactive segmentation method which integrates user scribbles into a localized geometric active model or localized level set, where scribbles are represented as a continuous energy term added into the level set functional.

In Chapter 4, we proposed a new interactive segmentation method which first model image as a hierarchical hypergraph; then, we used belief propagation to propagate the labels of seed pixels to the remaining unlabelled pixels. We implemented two belief

propagation algorithms: discrete belief propagation, and Gaussian belief propagation, in which the label variables are modeled as discrete and Gaussian variables, respectively. Also we propose a user interaction which allow users to change the connectivity of segmentation results. We also integrate a recommend scribble feature into the framework that intelligently recommends where the user should scribble next.

In Chapter 5, in order to evaluate interactive segmentation methods based on graphical models, we presented a new framework which combines user study with automated interaction through machine-generated scribbles. We also proposed new evaluation measures to quantify the user experience, and strategies for selecting where to put scribbles in the next iteration to mimic a real user. We then used this framework to evaluate several interactive segmentation methods, including our proposed method and the recommend scribble feature.

6.2 Future Work

Image segmentation results are useful in computer-assisted diagnosis, treatment planning and intervention tasks, in the biomedical world. It also has my applications n general purpose images. Our research supports these uses by contributing a flexible method of user interaction which produces results which are accurate and efficient, as compared to other methods.

There is room for improvement in our proposed three interactive segmentation methods. Some of them are discussed below. In the future, more user interactions can be explored to make the proposed methods easier for users to guide the process; also, the application areas of proposed methods can be also extended to further explore whether the proposed methods are useful in object recognition and image retrieval. For instance, we can extend our methods to 3D images. In addition, we can take advantage

of recent developments in Graphic Processing Unit (GPU) technology and parallel architectures, we can also implement algorithms for real-time 3D interactive image segmentation. Another direction is investigating learning methods and integrates them into the interactive segmentation methods. We expect by this approach the execution time in each step can be decreased progressively.

For interactive segmentation method based on Active Volume Models, texture and tensor-based information can be used to predict object ROIs so that AVM can work on color images or texture images. We can also explore adding user scribbles as interaction method, since they are more flexible than interactive polygons.

For interactive segmentation based method on level sets, we can use the scribbles as the initial interaction instead of an initial contour.

For our interactive segmentation method based on graphical models, we can explore other interactions. For example, users could draw rectangles around erroneous areas and the method would fix the segmentation automatically.













For the evaluation framework, instead of using accuracy rate, which is region-based, we can explore the boundary accuracy, which is possible more useful for evaluating object segmentations. We can also add more strategies of selecting the starting points and adding more scribbles. Also, a method for directly comparing real user interactions and simulated interactions would undoubtedly simplify any investigation of potential improvements to the automation strategies. Developing such a method is, however far from straightforward. The user interactions and automated interactions are both reactive and non-deterministic, which precludes a direct spatial comparison. It may, however, be possible to compare interactions based on shape or some other features. A detailed analysis of user interactions may provide more insights into how direct comparison may be realized.

Appendix A

Image Dataset for Evaluation

The image dataset used in Chapter 5 are 10 images from Berkeley segmentation data set and 2 cervigrams from National Library of Medicine at NIH. The first ten images are classified into five categories: uniform background with uniform foreground, uniform background with complex foreground, complex background with uniform foreground, complex background with complex foreground, and texture images. All the images and their categories are listed in Table A.1.

Table A.1: Image dataset and its categories

Uniform foreground and uniform background		
Uniform foreground and complex background		
Complex foreground and uniform background		
Complex foreground and complex background		
Texture		
Cervigram		

Appendix B

Recruitment Screening Protocol for Evaluation of Interactive Image Segmentation Methods

Name: _____ Participant # _____

Email: _____ Phone number _____

Appointment date/time: _____

1. To ensure we recruit a representative group of users, please tell us your gender.

[Recruit approx. 50-50 mix]

_____ Male

_____ Female

2. For classification purposes only, which age category do you fall into? [Recruit a mix]

_____ Under 18 EXCLUDE

18 - 39

40 - 59

60 - 74

75+

3. Again, to ensure that we recruit a representative mix of users, what is the highest level of education that you have completed? [\[Recruit a mix\]](#)

High school graduate or GED

Some college

Bachelor's degree

Graduate degree

Doctorate degree

4. What is your occupation? [\[Recruit a mix\]](#)

Usability Professional - EXCLUDE

Web developer - EXCLUDE

Application Programmer

Student

Educator

IT Professional

Scientist or Researcher

Health Care Professional

Librarian (including other information professionals)

Other. Please specify _____

5. Are you a Federal government employee? [\[Recruit as many as possible\]](#)

_____ Yes

_____ No

6. Are you familiar with the terms image segmentation and/or image manipulation?

[Recruit a mix of users]

_____ Yes

_____ No

_____ Not sure

7. How often have you performed the following tasks? [Exclude if Never in one or more categories]

	Never	Infrequently	Sometimes	Often	Daily
Use a basic photo or image editing program such as Photoshop, GIMP, or Picasa?					
Use software-based freehand drawing tools					
Use the typical features of an applications window such as title bar, toolbar, menu bar, scroll bar, and icons					

8. On a scale of 1 to 5 with 1 being extremely uncomfortable and 5 being very comfortable, how comfortable are you with sharing your opinion with others?

[Exclude if 1 or 2]

Uncomfortable 1 2 3 4 5 Comfortable

Appendix C

User Study Questionnaire

Questionnaires for Experiment 1 Specific method: _____

Q1: Ranking your satisfactions

- A. 5
- B. 4
- C. 3
- D. 2
- E. 1

Q2: What properties you think is true for this method

- A. More accurate in the first iteration
- B. Easy to correct the first iteration

-
- C. No fluctuation in the whole process (adding more scribbles should not make the result worse)
 - D. Fewer strokes are required
 - E. Fewer iterations until satisfied
 - F. Other reason, specify: _____

Questionnaires for Experiment 2 (Multiple methods)

Q1: What methods you like (list in order of most like to least like)

- A. BPHypergraph
- B. Fuzzy Connectedness
- C. Graph Cut
- D. Grow Cut
- E. MRF
- F. Random Walks

Q2: Comparing multiple methods, list the properties you think a good interactive segmentation method should have(in order of most important to least important)

- A. More accurate in the first iteration
- B. Easy to correct the first iteration

-
- C. No fluctuation in the whole process (add more scribbles may make the result worse)
- D. Fewer strokes are required
- E. Fewer iterations until satisfied
- F. Other reason, specify: _____

Questionnaires for Experiment 3

Q1: Do you find our scribble recommendation helpful

- A. Yes
- B. No
- C. Does not matter having it or not

References

- [1] Amazon mechanical turk. <http://www.mturk.com/>. 101
- [2] Fastica. <http://www.cis.hut.fi/projects/ica/fastica/index.shtml>. 50
- [3] Grabcut data set. <http://research.microsoft.com/en-us/um/cambridge/>. 99, 100
- [4] L. Z. B. Peng and D. Zhang. A survey of graph theoretical approaches to image segmentation. Technical report, Department of Computer Science, The Hong Kong Polytechnic University. 20, 21
- [5] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? a unified approach to segment extraction. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5305 of *LNCS*, pages 30–44. Springer, 2008. 63
- [6] X. Bai and G. Sapiro. Distancecut: Interactive segmentation and matting of images and videos. In *ICIP (2)*, pages 249–252. IEEE, 2007. 60, 63, 64
- [7] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010. 83
- [8] D. Batra, C. M. Univerity, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 100
- [9] M. Berthod. Bayesian image classification using markov random fields. *Image and Vision Computing*, 14(4):285–295, 1996. 104
- [10] M. Berthod, Z. Kato, S. Yu, and J. Zerubia. Bayesian image classification using markov random fields. *Image and Vision Computing*, 14:285–295, 1996. 63
- [11] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–

- 109, 1943. [49](#)
- [12] M. Bilgic and L. Getoor. Reflect and correct: A misclassification prediction approach to active inference. *ACM Transactions on Knowledge Discovery from Data*, 3(4):1–32, November 2009. [100](#)
- [13] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, pages 428–441, 2004. [31](#)
- [14] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, pages 428–441, 2004. [58](#), [60](#), [64](#), [88](#)
- [15] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, 2004. [100](#)
- [16] M. Bleyer and M. Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Image Commun.*, 22(2):127–143, Feb. 2007. [7](#)
- [17] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, pages 19–26. Morgan Kaufmann, San Francisco, CA, 2001. [62](#), [63](#), [64](#)
- [18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001. [60](#), [64](#)
- [19] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001. [104](#)
- [20] T. Buck, H.-H. Ehrlicke, W. Straer, and L. Thurffjell. 3-d segmentation of medical structures by integration of raycasting with anatomic knowledge. *Computers and Graphics*, 19(3):441–449, 1995. [22](#)
- [21] J. E. Cabral, K. S. White, Y. Kim, and E. L. Effmann. Interactive segmentation of brain tumors in MR images using 3D region growing. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1898 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 171–181, Sept. 1993. [22](#)
- [22] V. Caselles. Geometric models for active contours. In *Proceedings of the 1995 Interna-*

- tional Conference on Image Processing (Vol. 3)-Volume 3 - Volume 3*, ICIP '95, pages 3009–, Washington, DC, USA, 1995. IEEE Computer Society. 14
- [23] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1997. 17, 44, 45
- [24] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10:266–277, 2001. 18, 23, 45
- [25] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009. 97
- [26] K. C. Ciesielski, J. K. Udupa, P. K. Saha, and Y. Zhuge. Iterative relative fuzzy connectedness for multiple objects with multiple seeds. *Comput. Vis. Image Underst.*, 107(3):160–182, 2007. 64
- [27] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:666, 1996. 78, 80
- [28] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. 63
- [29] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. 7
- [30] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72:195–215, 2007. 14
- [31] L. Ding and A. Yilmaz. Image segmentation as learning on hypergraphs. In *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*, pages 247–252, Washington, DC, USA, 2008. IEEE Computer Society. 58, 88
- [32] O. Duchenne and J.-Y. Audibert. Fast interactive segmentation using color and textural information. Technical Report Technical report 06-26, CERTIS, 2006. 60
- [33] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Ségonne. Segmentation by transduction. In *CVPR*, 2008. 63, 64

- [34] A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. d. A. Lotufo. User-steered image segmentation paradigms: live wire and live lane. *Graph. Models Image Process.*, 60(4):233–260, July 1998. 21
- [35] X. Fan, P. Bazin, and J. Prince. A multi-compartment segmentation framework with homeomorphic level sets. *CVPR*, 2008. 21
- [36] P. Favaro and S. Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:631–637, 2004. 7
- [37] Felzenszwalb, Pedro, Huttenlocher, and Daniel. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, October 2006. 65
- [38] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, Sept. 2004. 20
- [39] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004. 63, 68
- [40] P. Fua and C. Brechbühler. Imposing hard constraints on soft snakes. In *ECCV*, pages 495–506, 1996. 31
- [41] P. Fua and C. B. Y. Imposing hard constraints on soft snakes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 495–506. Springer-Verlag, 1996. 23
- [42] J. Fwu and P. M. Djuric. Unsupervised vector image segmentation by the icm method. In *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*, pages 2235–2238, Washington, DC, USA, 1996. IEEE Computer Society. 63
- [43] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. pages 716–723 vol.1, 2003. 63
- [44] F. Ge, S. Wang, and T. Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *Computer Vision and Pattern Recognition*, pages 1146–1153, 2006. 86
- [45] F. Ge, S. Wang, and T. Liu. New benchmark for image segmentation evaluation. *J.*

- Electronic Imaging*, 16(3):033011, 2007. 97
- [46] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:1768–1783, 2006. 9, 21, 22, 104
- [47] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006. 31
- [48] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006. 58, 62, 63, 64, 68
- [49] D. Greig, B. Porteous, and A. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. *Royal Journal on Statistical Society*, 51(2):271–279, 1989. 20
- [50] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 100
- [51] K. ho Seo, J. ho Shin, W. Kim, and J. jang Lee. Real-time object tracking and segmentation using adaptive color snake model. *International Journal of Control, Automation, and Systems*, 4(62):236–246, 2006. 6
- [52] X. Jiang, C. Marti, C. Irniger, and H. Bunke. Distance measures for image segmentation evaluation. *EURASIP J. Adv. Sig. Proc.*, 2006, 2006. 97
- [53] S. Jung, C. Lee, K. Kim, M. Jeong, and G. G. Lee. Data-driven user simulation for automated evaluation of spoken dialog systems. *Comput. Speech Lang.*, 23(4):479–509, Oct. 2009. 101
- [54] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision*, 1:321–331, 1987. 31
- [55] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988. 12, 23, 44
- [56] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(8):799–813, 1989. 63
- [57] C. L. Lam and S. Y. Yuen. An unbiased active contour algorithm for object tracking. *Pattern Recogn. Lett.*, 19(5-6):491–498, Apr. 1998. 6

- [58] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008. [xiv](#), [45](#), [46](#), [47](#), [48](#)
- [59] A. Levinstein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Sidiqqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009. [66](#)
- [60] C. Li, C. Kao, J.C.Gore, and Z. Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Imag. Proc.*, 17(10):1940–1949, October 2008. [32](#), [36](#)
- [61] Y. Li, D. Lu, X. Lu, and J. Liu. Interactive color image segmentation by region growing combined with image enhancement based on bezier model. *Image and Graphics, International Conference on*, 0:96–99, 2004. [63](#)
- [62] L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):529–540, 1990. [22](#)
- [63] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17:158–175, 1995. [14](#), [44](#)
- [64] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2):158–175, Feb. 1995. [14](#)
- [65] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:158–175, 1995. [15](#)
- [66] F. Mao, J. D. Gill, and A. Fenster. Technique for evaluation of semi-automatic segmentation method. In *Proceedings of the SPIE: Medical Imaging*, pages 10271036,, 1999. [101](#)
- [67] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. [87](#), [102](#)

- [68] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB/CSD-01-1133, EECS Department, University of California, Berkeley, Jan 2001. [99](#)
- [69] K. McGuinness and N. E. O'Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, February 2010. [100](#)
- [70] K. McGuinness and N. E. O'Connor. Toward automated evaluation of interactive segmentation. *Comput. Vis. Image Underst.*, 115(6):868–884, June 2011. [97](#), [98](#), [99](#), [101](#)
- [71] T. Mcinerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. pages 840–845, 1999. [23](#)
- [72] T. Mcinerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73 – 91, 2000. [31](#)
- [73] M. Meng, L. Fan, and L. Liu. A comparative evaluation of foreground/background sketch-based mesh segmentation algorithms. [97](#), [98](#)
- [74] O. V. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the bhattacharyya gradient flow. *IEEE Transactions on Image Processing*, 16(11):2787–2801, 2007. [49](#)
- [75] E. Moschidis and J. Graham. A systematic performance evaluation of interactive image segmentation methods based on simulated user interaction. In *Proceedings of the 2010 IEEE international conference on Biomedical imaging: from nano to Macro*, ISBI'10, pages 928–931, Piscataway, NJ, USA, 2010. IEEE Press. [101](#)
- [76] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989. [17](#)
- [77] H. Nickisch, C. Rother, P. Kohli, and C. Rhemann. Learning an interactive segmentation system. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '10, pages 274–281, New York, NY, USA, 2010. ACM. [101](#)
- [78] R. Nock and F. Nielsen. Grouping with bias revisited. *Computer Vision and Pattern*

- Recognition, IEEE Computer Society Conference on*, 2:460–465, 2004. [60](#), [63](#), [64](#)
- [79] I. Nwogu and J. J. Corso. Bp2: Beyond pairwise belief propagation labeling by approximating kikuchi free energies. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008. [66](#)
- [80] S. Olabarriaga and A. Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2):127–142, June 2001. [xii](#), [8](#), [95](#)
- [81] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *JOURNAL OF COMPUTATIONAL PHYSICS*, 79(1):12–49, 1988. [14](#)
- [82] M. Ozkan, B. Dawant, and R. Maciunas. Neural-network-based segmentation of multi-modal medical images: A comparative and prospective study. *IEEE Trans. Medical Imaging*, 12:534–544, 1993. [7](#)
- [83] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition*, CVPR’03, pages 145–152, Washington, DC, USA, 2003. IEEE Computer Society. [21](#)
- [84] D. L. Pham, C. Xu, and J. L. Prince. A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–338, 2000. [6](#)
- [85] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *Image Processing, IEEE Transactions on*, 16(4):1046–1057, 2007. [60](#), [63](#), [64](#)
- [86] W. Reddick, J. Glass, E. Cook, T. Elkin, and R. Deaton. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. *IEEE Trans. Medical Imaging*, 16:911–918, 1997. [7](#)
- [87] M. O. Riedl and R. S. Amant. Toward automated exploration of interactive systems, 2002. [101](#)
- [88] S. Rital, H. Cherifi, and S. Miguet. Weighted adaptive neighborhood hypergraph partitioning for image segmentation. In *ICAPR (2)*, pages 522–531, 2005. [66](#)
- [89] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, August 2004. [xv](#), [87](#), [92](#)

- [90] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of CVPR*, June 2006. 6
- [91] A. S. B. Samma and R. A. Salam. Adaptation of k-means algorithm for imagesegmentation. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 38, 2009. 63
- [92] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A level set model for image classification. *International Journal of Computer Vision*, 40:187–197, 2000. 18
- [93] R. E. Schapire. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification*, 171:149–171, 2002. 7
- [94] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, June 1999. 80
- [95] T. Shen, Y. Zhu, X. Huang, J. Huang, D. Metaxas, and L. Axel. Active volume models with probabilistic object boundary prediction module. *MICCAI*, 5241:331–341, 2008. 23
- [96] T. Shen, Y. Zhu, X. Huang, J. Huang, D. Metaxas, and L. Axel. Active volume models with probabilistic object boundary prediction module. In *MICCAI (1)*, pages 331–341, 2008. 30
- [97] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000. 20
- [98] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 62, 63, 64
- [99] G. Sivewright and P. Elliott. Interactive region and volume growing for segmenting volumes in mr and ct images. *Med. Inform.*, 19(1):71–80, 1994. 22
- [100] O. Subakan and B. Vemuri. Image segmentation via convolution of a level-set function with a Rigaut kernel. *CVPR*, 2008. 21
- [101] R. Tsai and S. Osher. *Level Set Methods and Their Applications in Image Science*, 2003. 44
- [102] J. K. Udupa, V. R. LeBlanc, Y. Zhuge, C. Imielinska, H. Schmidt, L. M. Currie, B. E.

- Hirsch, and J. Woodburn. A framework for evaluating image segmentation algorithms. *Comp. Med. Imag. and Graph.*, 30(2):75–87, 2006. [95](#), [97](#)
- [103] J. K. Udupa and S. Samarasekera. *Graphical Models and Image Processing*, (3):246–261. [104](#)
- [104] J. K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *CVGIP: Graphical Model and Image Processing*, 58(3):246–261, 1996. [64](#)
- [105] O. Veksler. Star shape prior for graph-cut image segmentation. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 454–467, Berlin, Heidelberg, 2008. Springer-Verlag. [61](#)
- [106] A. V. Vezhnevets. "growcut" - interactive multi-label n-d image segmentation by cellular, 2005. [104](#)
- [107] V. Vezhnevets and V. Konouchine. "grow-cut" - interactive multi-label n-d image segmentation. In *Graphicon*, 2005. [60](#), [63](#), [64](#)
- [108] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008. [61](#), [78](#)
- [109] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2008. [64](#)
- [110] Y. Weiss and W. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 23(13):2173–2200, 2004. [75](#)
- [111] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. 47(2):723–735, 2001. [72](#)
- [112] Z. Wu and R. Leahy. Tissue Classification In MR Images Using Hierarchical Segmentation. pages 1410–1414. IEEE, Oct. 1990. [20](#)
- [113] C. Xu, A. Yezzi, Jr., and J. L. Prince. On the relationship between parametric and geometric active contours. pages 483–489, 1999. [21](#)
- [114] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to Snakes for bimodal and trimodal imagery. *ICCV*, 2:898–903, 1999. [18](#)

- [115] S. X. Yu, R. Gross, and J. Shi. Concurrent object recognition and segmentation by graph partitioning. In *NIPS*, pages 1383–1390. MIT Press, 2002. [6](#)
- [116] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20(1):68–86, Jan. 1971. [20](#)
- [117] R. Zass and A. Shashua. A unifying approach to hard and probabilistic clustering. *Computer Vision, IEEE International Conference on*, 1:294–301, 2005. [60](#)
- [118] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:884–900, 1996. [44](#)

Curriculum Vita

Yaoyao Zhu

Short Bio

Yaoyao Zhu was born in Cixi City, Zhejiang Province, China. Her research interests include computer vision and image processing, medical imaging, computer graphics and machine learning.

Education

1995 B.S. in Electronics, Peking University.
1998 M. Sc. in Electronics, Peking University.
2002 M. Sc. in Computer Engineering, University of Cincinnati.

Research

1998-09 to 2001-05 Research Assistant, Applied Artificial Intelligence Laboratory, University of Cincinnati

2004-09 to 2006-05 Research Assistant, LONGLAB Laboratory, Lehigh University

2007-09 to 2012-05 Research Assistant, Image Data Emulation and Analysis (IDEA) Laboratory, Lehigh University

Work

2001-01 to 2002-01 Design Engineer, Lucent Technologies, Microelectronics Group
(now Alcatel-Lucent)
2006-06 to 2006-08 Summer Intern, Bell Labs, Lucent Technologies (now Alcatel-Lucent)

2007-06 to 2007-08 Summer Intern, National Library of Medicine (NLM), National Institutes of Health (NIH)
2011-05 to 2011-08 Summer Intern, National Library of Medicine (NLM), National Institutes of Health (NIH)
2012-06 to present Develop/Software Engineer, Oat Systems, Division of Checkpoint Systems

Publications

Journal

Xiaolei.Huang, **Yaoyao Zhu**, et al, “*Hybrid deformable image registration using a closed-form Free Form Deformation approach*”, Int. J. Biomedical Engineering and Technology, 2011

Zhiyun Xue, L. Rodney Long, Sameer Antani, Leif Neve, **Yaoyao Zhu**, George R. Thoma, “*A unified set of analysis tools for uterine cervix image segmentation*”, Comp. Med. Imag. and Graph. 34(8): 593-604 (2010)

Yaoyao Zhu, X.Huang, et al, “*Balancing the Role of Priors in Multi-observer Segmentation Evaluation*”, J. Sign. Process Syst., 2009

Qiang Wang, **Yaoyao Zhu**, et al, “*Wireless Sensor Networks: Challenges and Approaches*”, IEEE Network, Sept, 2006 (citation 116)

Lawrence Mazlack, **Yaoyao Zhu**, et al, “*Autonomous Granulation Using The Mountain Method*”, International Journal Of Intelligent Systems, April, 2005, v 20, n 4, 415-432

Conference, Abstract and Book Chapter

Yaoyao Zhu, Tian Shen, et al, “*Interactive Polygons in Region-based Deformable Contours for Medical Images*”, Proc. of the IEEE Intl Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009, Boston

Wei Wang, **Yaoyao Zhu**, et al, “*A Classifier Ensemble based on Performance Level Estimation*”, International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009, Boston. (Oral)

Tian Shen, **Yaoyao Zhu**, et al, “*Active Volume Models with Probabilistic Object Boundary Prediction Module*”, Proc. of the 11th Annual International Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI), 2008, New York.

Yaoyao Zhu, Xiaolei Huang, et al, “*Web-based Multi-observer Segmentation Evaluation Tool*”, 21st IEEE International Symposium on Computer-Based Medical Systems (CBMS), Jyvaskyla, Finland, 2008

Sarah Coppock, **Yaoyao Zhu**, et al, “*Experiments in Rough Set Based Data Mining*”, Proceedings ANNIE 2001, St. Louis, November, 2001, 339-344, published in: Smart Engineering Systems Design, v 11, ASME Press, New York

Aijing He, **Yaoyao Zhu**, et al, “*Data Discovery Using Rough Set Based Reductive Partitioning: Some Experiments*”, Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference 2001, Vancouver, July, 2001, 203-208

Lawrence Mazlack, Aijing He, **Yaoyao Zhu**, “*Rough Set Based Reductive Data Mining or Who Died On The Titanic?*”, Proceedings of International Workshop on Rough Set Theory and Granular Computing, v 5, n 1/2, May, 2001, 205-210

Lawrence Mazlack, Aijing He, **Yaoyao Zhu**, “*A Rough Set Approach In Choosing Partitioning Attributes*”, Proceedings of the ISCA 13th International Conference (CAINE-2000), November, 2000, 1-6

Yaoyao Zhu, “*Towards Better Segmentations*”, Lehigh Research Symposium 2010

Yaoyao Zhu, et al, “*Web-based STAPLE for Quality Estimation of Multiple Image Segmentation*”, NIH Research Festival, National Institute of Health, 2007