AUTOMATIC AND INTERACTIVE SEGMENTATIONS USING DEFORMABLE AND GRAPHICAL MODELS

BY MUSTAFA GOKHAN UZUNBAS

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Dimitris Metaxas

and approved by

New Brunswick, New Jersey

January, 2015

ABSTRACT OF THE DISSERTATION

Automatic and Interactive Segmentations Using Deformable and Graphical Models

by Mustafa Gokhan Uzunbas

Dissertation Director: Dimitris Metaxas

Image segmentation i.e. dividing an image into regions and categories is a classic yet still challenging problem. The key to success is to use/develop the right method for the right application. In this dissertation, we aim to develop automatic and interactive segmentation methods for different types of tissues that are acquired at different scales and resolutions from different medical imaging modalities such as Magnetic Resonance (MR), Computed Tomography (CT) and Electron Microscopy (EM) imaging.

First, we developed an automated segmentation method for segmenting multiple organs simultaneously from MR and CT images. We propose a hybrid method that takes advantage of two well known energy-minimization-based approaches combined in a unified framework. We validate this proposed method on cardiac four-chamber segmentation from CT and knee joint bones segmentation from MR images. We compare our method with other existing techniques and show certain improvements and advantages.

Second, we developed a graph partitioning algorithm for characterizing neuronal tissue structurally and contextually from EM images. We propose a multistage decision mechanism that utilizes differential geometric properties of objects in a cellular processing context. Our results indicate that this proposed approach can successfully partition images into structured segments with minimal expert supervision and can potentially form a basis for a larger scale volumetric data interpretation. We compare our method with other proposed methods in a workshop challenge and show promising results.

Third, we developed an efficient learning-based method for segmentation of neuron structures from 2D and 3D EM images. We propose a graphical-model-based framework to do inference on hierarchical merge-tree of image regions. In particular, we extract the hierarchy of regions in the low level, design 2D and 3D discriminative features to extract higher level information and utilize a Conditional Random Field based parameter learning on top of it. The effectiveness of the proposed method in 2D is demonstrated by comparing our method with other methods in a workshop challenge. Our method outperforms all participant methods except one. In 3D, we compare our method to existing methods and show that the accuracy of our results are comparable to state-of-the-art while being much more efficient.

Finally, we extended our inference algorithm to a proofreading framework for manual corrections of automatic segmentation results. We propose a very efficient and easy-to-use interface for high resolution 3D EM images. In particular, we utilize the probabilistic confidence level of the graphical model to guide the user during interaction. We validate the effectiveness of this framework by robot simulations and demonstrate certain advantages compared to baseline methods.

Preface

Parts of this research were originally appeared and published in International Conferences as follows:

Chapter 4: *M. G. Uzunbas, C. Chen, S. Zhang, K. M. Pohl, K. Li, and D. Metaxas, Collaborative multi organ segmentation by integrating deformable and graphical models, Medical Image Computing Computer Assistive Intervention vol. 16, no. 2, 2013.*

Chapter 5: *E. Bas, M. G. Uzunbas, D. Metaxas, and E. Myers, Contextual grouping in a concept: a multistage decision strategy for EM segmentation. IEEE International Symposium on Biomedical Imaging, 2012.*

Chapter 6: *M. G. Uzunbas, C. Chen, and D. Metaxas, Optree: a learning-based adaptive watershed algorithm for neuron segmentation., Medical Image Computing Computer Assistive Intervention vol. 17, no. 1, 2014.*

Acknowledgements

Many people deserve acknowledgement for preparation of this dissertation which took almost five years. My advisor, colleagues, friends and family all gave their endless support to help me reach to the end. I start by thanking my advisor, Dimitris Metaxas, for his great efforts to help me pursue my graduate study at Rutgers University computer science department. During the five years, his advice has been very valuable, and his long term vision and entrepreneurship kept me motivated throughout my research studies. Besides the financial support, his friendship (rather than a advisory role) has taught me a lot and always made me delighted.

I would also like to acknowledge my committee members: Ahmed Elgammal, Kostas Bekris and Dinggang Shen for their time, valuable feedbacks and suggestions.

I had wonderful colleagues during the last five year. First, I would like to mention Chao Chen. I started to work with him after he joined the CBIM when I was in my third year. In two year time, he has made clever and significant contributions on topics of this dissertation. Chao is the source of many anecdotes and discussions that I can remember through out my last two years. Erhan Bas from GE Global Research, has been a great colleague starting from the internship days at GE. After internship, he joined HHMI and we have always been in touch having enlightening phone meetings and discussions. He deserves special recognition for introducing me very interesting research topics and being a great co-author for a paper that is also included as part of this dissertation. I thank both of them for not only improving my work, but also improving my life. Shaoting Zhang from UNC Charlotte deserves special thanks, for his supports and friendship starting from the day I've joined CBIM. Without his helps, this thesis will be incomplete.

My years at Rutgers, in New Brunswick, was made enjoyable by my friends most of whom have been in US for an important milestone in their life. This meant a lot to us and we knew we have been fortunate since we were all passing through similar challenges and were getting support from each other that is priceless for all of us. From now on we will pursue our goals in different places but they will always be in my life.

And of course family. My parents, Fatma and Abdullah have always been thinking about me regardless of time difference and distance between us. My beloved sister, Gonul has been my best doctor and advisor during hard times despite the distances. Seeing the growing of my little niece Eylul during this five years has been enjoyable and illuminating a phd student's gloomy life. My beloved parents in-love, Melek and Mustafa have always encouraged me. Barkin, Elif and Eren have been the colorful part of my life inside the computer screen. They were always there unconditionally. Finally, I would like to thank my wife, love, Gizem, for being on my side all these years. We both have been continuing our phd study and have passed through similar difficulties. I thank her for supporting me unconditionally, being not only my best friend and a mentor as well.

Dedication

To my parents Fatma, Abdullah, grandmother Neriman and love Gizem.

Table of Contents

Abstract . . . ii

Preface . . . iv

Acknowledgements . . . v

Dedication . . . vii

List of Tables . . . xi

List of Figures . . . xii

1. Introduction . . . 1

1.1. Image Segmentation . . . 1

1.2. Segmentation via Energy Optimizing . . . 3

1.3. Segmentation Applications and their Requirements . . . 5

1.4. Contributions of this Dissertation . . . 6

1.5. Organization . . . 8

2. Review on Deformable Models and Graphical Models in Computer Vision . . . 9

2.1. Energy Minimization in Image Segmentation . . . 9

2.2. Deformable Models and Their Use in Medical Image Processing . . . 11

2.2.1. Model Representation . . . 12

Explicit Landmark Representation . . . 12

Implicit Level Set Representation . . . 13

2.2.2. Designing External Energy Term for Deformable Models . . . 15

Boundary Constrained Deformable Models . . . 15

Region Constrained Deformable Models . . . 16

Shape Prior Constrained Deformable Models . . . 18

2.3. Graphical Models 20

2.3.1. Undirected Graphical Models for Image Segmentation 23

Markov Random Fields 24

Conditional Random Fields 25

2.3.2. Inference and Learning for Random Fields 25

Sum-Product Belief Propagation 26

Graph Cuts 28

Parameter Learning for Conditional Random Fields 29

2.4. Comparison of Graphical Models & Deformable Models 30

3. Collaborative Multi Organ Segmentation by Integrating DM & GM 32

3.1. Introduction 32

3.2. Motivation 33

3.3. Related Work 34

3.4. Integrating Deformable Models and Graph Cuts 35

3.4.1. Alternating Energy Minimization Scheme 37

3.4.2. Implementation Detail 38

3.5. Experiment & Results 41

3.5.1. Knee Joint Bone Segmentation 41

3.5.2. Cardiac Segmentation 42

3.6. Conclusions 44

4. Review on Neuron Reconstruction from Electron Microscopy Images 45

4.1. Introduction 45

4.2. Challenges in Neuron Segmentation and Reconstruction 46

4.3. Previous Work 48

5. Multi-Stage Decision Strategy for Neuron Segmentation 50

5.1. Introduction 50

5.2. Related Work 52

5.3. Membrane Detection and Identification 53

5.4. Experiment & Results 59

5.5. Conclusions 60

6. OpTree: Learning Adaptive Watersheds for Neuron Segmentation 61

6.1. Introduction 61

6.2. Related Work 63

6.3. Hierarchical Tree-Derived Segmentation 66

6.4. CRF Inference and Training. 68

6.5. Computing the Optimal Tree-derived Segmentation. 70

6.6. Feature Types and Their Relation To Segmentation Tree 71

6.7. Experiment & Results 73

6.8. Active Proofreading for Accurate Neuron Reconstruction 75

6.9. Experiment & Results 82

6.10. Conclusion 83

7. Conclusions 85

References 87

List of Tables

3.1. Quantitative comparisons of our method with two relevant approaches, 1) Metamorphs [1] which integrates texture information into deformable models, and 2) a graphical model coupling MRFs and deformable models [2]. We reported the mean and standard deviation of voxel distances between segmented surfaces and ground truth, and volume overlap errors in proportions for 23 MRI and 15 CT scans. 42

6.1. Features. 72

6.2. ISBI 2012 Competition Results 74

6.3. Baseline comparison. 75

List of Figures

1.1. (a) Example for boundary delineation of brain structures from brain MR image. Notice the variation of contrast for different structures. Red curve captures the boundary of the high contrast object at all locations; while the green curve cannot match the ground truth (blue) everywhere. (b) Electron microscopy image showing a difficult scenario for partitioning the image into meaningful regions (neurons). Notice the dark membranes separating cell regions. (c) Detection of membranes (white) is challenging due to low contrast, diffused pixel locations and high texture variation. Also notice other entities (inside the cells) having similar appearance with the membranes. (d) A nice partitioning to segment neurons accurately. 3

2.1. A deforming model on a synthetic noisy image. In (a) external forces applied on the model are shown with their magnitude and direction. In (b) a model is shown after applying external and internal forces. Notice here, how internal forces preserve the smoothness and curvature of the model although external forces in (a) tries to drive the model towards edges. Image resource: [3] 10

2.2. Example shows segmenting a foreground object from (a) color image; using (b) only unary term of Eq. 2.2; (c) both unary and pairwise terms. Image source: [4] 11

2.3. Two different model representations: (a) Explicit, landmark based; (b) Implicit level-set based. Notice the how the iso level of a three dimensional function on the right, corresponds to the two dimensional contours on the left. Image source: [5] 12

2.4. Example graphical models representing joint distribution of three random variables. Notice the differences on the links: (a) directed; (b) undirected 20

2.5. (a) A directed graphical model representing joint probability distribution of three random variables; (b) An undirected graphical model representing joint probability distribution of three random variables. Note here that, $\mathcal{E}$ is enumeration of possible cliques in this graph: $\mathcal{E} \rightarrow 2^{\{x_1,x_2,x_3,x_4\}}$. 22

2.6. (a) A fully connected undirected graph. (b) and (c) Two factor graphs each of which corresponds to the undirected graph in (a). 22

2.7. (a) Two Bayesian networks showing conditional independence of variables $x_1 \perp x_2 \mid x_3$; (b) An undirected graphical model representing the conditional independence of random variables $x_1 \perp x_2 \mid x_3. The illustrations are adopted from [6]$ 23

2.8. A simple MRF model that captures Markovian assumptions for image labeling problem. Y_i denotes observation and X_i is a random variable denoting state of the pixel i. 24

2.9. Message computation on tree structured graphical models for message passing algorithm. The illustrations are adopted from [6] 27

2.10. Comparison of Graphical Models and Deformable Models. 31

3.1. Top: binary graphical model (left) would produce wrong regions for either labels, but multi-label graphical model (right) is correct; Bottom: comparison of combining DM with binary GM (left) and multi-label GM (right). 34

3.2. Iterative, model constrained estimation of labels. (a) User provided model initializations and background cues (strokes in magenta). (b-d) multi-phase graph cut results and states of the deformable models at consecutive iterations. In top row, gray color represents the label for blue model (left ventricle) and black color represents the label for green model (right ventricle), white label represents the background. In bottom row, deformable models at consecutive iterations are shown. (e) Plot of E_{ext} energy values computed at each iterations. . . 37

3.3. Label unification and its effect to the segmentation: (a) Example kernel density estimate of the regions underneath the models. (b)Top row MAP-MRF labeling without unification (4 foreground labels). Bottom row, MAP-MRF labeling with unification (2 foreground labels). (c)-(d) Signed distance maps for two different models with (bottom) and without (top) unification.(e) Resulting segmentations obtained w/o unification. 39

3.4. Experiment on knee joint MR image. (a) Three model initializations and background cues (strokes) for segmentation of tibia, femur and patella. (b) Result of [1]. (c) Result of [2]. (d) Result of our method. 40

3.5. Quantitative comparisons on the heart CT image. (a) Four model initializations and background cues (strokes) for segmentation of Right Atrium, Right Ventricle, Left Ventricle and Left Atrium. (b) Results of [1]. (c) Results of [2]. (d) Results of our method. 43

3.6. 3D visualization of results using our method; (a) Cardiac four chamber data. (b) Knee joint bones. 44

4.1. Sample image patches illustrating the challenges of detection and reconstruction of neurons from electron microscopy images. (a) Weak, cluttered or even no local membrane evidence at the locations pointed by blue arrows; (b) Complex appearance inside the cytoplasm of neuron cells; (c) other organelles (i.e., mitochondria (red) and vesicles (green)) having membranes with similar appearance features; (d) High variation in the shape of neuron structures. 47

5.1. Illustration of a random forest classifier performance on membrane prediction. Training is based on two classes: membrane vs. non-membrane. (a) Input image patch; (b) prediction result. Bright color means higher probability for being membrane. 51

5.2. The block diagram of our workflow with its training and test parts. 53

5.3. (a) Result of 1st stage random forest classifier on a test image. (b) A closer look to the detected point locations on the ridge of the cell membrane probability map. Note that the probability map is just the red channel of image in (a). Unit vectors in the tangent space (red) and the vectors (green) that are orthogonal to the tangents are also rendered. 55

5.4. Illustration of feature vectors for the three classes: L_1 (1) partition cytoplasm into distinct clusters; L_2 (3, 5) samples are generally observed in uncertainty regions (dense blue); L_2 (2, 4) samples can be found between cytoplasm and synapse/mitochondria transition. 56

5.5. (a) A classification result via second-stage random forest classifier based on the features shown in Fig. 5.4. (b) Projection points with their label colormap. Points classified as cell boundary are shown in blue and the others in yellow. Red points are discarded since they have no valid connection to other points on the constrained graph. (c) Projection points rendered with their initial affinity colormap. (d) Projection points rendered with their adjusted affinities. 57

5.6. (a) Initial set of nodes that are chosen above a very large affinity threshold. Note that points are rendered with affinity colormap. (b) Final cytoplasmic boundary points obtained after region growing on the sparse graph. 59

5.7. Final segmentation results for an example section from the test set. 59

5.8. The visual summary of our framework. 60

6.1. (a) The EM image patch; (b) the ground truth; (c) the boundary likelihood map (dark pixels have high values); (d) the watershed segmentation and its tree, built using the boundary likelihood map as the landscape function; (e) the watershed segmentation with a higher threshold; (f) the result of our algorithm. 62

6.2. (a) Constraints for segmentation labelings: The root node has infinite potential for label 1. The leaf nodes have infinite potential for label -1 since it only has labels 0 and 1 feasible. For any internal node s_i, and its child s_k in T, the edge (s_i, s_k) has only $(-1,-1)$, $(-1,0)$, $(1,1)$ and $(0,1)$ as feasible label combinations. So, an edge potential $\psi_{i,k}(y_i, y_k)$ is infinite for $\{(y_i = 0, y_k = 0), (y_i = 0, y_k = -1), (y_i = -1, y_k = 1), (y_i = 1, y_k = -1), (y_i = 1, y_k = 0)\}$ label configurations. (b) We find a tree derived segmentation that best approximates the ground truth segmentation $\widehat{S}$. 69

6.3. Boundary features. (a): external boundaries (red) and internal boundaries (blue), illustrated in 2D; (b) an example normal line segment at a boundary point (purple line), its profile is extracted using the likelihood map along the line (also shown); (c) normal lines of the whole external boundary, different colors correspond to boundary patches bewtween the node and different neighbors. . . . 72

6.4. (a):Intensity image; (b) probability map obtained by a classifier; (c) initial superpixels obtained by applying watershed transform on probability map shown in (b); (d) our result 76

6.5. The flowchart of the interactive framework. 78

6.6. Boundaries a user have to proofread (blue) in a classical framework. Our system only highlights a small subset based on the marginals. Cyan boundaries are the candidates to be marked as split and yellow boundaries are the candidates to be markes as merge. 78

6.7. Transfer of node labels and marginals to internal boundary labeling and marginals. Dashed line represents a *merge* label while solid line is a *split*. 79

6.8. (a) in an example segmentation, there are 16 boundaries in the watershed transform, including 8 split (solid) and 8 merged (dashed); (b) a tree (nodes on the curved paths are not shown for clarity); (c) b is labeled merged; (d) b is labeled split. 79

6.9. (a): our system only show boundaries with low marginals (yellow if labeled split, cyan if labeled merged); (b): before user input; (c): after user input, many boundaries are fixed. 83

6.10. (a) Interaction Simulation; (b) 3D Result. 84

Chapter 1

Introduction

This dissertation presents new methods for medical image segmentation. The applications covered in this dissertation are rather broad, ranging from fully automatic to semi-automatic image segmentation methods at different imaging modalities, scales and dimensions. In practice, each application has different requirements and difficulties motivating use of different approaches fine tuned to the problem. In general, unfortunately, a segmentation method solving a particular problem cannot solve another problem right away even if the problems look similar. This actually motivates researchers and engineers to innovate new methods and techniques because it is generally the first step in any automatic analysis or interpretation of an image. Therefore, for each application that we focus in this dissertation, we analyze the segmentation task and the requirements carefully and propose new methods that brings certain advantages towards accuracy, tractability and efficiency.

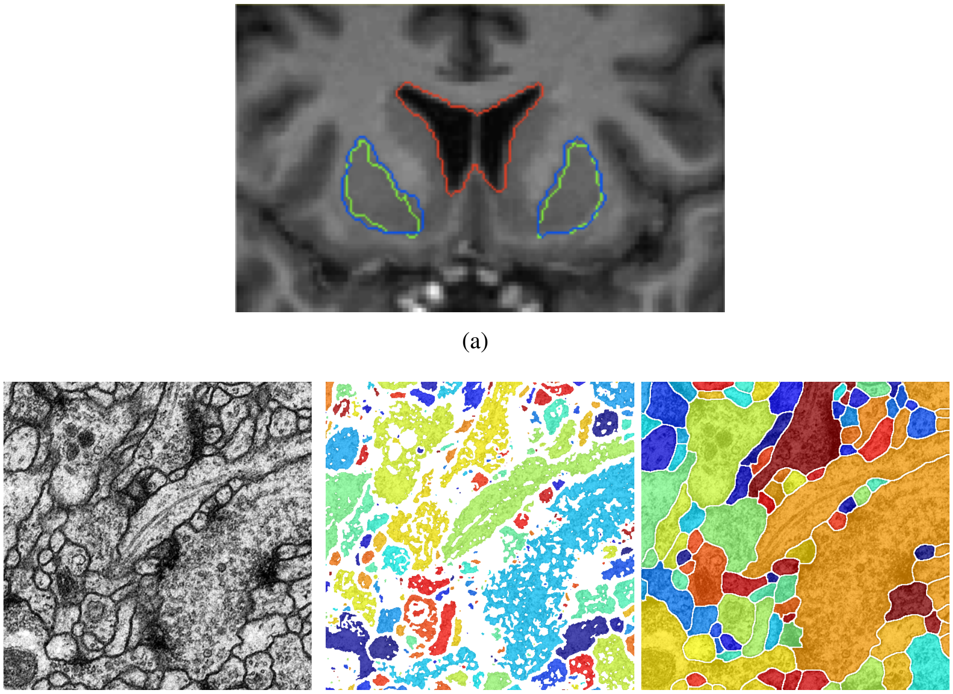
We present new methods for 3D organ segmentation from MR and CT images (macroscopic), 2D cellular region segmentation and 3D neuron structure extraction from EM images (microscopic). The purpose of this chapter is to introduce the problems addressed in this dissertation and provide a brief description of the main contributions and an outline of the document.

1.1 Image Segmentation

In this dissertation, we focus on two main principles to formulate an image segmentation task. Our first segmentation task relies on delineation of the boundaries of single/multiple foreground object(s). It is done by finding a set of boundaries/edges that separates foreground object from the background. Here, an edge is a surface/contour with a measurable area/length where the nearby pixels change significantly implying the discontinuity of image intensity values. This type of image segmentation targets finding edges and local discontinuities; however many advanced methods in practice don't limit themselves to consider only edge information. Many real world image segmentation problems imply challenges such as low-contrast, diffused boundaries, intensity inhomogeneity and high texture variation inside the target object. Fig. 1.1(a) shows a scenario for segmenting brain structures from MR images and as clearly noticeable, some structures are almost impossible to identify by considering only edge information. In the presence of these difficulties, use of other high-level image features such as intensity statistics, shape information of the target object, texture information has been very popular. When segmenting multiple objects simultaneously even more complex information such as relative position has been shown to be very useful to augment the results. However, all these approaches come with their own consequences and there are always trade-offs that have to be considered in terms of maintanenance, applicability, accuracy and speed.

The second segmentation task that we focus in this thesis can be realized by partitioning the image domain into multiple regions. In this problem, image regions are labelled according to low level and high level observations (features) and regions with different labels correspond to distinct objects with connected pixels. In a simple scenario, pixels with the same label should share similar characteristics like color, intensity, or texture while being discriminative among different labelled regions. But, sometimes this is not a good assumption to make in practice. For example, neuron cells on EM images as shown in Fig. 1.1(b) are separated with ambiguous dark membranes and present similar intensity and texture properties. In this scenario, every single cell should be labelled differently considering the membranes. However, in many cases, the observed indicators (e.g dark pixels) are noisy and not easily distinguishable. As can be seen in Fig. 1.1(b), membranes at some locations look diffused and their contrast is not good enough to distinguish nearby cells (see a bad partitioning result in Fig. 1.1(c)). Moreover, there are other entities having similar properties. In such scenarios, in addition to the local relationships of the pixels, use of global and local higher-order relationships has been shown to improve the accuracy significantly (see a very good partitioning result in Fig. 1.1(d)).

The key to a successful segmentation application is to use/develop the right method for the right application. In the literature, there are two principal approaches commonly used to



(a)

(b) (c) (d)

Figure 1.1: (a) Example for boundary delineation of brain structures from brain MR image. Notice the variation of contrast for different structures. Red curve captures the boundary of the high contrast object at all locations; while the green curve cannot match the ground truth (blue) everywhere. (b) Electron microscopy image showing a difficult scenario for partitioning the image into meaningful regions (neurons). Notice the dark membranes separating cell regions. (c) Detection of membranes (white) is challenging due to low contrast, diffused pixel locations and high texture variation. Also notice other entities (inside the cells) having similar appearance with the membranes. (d) A nice partitioning to segment neurons accurately.

approach these two different segmentation tasks. They are both formulated as energy optimization problems. A gradient descent local optimization method called Deformable Models has taken great attention in solving the first segmentation task. For the second task, graph based global optimization techniques called Graphical Models have taken huge attention to solve the labeling problem in a probabilistic framework. In the next section, these energy optimization based approaches will be explained in more detail.

1.2 Segmentation via Energy Optimizing

Optimizing an energy function generally means finding the values at which a function reaches its minimum or maximum value. Energy minimization/maximization has been a key tool in image segmentation in the last twenty years. This increasing popularity has mainly been the result of the successes of Graphical Models and Deformable Models. Deformable Models find locally optimal segmentation results by deforming an initial physical model (contour/surface). The model is the optimizer of an energy/cost functional. Graphical Models find globally or near globally optimal results by solving a graph node labeling problem using graph theory and probability theory. The discrete labeling configuration is the minimizer of the cost/energy function. Both categories are able to achieve satisfactory results; however, each model has its own strengths and weaknesses.

Deformable Models can capture the details of the target object very well if the initialization is good. First a model is initialized on the image domain, then several driving forces are applied physically on the model to deform it towards the desired object boundary. The resulting model is deformed iteratively into the final shape and position via combination of several external forces computed from the image. The external forces are regularized with internal constraints to preserve smoothness. These internal forces are computed from the model itself. The balance between different forces determine the behavior of the model. In an ideal scenario, the model can converge to a subpixel accurate, natural looking result. However, in general, images are noisy, finding the best initialization is difficult, and fine tuning effects of external forces is not very practical. Thus, in practice Deformable Models are usually combined with other methods which can find good initialization, provide hard constraints to get out of a local minima.

Graphical Models provide probabilistic frameworks to formulate the discrete labeling problem. In graphical models, graphical representation is an indispensable tool, that allows manipulations on the probability distribution over random variables and encodes conditional dependencies and independencies. This flexibility permits factorization of joint probabilities into simpler formulations such as unary and pairwise terms. Here, the pairwise term preserves the smoothness of the labeling configuration so that neighboring nodes tend to get same label. The unary term encodes the likelihood of each node to get certain class labels. In general, minimization of such an energy functional to find the best labeling configuration is NP-complete or NP-hard. Only under certain circumstances (suitable graph structure or cost functions), this optimization problem can be solved in polynomial time, otherwise approximated. These restrictions makes

Graphical Models prone to segmentation errors. Depending on the chosen parameters the globally optimal result is not guaranteed to be a good segmentation. For example, a very common observed undesirable effect in practice is called shrinking bias that is caused because of the minimization scheme. Also since it is discrete labeling problem based on the connectivity of the graph structure, they can exhibit blockiness artifacts. Also, as in Deformable Models case, the finding the right balancing factor between unary and pairwise terms in the energy functional is still a problem which determines the behavior of the model.

1.3 Segmentation Applications and their Requirements

In organ segmentation task, important point is to achieve accurate, detailed, smooth and natural looking results. Deformable Models represent the segmentation results as real physical objects (model) in 2D/3D space. So they have taken huge attention to meet the requirements of organ segmentation from MR and CT images. Besides their certain advantages, if the initialization is not good enough, boundaries of the target object is not clear or there is intensity inhomogeneity, then naive Deformable Model techniques would not be satisfactory. In such scenarios, use of statistical training information such as shape, relative positioning and appearance of target object is very useful if training data is available. In organ segmentation applications, creating and maintaining training sets is possible since organs mostly present similar properties in a particular image modality. However, note that, training sets are not always available and maintaining large sets of accurate training data is not always feasible.

The requirements of neuron structure segmentation from Electron Microscopy images is different than organ segmentation. Here, the aim is to find groups of pixels (regions) regardless of their shape and size such that these groupings correspond to neuron cells that are separated by their membranes. In other words, the 2D/3D image has to be decomposed into multiple regions that corresponds to neuron cells. For this type of segmentation task, Graphical Models are better choice due to the fact that, they are able to produce consistent labelings by considering local judgements of similarity and global consistency of the regions/pixels. Their efficiency can be improved even further by using superpixel generation methods as pre-processing step. A superpixel representation greatly reduces the number of nodes in the graph structure by replacing the rigid structure of the pixel grid. Superpixel generation algorithms rely upon spatial location and color/texture distribution of the pixels. Therefore, the features/observations obtained from superpixels provide better cues for the inference task. They can robustness of the model. This improvement, makes the use of Graphical Models even more effective for partitioning of large, high resolution images.

A reliable and accurate segmentation of an image is, in general, very difficult to achieve by purely automatic means. Also, there is great deal of mistrust from expert users towards fully automatic image analysis. That's why, interactive segmentation is an important paradigm in image processing. Ideally, such an interactive system should be designed as to enable a user to create excellent segmentation results with a minimal amount of time and effort. The most basic interactive segmentation methods consist of a drawing application which the user can use to segment an image by labeling the individual image parts. But, this basic approach is slow and tedious; thus, there is great demand for algorithms that help the user in segmenting image data. These algorithms should speed up the segmentation process relying only on sparse annotations instead of the dense image labeling. In fact, such functionality would be tremendously useful in life sciences where biologists need to extract and measure objects of interest from microscopic images.

1.4 Contributions of this Dissertation

The first contribution of this dissertation is combining graphical models and deformable models into a collaborative framework for multi organ segmentation for cardiac and knee joint bone segmentation from CT and MR images, respectively. We present a hybrid energy minimization framework where the deformable models and graphical models are combined in a collaborative way. In an alternating iterative optimization scheme, two different models are improved at each step towards more accurate segmentation. Given an initial deforming model (surface/contour), the parameters of the GM is learned and constrained in a data driven approach. This initialization actually corresponds to providing initial cues as in classical graph-cut methods [7]. The deforming model is deformed according to the attraction force computed based on the GM inference. The overall framework takes advantage of these two segmentation paradigms. The final results are smooth, natural looking and can capture details while the optimization scheme is enhanced with globalization aspect of the GM. The proposed hybrid method is designed to satisfy the following requirements: 1) it can efficiently and effectively segment multiple objects simultaneously, 2) the resulting segmentations (contours/surfaces) of multiple objects don't intersect, 3) the iterative model deformation mechanism is robust to local high and low contrast challenges in cardiac CT images and knee MR images.

The second contribution of this dissertation is to design a multi stage decision strategy for neuron segmentation from ssT electron microscopy images [8]. We propose a two step classification mechanism to identify cell membranes between adjacent cells. The first step utilizes a random forest classifier to predict pixel wise membrane class label with lots of false positives. The second step utilizes the underlying differential geometric properties of cell membranes in a biologically inherited framework. This unsupervised technique involves extracting highly sparse low dimensional representation of the data and characterization of this representation in a biological context. We solve this contextual characterization problem using a graph based local optimization scheme. Once the membranes are identified the cells can be separated well despite the diffused and confusing membrane appearance.

The third contribution of this dissertation is to create a tree structured CRF model to segment neuron structures from 2D ssTEM [8] and 3D FIBSEM [9] images in a learning fashion. We use a region (*superpixel*) based approach where the relationships between the regions are encoded in a hierarchical tree and this tree is actually the graph of the graphical model. We employ efficient and exact belief propagation method to solve the inference problem. We design effective features per regions that help improve the inference accuracy. Because our model is tree structured, the inference time is very short even for large 3D images. Also, we show that the accuracy of the results are comparable to the state of the art.

Finally, we propose a method to extend the tree structured CRF model into an active *proofreading* mechanism using the marginals of the graphical model as the confidence level of the segmentation. Since the computation of marginals on a tree structured graph is very efficient and exact, we transferred node marginals to boundary marginals and highlighted the least confident boundary locations to ask user input. We designed a tree update schemes to take the user actions (i.e split and merge) and update the structure of the tree. With this feedback mechanism, we provide an interface where the user can reach globally improved accurate results with very few efforts compared to other baseline approaches.

1.5 Organization

This dissertation is organized as follows. **Chapter 2** reviews the two well studied segmentation frameworks Deformable Models and Graphical Models. In the review of Deformable Models, we categorize them according the terms used in energy definition as per the improvements upon several image segmentation challenges. In the review of Graphical Models, we get into details of two energy optimization schemes that are specified for loopy graphs and acyclic graphs. We review their essentials and relate them to image segmentation problem. At the end, we provide a summary and comparison of Graphical Models with Deformable Models, as well.

Chapter 3 introduces the main theory of the proposed collaborative multi object segmentation framework, evaluates its performance on cardiac and knee joint segmentation and discusses its limitations and potential extensions. The previous version of this chapter appeared as [10].

Chapter 4 reviews neuron structure segmentation and reconstruction from electron microscopy images. We introduce recent developments for nanometer scale neuron reconstruction in the neuroscience field, then, we review its challenges from image processing and machine learning perspective and finally, we examine the existing machine learning and vision based approaches.

Chapter 5 and 6, introduce our proposals for image partitioning problem towards neuron segmentation. In **Chapter 5**, we propose a multi stage decision mechanism which is inherited from a contextual grouping idea. We present its performance on ssTEM images, illustrate the results and discuss its limitations. The previous version of this chapter appeared as [11]. In **Chapter 6**, a region-based hierarchical CRF model for image partitioning is explained. We give its theoretical aspects and evaluate the results on 2D ssTEM and 3D FIBSEM images. We discuss its limitations and further extensions. Last but not least, we explain a active *proofreading* method by giving the details of the framework and comparing its results to the state of the art. The previous version of this chapter appeared as [12]. Finally, Chapter 7 summarizes this thesis work and discusses the possible directions for future research.

Chapter 2

Review on Deformable Models and Graphical Models in Computer Vision

In this chapter, we provide background material relevant to the development of the material in later chapters. First, we provide background information on energy minimization based segmentation methods. We start by formulating cost functions whose minimum value corresponds to segmentation. In particular, we review two different function minimization approaches for continuous and discrete variables. We give the details of Deformable Models and Graphical Models with an emphasis on local and global optimization techniques. We also compare these two approaches analyzing their advantages and shortcomings within segmentation applications.

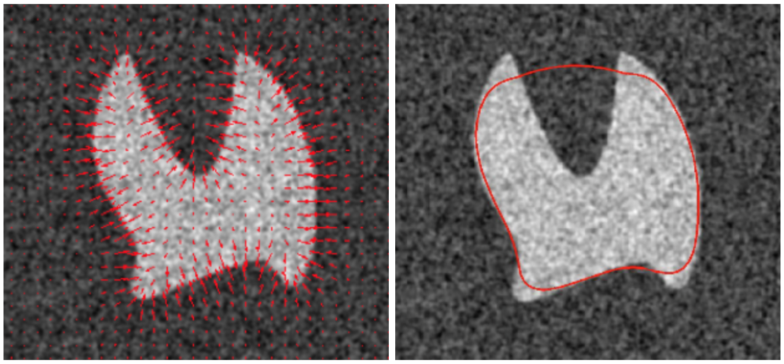
2.1 Energy Minimization in Image Segmentation

In energy optimization based segmentation problems, energy function is usually defined in terms of variables which represent the segmentation like a contour or pixel label. These variables are the actual minimizer/maximizer of the cost function. They encode a desired property of the image and can be continuous or discrete. For the case of continuous variables, these problems are commonly referred to as deforming an initial continuos contour/surface iteratively to snap locally on the target boundary. The generic name for this framework is called Deformable Models.

The energy functional for Deformable Models always consists in the sum of two terms:

$$C^* = \underset{C}{\operatorname{argmin}}\, E_{int}(C) + E_{ext}(C) \tag{2.1}$$

Here, the internal energy E_{int} serves to impose a smoothness constraint. External energy E_{ext} depends on the features which are searched for in the image. Its purpose is to push/pull



(a) (b)

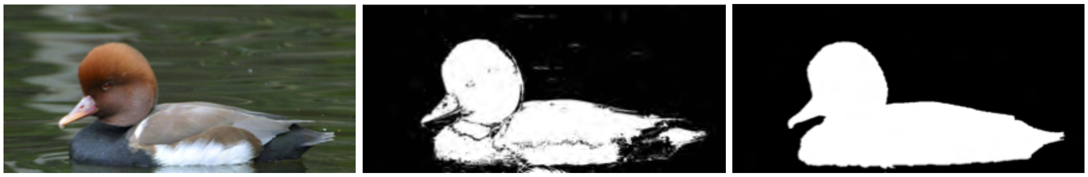
Figure 2.1: A deforming model on a synthetic noisy image. In (a) external forces applied on the model are shown with their magnitude and direction. In (b) a model is shown after applying external and internal forces. Notice here, how internal forces preserve the smoothness and curvature of the model although external forces in (a) tries to drive the model towards edges. Image resource: [3]

the deformable model towards the salient image features (i.e. termination of line segments, edges, etc.) [13]. Fig. 2.1 illustrates the effect of external and internal forces on a synthetic image. The moving model tries to minimize this total energy in a sense that the external energy is regularized with internal energy while detecting the edges. This is an ill posed problem and minimum of this energy can be obtained by gradient descent optimization. This optimization scheme requires the initialization of the variable C and its update in the steepest descent direction iteratively.

For the case of discrete variables, these problems are commonly referred to as labeling problem as they involve assigning the best set of labels to some hidden variables. Graphical Models combine probability theory with graph theory, the energy function which represents a probability distribution is minimized using graph based algorithms. The total energy functional for graphical models is usually composed of two main terms: unary and pairwise. The definition of the function can be given as following:

$$y^* = \underset{y}{\operatorname{argmax}}\, E_{unary}(x, y) + E_{pair}(y_i, y_j), i, j \in \mathcal{N} \tag{2.2}$$

According the above formula, the estimate y^* will trade off the of the unary term and pairwise term making decisions so that the prediction based on unary are also spatially consistent. The effects of unary term and pairwise term are illustrated in Fig. 2.2(b) and (c). In general, for



(a) (b) (c)

Figure 2.2: Example shows segmenting a foreground object from (a) color image; using (b) only unary term of Eq. 2.2; (c) both unary and pairwise terms. Image source: [4]

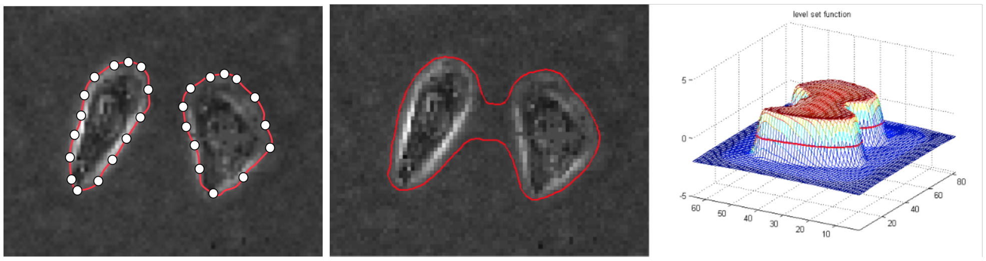
arbitrary unary and pairwise terms, solving for y^* exactly, yields a hard optimization problem. We will see in the following sections that to solve this problem exactly there will be certain restrictions for the definition of E_{pair} term and the label space. Also we will see that there are powerful ways of relaxing these constraints and obtaining approximate solutions for y^*.

In the following sections we will get into more theoretical details for these two different energy optimization based segmentation frameworks and analyze them individually.

2.2 Deformable Models and Their Use in Medical Image Processing

Deformable models pioneered by Kass et al. [13], are curves or surfaces that move under the influence of internal and external image forces as formulated in Eq. 2.1. The evolution process corresponds to an iterative optimization of a variational cost functional. The approach in [13] represents the boundary between regions as a closed contour and the contour is evolved until it converges to the boundaries of objects.

There are several different types of Deformable Models methods. The classification can be done in two ways: the way a model is represented; the way the external term is designed. There are two different ways of representing a model: explicit landmark based and implicit level set based. We will get into the details of these representations soon. Beside the variety of model representation, the main focus of research in deformable models field is on the external force term $F(s)$ in 2.3 and 2.4. Compared to the graphical models, deformable models allow to use very complex structured energy terms due to the gradient descent optimization scheme. So far there have been five categories of external forces defined for deformable models: edge (boundary) based, region based, statistical prior based, learning based and spatial constraint based. We will show the details of these approaches and their use case in the following. Before



(a)

(b)

Figure 2.3: Two different model representations: (a) Explicit, landmark based; (b) Implicit level-set based. Notice the how the iso level of a three dimensional function on the right, corresponds to the two dimensional contours on the left. Image source: [5]

that, we start by reviewing different model representation methods.

2.2.1 Model Representation

There are two different ways of representing a model: explicit landmark based and implicit level set based. Fig. 2.3 illustrates them.

Explicit Landmark Representation

In explicit representation the contour or the surface is represented by a finite set of parameters, e.g. the spatial positions of points on the curve or surface. They are used to reconstruct the evolving contour/surface by connecting them with line segments [13]. One difficulty of this approach is keeping connectivity of the points on the surface which are very likely to change during the evolution. Another consideration is that, the discretization should be fine enough to reconstruct the surface (see Fig. 2.3(a)). Moreover, if the points come too close together, they may cross each other if the time step is not adjusted properly. A solution for such problems is to redistribute the points every few time steps, and add or remove points where this is necessary. However, this task becomes complicated, especially in three dimensions. A more serious problem arises in the presence of a change in the topology. Generally, the parametric approach is not capable of handling topology changes, unless special constraints are implemented for detecting possible splitting and merging of contours [14].

Representing the model parametrically, $v(s) = (x(s), y(s))$, the energy definition given in Eq. 2.1 is written more explicitly as:

$$E(C) = \int_0^1 \Big(\alpha \mid C^{'}(s) \mid^2 + \beta \mid C^{''}(s) \mid^2 + \gamma F(s)\Big) \tag{2.3}$$

where α, β, and γ are real positive constants to weight the smoothness constraints and image forces, and F is a external energy term which is based on image features. [13]. Here, s corresponds to image coordinates $(x(s), y(s))$ and $C(s)$ is the evolving curve parameterized with spatial coordinate s. The first-order term makes the model move like a membrane and second-order term makes it move like a thin plate. The internal energy of the model is increased with large values of α. In such case, it stretches more and more, whereas for small values of α the energy function becomes insensitive to the amount of stretch. Similarly, with large values of β the model gets smoother shapes, otherwise the energy becomes insensitive to curves in the model. When both α and β are small the constraints on the size and shape of the snake becomes less [13].

Implicit Level Set Representation

Level sets were first introduced by Osher and Sethian in 1988. They use an implicit representation of the contour to represent the boundaries.They are numerical techniques for tracking an evolving surface without describing the evolution of the contour itself. They operate on a function that is defined in one dimension higher and the model is described as the iso-contour of this function (see Fig. 2.3(b)). In order to model an evolving model, the level set function depends on time as well as space.

Let $\phi = \phi(s,t)$ be the level set function. Then the interface C, at any point in time t, is given as the set of points in space that corresponds to the zero level iso-contour of ϕ, i.e. $C(t) = x : \phi(s,t) = 0$. The level set function $\phi : \Omega \times [0, \infty] \rightarrow \Re$, is a scalar valued function of both space and time variables. Since we restrict our attention to the image segmentation problem, it is defined on the same rectangular domain as the image $\Omega \in \Re^n$.

The Euclidean distance transform is used to embed the shape of model as the zero level set of a distance function in the higher dimensional space. By definition Ω is bounded since it refers to the image domain and the model defines a partition of the image domain into three: region that is enclosed by the model C, interior region $\mathcal{R}_{int}$ and background region $\Omega - \mathcal{R}_{out}$.

$$\phi_C(s) = \begin{cases} -D(s,C) & \text{if } s \in \mathcal{R}_{int}, \\ 0 & \text{if } s \in C, \\ D(s,C) & \text{if } s \in \mathcal{R}_{out}. \end{cases}$$

where $D(s,C)$ refers to the minimum Euclidean distance between the image pixel location $s = (x,y)$ and the model C. It gets positive values if the point lies outside the region, and negative if inside the region. For points that lie on the initial interface, the distance is zero.

Level sets can handle topology changes inherently in any dimension where parametric models require change in representation for different dimensions. The level set method works equally well in any dimension. For these reasons, active contour models based on level set methods have received considerable attention. The works proposed by Caselles et al. [15] and Yezzi et al. [16] are the first examples of level set based deformable models. They are called Geodesic Active Contours and Geometric Active Contours, respectively. They drive the curve (iso level) in the normal direction of contour curvature attracted with a force function. The energy functional is defined as:

$$E(C) = \int_0^1 F \cdot \mid C^{'}(s) \mid ds \tag{2.4}$$

When minimizing the level set function, the front curve deforms along its normal direction $C^{''}$, and this speed is controlled by the speed function F. Depending on the choice of the F function the front of the level set is propagated at a velocity proportional to its curvature in the normal direction. For the details of the derivation, please see [15] and [16].

There are two concerns associated with the level set representation that need to be addressed. The main concern is in its computational complexity since it has one dimension higher than the original shape. This efficiency problem is solved by using only a narrow band around the zero level of the embedding space as the working domain for segmentation [17]. The other concern is the validity of the iso level contours after the model is updated. One solutions for this can be using reinitialization mechanism during deformations [18].

2.2.2 Designing External Energy Term for Deformable Models

There has been significant amount of effort in the literature to design the F function shown in 2.3 and 2.4. Different external forces have been proposed to improve the performance of deformable models. The initial development of deformable models was based on image gradient. Soon after, regional definitions got into the horizon. More advanced appearance models followed the simple intensity based models. Then, use of more global constraints such as statistical priors of appearance, shape and pose become popular due to robustness to occlusion and low contrast, and noise. More recently, online data driven approaches have shown to be more robust external driving forces for deformable models when training information is not available. In the following these categories will be summarized.

Boundary Constrained Deformable Models

For explicit model representation the external term E_{ext} in Eq. 2.3 is designed to attract the contour points toward edges, which are points of high intensity gradient. The energy functional is given by

$$E(C) = \alpha \int_0^1 \mid C^{'}(s) \mid^2 ds + \beta \int_0^1 \mid C^{''}(s) \mid^2 ds + \gamma \int_0^1 \mid \nabla I(C(s)) \mid ds \tag{2.5}$$

This approach is originally called Snakes and it requires a good initial curve for accurate segmentation [13]. The reliance on edge information makes the model sensitive to image noise and to various other image artifacts. For example, the contour may get stuck in local minima due to strong edges inside or outside the object's true boundary. To solve these problems, other deformable model types that are extensions of this initial framework have been proposed. One important of them targets to increase the attraction range of the original Snakes by using gradient vector flow [19]. This field is computed as a spatial diffusion of the gradient of an edge map derived from the image. This computation attracts the contour that is far from the object via diffuse forces.

The first level set based methods [15, 16] also use gradient based term. The formula shown in 2.4 can be written as:

$$\begin{aligned} E(C) &= \int_0^1 F \cdot \mid C^{'}(s) \mid ds \\ F &= g(\mid \nabla I(C(s)) \mid) \\ g(\nabla I) &= \frac{1}{\mid 1 + \nabla I \mid^2} \end{aligned} \tag{2.6}$$

Here, the speed function F depends on the image gradient ∇I and, it is positive in homogeneous regions and close to zero at edges. Hence the curve deforms at a velocity proportional to its curvature in homogeneous areas and stops at strong edges.

Region Constrained Deformable Models

To address the limitations of boundary based approaches, there have been significant efforts in the literature to integrate region information into deformable models. The difficulties in parametric models bring us to one of the main advantages of level set methods since they represent the inner and outer region of the model implicitly. This makes the use of level set based representation very suitable for designing a region constrained deformable model.

A well-known example for the region modeling cost function is the Mumford-Shah functional [20]. Mumford-Shah functional has emerged before the active contour framework. It defines the segmentation problem as follows: given an observed image u_0, find a decomposition Ω_i of Ω, where $\Omega \in \mathbb{R}$, such that the segmented image u varies smoothly within each Ω_i, and discontinuously across the boundaries of Ω_i. In [21], level set based deformable model solution for a simplified case of the Mumford-Shah functional have been proposed. The image is assumed to be formed of two regions of piecewise constant intensities of distinct values. In the energy functional, a fitting term is defined and this fitting term is composed of differences between mean intensity values and observed intensities inside and outside the segmenting curve.

$$\begin{aligned} E(C) &= \int_{R_{int}} \mid I(s) - c_{int} \mid^2 ds + \int_{R_{out}} \mid I(s) - c_{out} \mid^2 ds \\ &+ \gamma Length(C) + \mu Area(R_{int}) \end{aligned} \tag{2.7}$$

where C, c_{int} and c_{out} represent the mean intensities of interior and exterior regions, respectively, and $I(s)$ refers to the image intensity value at pixel s. This approach drives the model according to the first order intensity statistics assuming the homogeneity of spatial features. The advantage of this method compared to edge based method, is being more robust to noise since intensity statistics provide more global information compared to using only gradient information. Multi-phase version of this approach has been proposed by [22]. In [22], piecewise-smooth approximations of the Mumford-Shah functional are derived for multiple regions/models in a variational level set framework. The estimation of the region mean intensities and update of the level sets are computed as alternating steps in an iterative scheme.

In the above approach, the piecewise-constant or piecewise-smooth assumptions limit the applicability in segmenting objects whose interiors have textured appearance and/or complex multimodal intensity distributions. More recent approaches focus on more robust region models. In particular, they estimate intensity/texture statistics of the region inside a deformable model using parametric (e.g. Gaussian, Mixture-of- Gaussian) or nonparametric methods [1, 14]. Using these appearance models, the model is deformed such that the statistical coherence inside the model is preserved. The parametric appearance models, make assumptions on the intensity distribution of all pixels inside an object without any training stage. If a Gaussian distribution is assumed, the intensities of an object are parameterized by a mean intensity μ and a variance σ^2.

$$P(i \mid h) = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-(i-\mu)^2}{2\sigma^2} \tag{2.8}$$

where $i \in 0, 1, ..., 255$ denotes an intensity value, and h represents pixel intensities inside the object. If a Mixture-of-Gaussian assumption is made, the object intensities are parameterized by the means and variances of several component functions, each being a Gaussian.

Other than parametric models, use of nonparametric kernel density estimation [23] is also very popular [24]. It represents a generalization of the Mixture-of-Gaussian model, where it does not make assumptions about the number of modes in a distribution. One important advantage of the nonparametric representation is that it can be approximated directly from pixel intensities inside a deformable model, requiring no priori parameter knowledge [1], [14]. As the model deforms, its interior region (pixels) changes, hence the nonparametric intensity distribution gets updated automatically. This way, models can capture target regions which have complex multi-model intensity distributions. On the other hand, the nonparametric kernel density approximation can not account for the spatial correlation (i.e. context information) between neighboring pixels, nor the pattern of the basic building blocks such as texture elements. This limits their applicability in scenarios where the object has highly varying texture or very similar intensity distribution with the background.

It is also possible to learn the statistics of a known object a priori when training images are available. The learned prior model can be used to guide the detection and segmentation of the object. Common ways of learning statistical prior models listed as Principal Component Analysis (PCA) [25, 26], Kernel PCA [27], isometric feature mapping [28], and local linear embedding [29]. There are also supervised classification methods that can be used for modeling appearance. When enough training data with rich set of features and ground truth annotations is available, popular learning methods such as AdaBoost [30], Support Vector Machines [31] are Neural Networks [32] can be used. However, one big limitation of the learning based methods is in the laborious training data collection and annotation process.

Shape Prior Constrained Deformable Models

In deformable model methods, a penalty on the length of the segmenting curves can be thought as the simplest shape prior for the objects in the scene. However, in many applications, more information is available particularly, the shape of the objects. As in the case of appearance modeling, statistical modeling of shape variability can be effectively used to capture variations in the shape of an object of interest via offline learning [33, 26, 34].

Shape priors constrain the model deformation such that, the deform is toward object boundaries with in respect of characteristics of the object's shape. This probabilistically-encoded high-level shape knowledge is often more robust for image interpretation; but they require much effort because they need collection/annotation of training data, alignment/registration of training examples, and learning of statistics for samples using modeling tools [26, 35].

There are numerous deformable model methods that enforce constraints on the underlying shapes as in [36, 26]. In [36], the authors find a set of points across a set of training images to construct a statistical model of shape variation. In [26], PCA is used to capture the variability of shapes. These techniques can handle only unimodal, Gaussian-like shape densities. In addition to the parametric approaches, nonparametric shape densities learned from training shapes have also been proposed in [35, 37]. In those works, it is assumed that the training shapes are drawn from an unknown shape distribution and this distribution is estimated by extending a kernel density estimator to the space of shapes. They construct the energy functional in a Bayesian framework, using region statistics like mean and variance (likelihood), and estimated shape distribution term (prior).

Different strategies for the design of the external term of a deformable model help the minimization of the total energy together with the internal term (i.e. smoothness, area constraints). These methods target solving the local minima problem in the presence of spurious edges, noise, inhomogeneity and occlusion complementarily to prevent the model from leaking at boundary gaps or getting stuck at local neighborhood. When different terms are combined together, their weighting factor should be tuned carefully to get the best result.

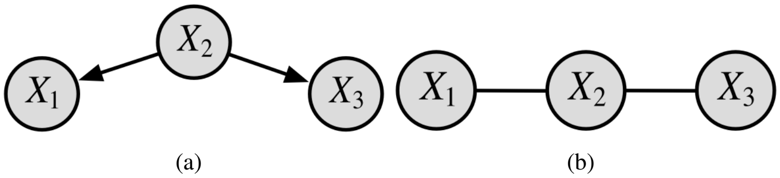


Figure 2.4: Example graphical models representing joint distribution of three random variables. Notice the differences on the links: (a) directed; (b) undirected

2.3 Graphical Models

Image segmentation using Graphical Models can be defined as exploiting statistical models to extract high level information from image. Image observations are related to solutions with some uncertainty and Graphical Models can provide a full probability distribution over all feasible solutions and the most probable solution could be computed using inference algorithms.

Graphical Models are represented by a set of observation variables and output variables [6],[4],[38]. They not only define the variables but also the interaction between these variables for which they provide elegant means.They encode joint or conditional probability distribution of these variables by means of a graph. A graph comprises nodes that are connected with edges. Nodes correspond to the random variables, and edges correspond to direct interactions between them (see Fig. 2.4 for simple graphical model structures). As known from graph theory, edges on a graph can exist in two form: directed and undirected. Depending on the type of the edges (links), Graphical Models can be named as *directed graphical models* and *undirected graphical models* (see Fig. reffig:graphicalModel for illustrations of directed and undirected graphical models). Directed graphical models are also known as *Bayesian Networks* and they are useful for expressing causal relationships between random variables. More information about directed models can be found in [39, 38]. Undirected graphical models are better suited to capture the mutual dependence between pair of nodes (random variables). The most common type of Graphical Models used for image processing are undirected graphical models [6, 4, 38]. The graph structure can encode the conditional dependence and independence among the random variables. First we explain the conditional dependency property of directed graphical models. Graphical models provide efficient representation for conditional distribution of multiple random variables with the directed links between them. Joint distribution of multiple random variables can be expressed as the product of factors that are defined only on a subset of the random variables. These factors are actually conditional distribution of the subset of random variables. The graph shown in Fig. 2.5(a) illustrate a directed graphical model that express conditional distributions among subsets of the random variables. Note that the direction of the links added to the graph are from the nodes (variables) on which the distribution is conditioned. The graphical model in Fig. 2.5(a) denotes that the joint distribution is given by the products (over all of the nodes of the graph) of conditional distribution of each node conditioned on the parents of that node in the graph. Thus for a graph with N nodes, the joint distribution is given by

$$p(X) = \prod_{n=1}^{N} p(x_n \mid pa_n) \tag{2.9}$$

where pa_n symbolize the set of parent nodes of x_n, and $X = \{x_1, x_2, \ldots, x_n\}$

The edges on undirected graphical models do not represent conditional distributions but arbitrary non-negative functions. These non-negative functions are defined over sets of variables that are local on the graph and they are actually the factorization of the joint distribution. The appropriate notion of this locality is defined by a a graphical concept called *clique* which is a subset of nodes that are fully connected. Thus the joint distribution is written as the product of functions of cliques.

$$p(X) = \frac{1}{Z} \prod_{\mathcal{C} \in \mathbf{C}} \phi_C(X_C) \tag{2.10}$$

where **C** is the set of cliques in the graph and Z is the partition function that normalizes the $p(X)$.

$$Z = \sum_{x \in X} \prod_{\mathcal{C} \in \mathbf{C}} \phi_C(X_C) \tag{2.11}$$

Note here that, there is no assumption on the functions $\phi_C(X_C)$ (also named as potentials) to have a specific probabilistic interpretation except $\phi_C(X_C) \geq 0$. This simple representation is called factorization, and it can be made explicit on the graph by introducing additional nodes for the factors in addition to the nodes representing the variables. This modified graph is called *Factor Graph* and Fig. 2.6 illustrates two different factorization of fully connected undirected

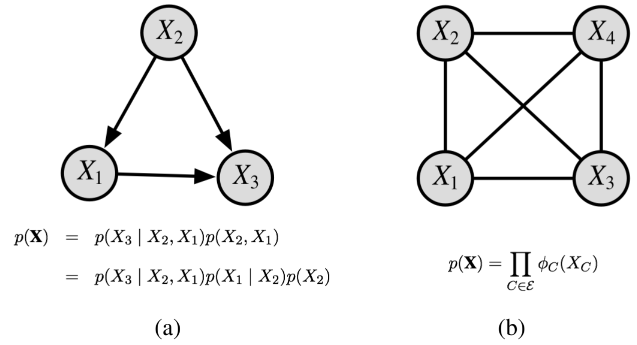

X_2
X_1
X_3
$p(\mathbf{X}) = p(X_3 \mid X_2, X_1)p(X_2, X_1)$
$= p(X_3 \mid X_2, X_1)p(X_1 \mid X_2)p(X_2)$
(a)
X_2
X_4
X_1
X_3
$p(\mathbf{X}) = \prod_{C \in \mathcal{E}} \phi_C(X_C)$
(b)

Figure 2.5: (a) A directed graphical model representing joint probability distribution of three random variables; (b) An undirected graphical model representing joint probability distribution of three random variables. Note here that, $\mathcal{E}$ is enumeration of possible cliques in this graph: $\mathcal{E} \to 2^{\{x_1,x_2,x_3,x_4\}}$.

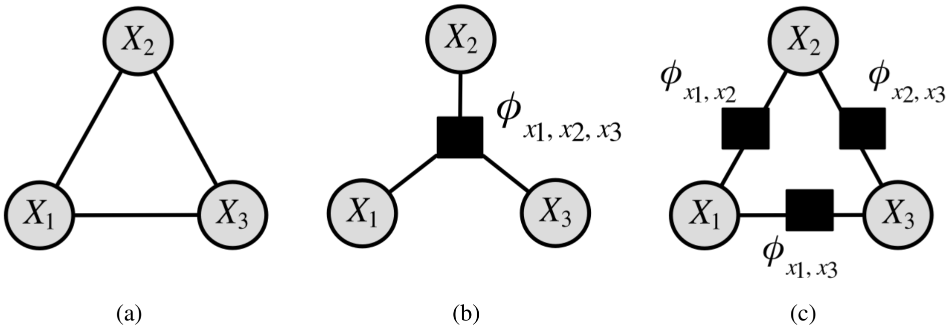
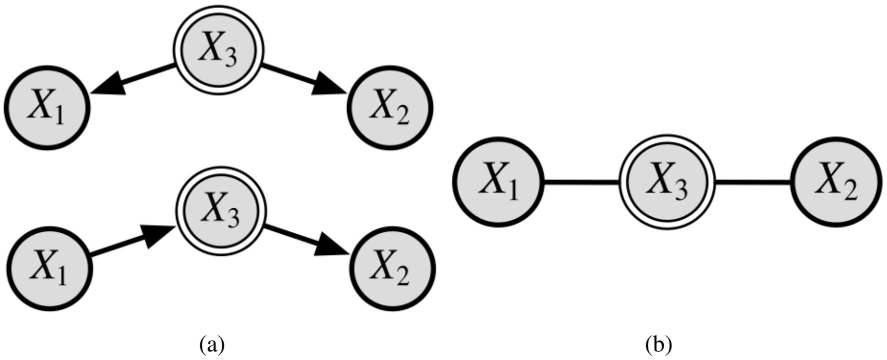


Figure 2.6: (a) A fully connected undirected graph. (b) and (c) Two factor graphs each of which corresponds to the undirected graph in (a).

graph. Note here that, factorization does not imply any conditional independence. We will focus on conditional independence later on. Also, we will see in Section **??**, there are principled algorithms that can exploit the factor graph structure to efficiently calculate the local marginal probabilities or the most probable labels over nodes (this is also called MAP estimation).

Like conditional distributions, conditional independence properties of multiple random variables play an important role. The probability distributions specified by graphical model differ by the conditional independence assumptions encoded in the graph. Consider three variables x_1, x_2, x_3 and suppose distribution of x_1 conditioned on x_2 and x_3 is independent of x_2:

$$p(x_1 \mid x_2, x_3) = p(x_1 \mid x_3) \tag{2.12}$$



(a)

(b)

Figure 2.7: (a) Two Bayesian networks showing conditional independence of variables $x_1 \perp x_2 \mid x_3$; (b) An undirected graphical model representing the conditional independence of random variables $x_1 \perp x_2 \mid x_3. The illustrations are adopted from [6]$

Using this, we can also write the joint distribution of x_1 and x_2 conditioned on x_2 as:

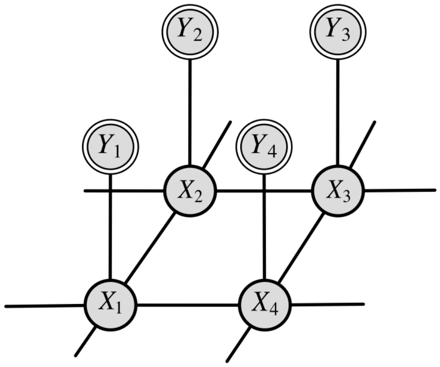
$$p(x_1, x_2 \mid x_3) = p(x_1 \mid x_2, x_3) \cdot p(x_2 \mid x_3) = p(x_1 \mid x_3) \cdot p(x_2 \mid x_3) \tag{2.13}$$

This says that given x_3, the variables x_1 and x_2 are statistically independent. This is represented with a directed graphical models as shown in Fig. 2.7(a).

Conditional independence property of random variables hold for undirected graphical models as well. Moreover, it is even possible to determine the independencies by graph separation operations. Basically, given the graph nodes $\mathcal{V}$ composed of sets if nodes $V_1, V_2, V_3 \subseteq \mathcal{V}$, if any path from a node in V_1 to a node in V_2 includes at least one node in V_3, then $x_1 \perp x_2 \mid x_3$. Fig. 2.7(b) illustrates an example for conditional independence of two random variables on an undirected graph. We will see in Section **??** that, the conditional independencies among random variables are actually the core of performing inference and learning in graphical models.

2.3.1 Undirected Graphical Models for Image Segmentation

Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) are two types of undirected graphical models that have been widely used for image processing.



(a)

Figure 2.8: A simple MRF model that captures Markovian assumptions for image labeling problem. Y_i denotes observation and X_i is a random variable denoting state of the pixel i.

Markov Random Fields

MRFs are generative models that models the joint probability of the observations and the labels. Holding the Markovian property, they model the interactions among random variables within a selected neighborhood system. In MRFs, image pixels are strongly related to the labels and due to the Markovian assumption local random variables are assumed to be conditionally independent of other random variables given the random variables in their neighborhood.These prior assumptions can be represented via undirected graphs using observed and hidden variables as shown in Fig. 2.8.

The hidden variables $\mathbf{X} = \{X_1, X_2, ..., X_m\}$ can take a value from the set of labels $L = \{l_1, l_2, ..., l_n\}$ and the variables Y_i correspond to the image pixels $i \in V = \{1, 2, ..., m\}$ values. The neighborhood system N is defined by the sets N_i, $\forall i \in V$, where N_i denotes the set of all neighbors of the variable x_i.

Based on the Hammersley-Clifford theorem [40, 41] if the potential functions are strictly positive, a MRF model can formulate a joint probability distribution as a Gibbs distribution which can be factorized into a log linear model.

$$\begin{aligned} p(\mathbf{X}, \mathbf{Y}) &= \frac{1}{Z} \exp^{-E(\mathbf{X}, \mathbf{Y})} \\ E(\mathbf{X}, \mathbf{Y}) &= \sum_{i \in V} \phi_i(X_i) + \sum_{i \in V, j \in N_i} \phi_i(X_i, X_j) \end{aligned} \tag{2.14}$$

Here, $\phi_i(X_i)$ is called unary potential since it is associated with only one variable while

$\phi_i(X_i, X_j)$ is called pairwise potential. N_i is the selected neighborhood and Z is a normalization constant (i.e. partition function), which can be calculated by marginalizing over all random variables in the MRF.

Conditional Random Fields

CRFs can be viewed as MRFs globally conditioned on the data. They are discriminative models that directly model the posteriori probability distribution of a set of random variables given the observations. The conditional distribution $P(\mathbf{X} \mid \mathbf{Y})$ over the labelings is a *Gibss* distribution can can be written exactly in the same form of MRFs with a slight difference by conditioning each random variable upon a set of global observations. The potential functions $\phi_i(.)$ map both clique assignments and observations to nonnegative real numbers.

2.3.2 Inference and Learning for Random Fields

Having defined the factorization of the probability distribution and specified the energy function now we explain the two important tasks: inference and learning the model parameters. We will define the different types of learning and inference problems that will be used in later chapters. The two important inference problems frequently encountered in computer vision applications are Maximum a Posteriori (MAP) inference and *probabilistic inference*. In MAP inference, $X^* \in L$ is estimated which corresponds to maximum probability. In *probabilistic inference*, we find the log partition function and the marginal distribution of for each state of random variables.

For general (loopy) graphs this problem is known to be NP-hard, but for tree structured graphs the problem can be solved efficiently. For loopy graphs, when solving binary labeling problem, if special cost functions are used then graph cut [42, 4] algorithms can solve the inference exactly and efficiently. For multi labeling problems, graph cuts can provide approximate solutions, as well [43]. We start to explain the belief propagation method which is exact for acyclic graphs and provides an approximation in the general case.

Sum-Product Belief Propagation

The sum-product algorithm is a dynamic programing algorithm that can compute the partition function Z and the marginal distributions for all variables. It was initially designed to work on tree-structured factor graphs to provide exact solutions but later on modified version of it, called Loopy Belief Propagation was proposed to find approximate solutions on general (loopy) factor graphs. It is called a message passing algorithm and it works by passing real valued functions called messages along the edges between the nodes [44]. Let's say we want to compute marginal probability $p(X_i)$. The marginalization is done by summing the joint distribution $p(X_i) = \sum_{\mathbf{X}/X_i} p(X_j)$ over all variables except X_i. The joint distribution of variables $\mathbf{X}/X_i$ can be defined in terms of factor graph expression. This actually allows us to do a mathematical trick by replacing summation and product operations. Graph factorization provides two type of messages so that these equation can be written more explicitly. They are shown in Fig. 2.9. The first type is called variable-to-factor (2.9(b)) and the other type is factor-to-variable message (Fig. 2.9(a)).

The marginal probability equation is given as:

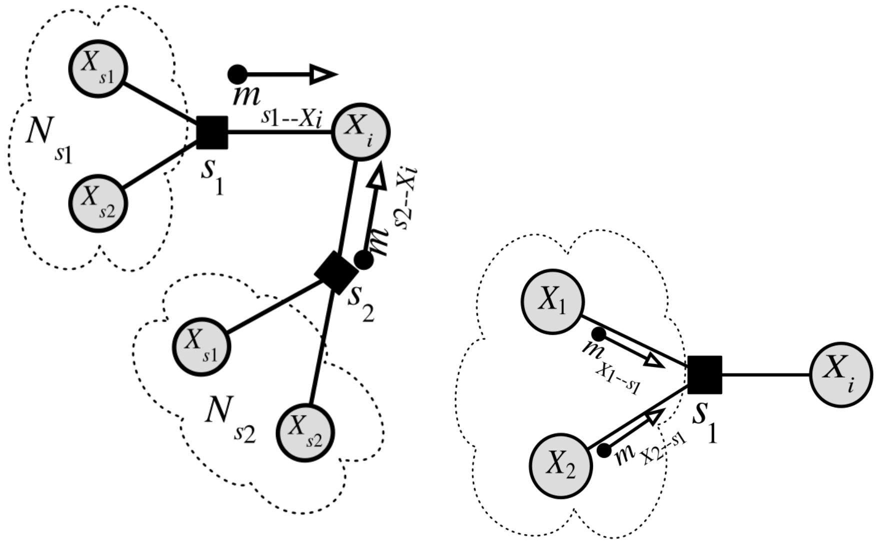
$$p(X_i) = \prod_{s \in N_i} \left[\sum_{X \in N_s / X_i} N_s(X_i, X_s) \right] \tag{2.15}$$

where N_s denotes the set of factor nodes that are neighbors of X_i and N_s represents set of all variables in the subtree connected to the variable X_i via the factor node s. $N_s(X_i, X_s)$ represents the product of all the factors in the subgroup. To simplify the Eq. 2.15, we can write

$$m_{s \to X_i} \equiv \sum_{X \in N_s / X_i} N_s(X_i, X) \tag{2.16}$$

This gives us the factor-to-variable message definition. So the marginal probability $p(X_i)$ can be given by the product of all incoming messages arriving at node X_i.

If we take a deeper look into the definition of $N_s(X_i, X)$ we realize that, it can be described



(a) (b)

Figure 2.9: Message computation on tree structured graphical models for message passing algorithm. The illustrations are adopted from [6]

by a factor sub-graph and so can itself be factorized into

$$N_s(X_i, X_s) \equiv \sum_{X \in N_s/X_i} f_s \cdot \prod_{j \in N_s/X_i} m_{X_{sj} \rightarrow s} \tag{2.17}$$

Here f_s is the factor associated with factor node s. Substituting Eq. 2.17 into 2.16, and using log-sum-exponential expression, we obtain

$$m_{s \rightarrow X_i} = \log \sum_{X \in N_s/X_i} exp(-f_s + \sum_{j \in N_s/X_i} m_{X_{sj} \rightarrow s}) \tag{2.18}$$

See Fig. 2.9(a) for a visual illustration.

Finally, it is possible to derive an expression for computing the messages from variables nodes to factor nodes using the same principal (i.e. sub-graph factorization). According to this factorization, the product is taken over all neighbors of node X_j except for node s_j. This is represented as

$$m_{X_j \rightarrow s_j} = \prod_{j \in N_s/s_j} m_{s_j \rightarrow X_j} \tag{2.19}$$

See Fig. 2.9(b) for a visual illustration.

Here message ordering is important since each type of message depends on the previously computed messages. There are two special cases when messages do not depend on previous: 1) The variable-to-factor message in which no other factor is adjacent to the variable. 2) The factor-to-variable messages in which no other variable is adjacent to the factor. For tree-structured factor graphs there always exist at least one such message that can be computed initially. The computed message in turn enables the computation of other messages.

We can summarize the inference algorithm in three steps: First, the graph is orientated by choosing one node as the root and the leaf nodes. In the first step, messages are passed inwards: starting at the leaves, each node passes a message along the edge towards the root node. The tree structure guarantees that it is possible to obtain messages from all other adjoining nodes before passing the message on. This continues until the root has obtained messages from all of its adjoining nodes. When reached to the root, the partition function can be computed [44, 4]. The second step involves passing the messages back starting from the root towards the leafs. The algorithm is completed when all leaves have received their messages.

Similar to marginal computation, during forward passing and back passing instead of using summation (marginalization of all variables) if max operation is used (best state is chosen) at the end of back tracking one can compute the most probable state for every node. This is called MAP inference and since max operation is used instead of summation, the algorithm takes name *max-product*. This algorithms is generalization of *viterbi* algorithm [6].

Graph Cuts

Graph cuts were used in computer vision for the first time by Greig et al. [45]. It has been shown that if an Ising model [46] was used for defining the pairwise potentials of a two label MRF, then an exact MAP solution can be obtained in polynomial time by solving a st-mincut problem. A restricted class of energy functions on binary variables can be minimized globally by the binary graph cut method. To do this, the regular bipartite trap of MRF is changed into an undirected auxiliary graph that contains two extra special nodes, the source s and the sink t. Also for each edge a non-negative weight is assigned. The minimum s?t cut of the graph separates the nodes s and t such that it has the smallest overall weight. This optimal solution to the original energy minimization problem returns the solution for labeling problem.

The class of binary energy functions can be solved by graph cuts is of the form:

$$E(\mathbf{Y},\mathbf{X}) = \sum_{i\in V} \mu_i(Y_i, X_i) + \sum_{i,j\in N} \varphi_{i,j}(Y_i, Y_j) \tag{2.20}$$

Here, $\mu_i(Y_i, X_i)$ and $\varphi_{i,j}(Y_i, Y_j)$ denote unary and pairwise factors, respectively. The unary terms are restricted to be non negative while the pairwise terms should satisfy sub modularity [47]. According to the sub modularity definition, the pairwise factors encourage their adjacent variables to take the same state.

For multi labeling problem it has been shown that move making local search algorithms such as α-β- swap and α- expansion approximately global solutions can be obtained. The α- expansion algorithm is the most popular graph cut algorithm for discrete multi label MAP inference. In an iterative scheme, for a given label α, the expansion neighborhood allows every node to either remain in tis current state or to change its state to α. Every single step is actually a binary labeling problem that finds the optimal solution within the neighborhood of the current solution.

Parameter Learning for Conditional Random Fields

Instead of specifying a single-fixed model we can introduce free parameters into the model. Given some training data we can adjust these free parameters to effectively learn a good mapping between observation and output variables. This is known as parameter learning and training the model.

Given an observation y and a parameter vector w, the conditional probability of a labeling x is

$$P(\mathbf{X}|\mathbf{Y}, w) = \exp(-E(\mathbf{X}|\mathbf{Y}, w) - \log Z(\mathbf{Y}, w)), \tag{2.21}$$

where $Z(\mathbf{Y}, w) = \sum_{x\in L} \exp(-E(X|Y, w))$ is the *partition function*. The energy is often defined as the negative inner product $E(x|y, w) = -\langle w, \psi(x, y)\rangle$, in which $\psi(x, y)$ is the concatenation of features of all nodes and edges.

We train the parameter vector w from a given set of training data $\{(x^n, y^n)\}$, $n = 1, \ldots, N$.

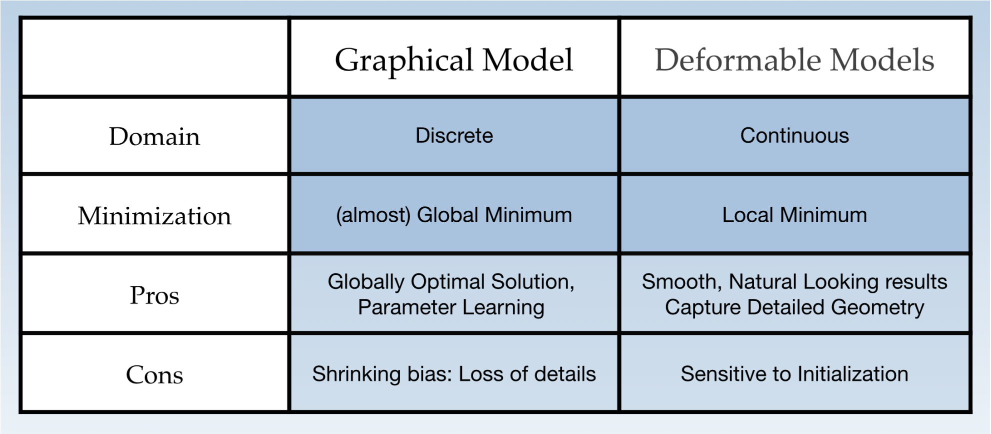
This can be achieved by minimizing the convex loss function defined as

$$loss(w) = \|w\|^2 + c\sum_{n=1}^{N} [\log Z(y^n, w) - \langle w, \phi(x^n, y^n)\rangle]. \tag{2.22}$$

This optimization problem is called as maximum conditional likelihood estimation and does not have a close formed solution. But it is convex and we can find the optimal w^* using gradient descent which is the most straight-forward technique to numerically solve problem. Starting with an arbitrary estimate of the w, one iteratively approaches the minimum. For more details of the parameter learning we refer the reader to [4].

2.4 Comparison of Graphical Models & Deformable Models

We summarize the differences and similarities of Deformable Models and Graphical Models in Fig. 2.4. The two well studied, popular methods are both energy minimization based, but they are completely different approaches. One main difference is the domain in which they are defined. Deformable Models are defined in continuos domain while Graphical Models are discrete. Correspondingly, DM is continuous domain optimization problem whereas GM is discrete optimization problem. DM optimization provides locally optimal solutions while GM optimization returns globally optimal solutions. Both have certain advantages and disadvantages. With DM it is possible to get natural looking fine detailed segmentation surfaces. With GM methods it is possible to get quick and initialization independent reasonably good results. In the next chapters, we are going to to show that it is possible to combine them taking advantage of their pros. Also we will show that for particular applications GMs are preferred when probabilistic learning tools are desired to arbitrarily partition the image into multiple parts rather than focusing a particular object in the scene.



	Graphical Model	Deformable Models
Domain	Discrete	Continuous
Minimization	(almost) Global Minimum	Local Minimum
Pros	Globally Optimal Solution, Parameter Learning	Smooth, Natural Looking results Capture Detailed Geometry
Cons	Shrinking bias: Loss of details	Sensitive to Initialization

(a)

Figure 2.10: Comparison of Graphical Models and Deformable Models.

Chapter 3

Collaborative Multi Organ Segmentation by Integrating DM & GM

Organ segmentation is a challenging problem on which significant progress has been made. Deformable models (DM) and graphical models (GM) are two important categories of optimization based image segmentation methods. Efforts have been made on integrating two types of models into one framework. However, previous methods are not designed for segmenting multiple organs simultaneously and accurately. In this chapter, we propose a hybrid *multi organ* segmentation approach by integrating DM and GM in a coupled optimization framework. Specifically, we show that region-based deformable models can be integrated with Markov Random Fields (MRF), such that multiple models' evolutions are driven by a maximum a posteriori (MAP) inference. It brings global and local deformation constraints into a unified framework for simultaneous segmentation of multiple objects in an image. We validate this proposed method on two challenging problems of multi organ segmentation, and the results are promising.

3.1 Introduction

Segmenting anatomical regions from medical images has been studied extensively and it is a critical process in many medical applications. For MR or CT images, distinct boundaries between organs may be blurred and ambiguous. Furthermore, the intensity profile of the region inside the object can represent inhomogeneity. Thus accurate segmentation of these organs from these images is very challenging. Two important categories of optimization based image segmentation methods are Graphical Model (GM) based [48, 42, 49] and Deformable Model (DM) based [13, 50, 51, 52, 53, 24] methods. Both categories are able to achieve satisfactory results by combining the confidence based on local evidence such as colors and textures and the confidence based on global evidence such as the smoothness of the boundary. However, each model has its own strengths and weaknesses.

GM methods can reach the global optimum because of efficient algorithms inspired by the max-flow min-cut theorem. Although in most cases, the used algorithm such as α-expansion [43] is still a local search algorithm, one can claim the solution is approximately optimal due to the extremely large local search neighborhood. The unique solution of GM methods could achieve reasonably good segmentation quality, but sometimes misses the fine details, which could be critical in medical context. On the other hand, DM methods, given a reasonably good initialization, would have the flexibility to deform to nearby local minimum of the energy which could achieve segmentation with nice details. However, due to the complicated energy model, DM methods use gradient descent method, and thus could be trapped in local minima that are very far from the ground truth.

3.2 Motivation

In this chapter, we propose a hybrid model to naturally cope with the *multi organ* segmentation problem by integrating GM and DM methods. Our main idea is to formulate a multi label graphical model problem at each iteration, and use the MAP inference result as part of the gradient (i.e., external forces) of deformable model. Such inference task could be solved efficiently using α-expansion algorithm [43]. The main advantage of our hybrid model is that in the multiple labeling graphical model, regions of different labels bring high level constraint naturally and implicitly to the external forces of deformable models. Therefore, two similar neighboring regions can still be separated easily, and the resulting regions are more accurate as per the ground truth. See Figure 3.1 for an illustration and Section 3.5 for more comparisons. Besides the accuracy and robustness for multi organ segmentation, this proposed model does not need any offline learning (in contrast to [54]), has few parameter, and contains the advantages of both GM and DM methods. We have applied this proposed method to two challenging clinical applications, i.e., knee joint bone segmentation and cardiac segmentation, and achieved promising results.

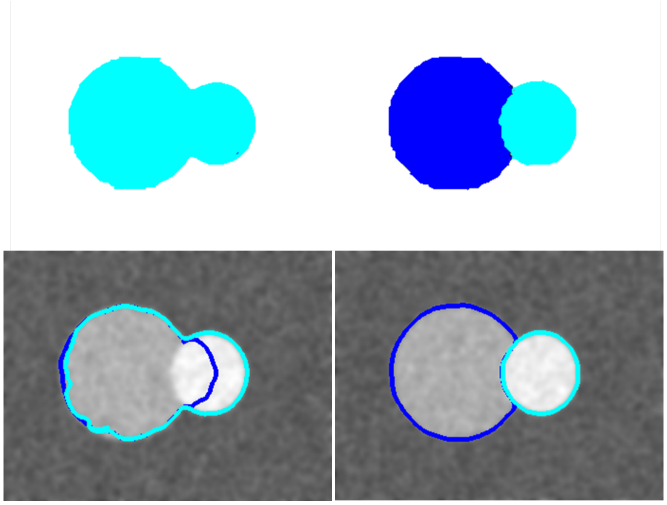


Figure 3.1: Top: binary graphical model (left) would produce wrong regions for either labels, but multi-label graphical model (right) is correct; Bottom: comparison of combining DM with binary GM (left) and multi-label GM (right).

3.3 Related Work

Several works combining the two models have been introduced in the computer vision and the medical imaging literature. This approach could be defined as integration of deformable models with learning-based classification. The general idea is to use graphical model at the gradient computation step of the deformable model, so that the deforming contour is less likely to be trapped at local minima that is far from the ground truth. Chen et al. [54] deform the contour by iteratively solving a graphical model optimization problem. At each iteration, the graphical model enforces that the deformation should respect priors trained offline and also should not be too far from the current contour. Huang et al. [2] formulated the 2D segmentation task as a joint inference problem of contour detection and pixel labeling so that the two models are tightly coupled. The active contour and the MRF are considered as two different modules, which are integrated into a graphical model, and the solution is given by a Bayesian decomposition that decouples computing the deformation and label field inference. Instead of the MAP inference, the marginals of the graphical model are used as part of the gradient in deformable model. Intuitively, their formulation can be seen as a deformable model that is driven by the probability field estimated with the MRF. It was proposed to overcome the limitations of edge-based probabilistic active contours.

However, these methods only use binary labeling when constructing the graphical model, which cannot be used directly for segmenting multi organs. One could extend these methods to segment multi organs straightforwardly by constructing a binary graphical model for each deforming contour. In this setting, the foreground and background labels correspond to the current concerned organ and the union of the rest, respectively. Unfortunately, this method would have difficulty to distinguish the two neighboring regions if they only have slightly different intensity distributions but are very different from the remaining regions (see an example in Figure 3.1).

3.4 Integrating Deformable Models and Graph Cuts

The goal of our segmentation method is to find multiple regions with smooth and closed boundaries. We start with the overall energy functional for m models forming a set **C** as:

$$E(\mathbf{C}) = E_{int}(\mathbf{C}) + \sum_{i=1}^{m} E^i_{ext}(\mathcal{C}_i) \tag{3.1}$$

Here, the first term E_{int} is the smoothness term (see [13] and [50]) and E^i_{ext} is the data term for contour i. In contrast to the classical choice such as the difference from a constant value [52] or the negative log likelihood of a given distribution [51], we assume a region of interest (ROI) for contour i is given (R_i), and define the i-th external energy as

$$E^i_{ext}(\mathcal{C}_i) = \frac{1}{Vol} \iiint_{\Omega} (\Phi_{\mathcal{C}_i}(x) - \Phi_{\partial R_i}(x))^2 dx \tag{3.2}$$

Here Ω is the image domain and Vol is its volume. We denote $\Phi_{\mathcal{C}_i}$ and $\Phi_{\partial R_i}$ as the signed distance functions of the contour and the boundary of the region of interest, respectively. Intuitively, minimizing this term would pull the contour, C_i, towards the boundary of ROI, ∂R_i. Our algorithm minimizes $E(\mathbf{C})$ using gradient descent method. At each iteration, we compute the gradient $\partial E / \partial \mathbf{C}$ and evolve the contours accordingly.

At each iteration, we define the ROIs by constructing a mutli-label graphical model depending on the current contours **C**. The MAP of the graphical model gives us the set of ROIs. We denote L as a labeling, which assigns to each pixel/voxel a label belonging to the label set

$\mathcal{L} = \{1, ..., m, m+1\}$, corresponding to regions inside the m contours and the background $(m+1)$. We compute the labeling optimizing the conditional probability

$$L^* = \underset{L}{\operatorname{argmax}} \{P(L \mid I, \mathbf{C})\} \tag{3.3}$$

The ROI R_i is then the set of voxels with label l^i in L^*. Assuming that the image data I and the deformable models $\mathbf{C}$ are conditionally independent given the labeling L, we define the posterior probability of labeling L:

$$\begin{aligned} P(L \mid I, \mathbf{C}) &= \frac{P(I, \mathbf{C} \mid L)P(L)}{P(I, \mathbf{C})} = \frac{P(I \mid L) \cdot P(\mathbf{C} \mid L) \cdot P(L)}{P(I, \mathbf{C})} \\ &= \frac{P(I) \cdot P(L \mid I)}{P(L)} \cdot \frac{P(L \mid \mathbf{C})P(\mathbf{C})}{P(L)} \cdot \frac{P(L)}{P(I, \mathbf{C})} \\ &= \frac{P(I) \cdot P(\mathbf{C})}{P(L) \cdot P(I, \mathbf{C})} \cdot P(L \mid I) \cdot P(L \mid \mathbf{C}) \end{aligned} \tag{3.4}$$

With the assumption that each labeling has equal prior probability, and I and $\mathbf{C}$ are fixed we have

$$P(L \mid I, \mathbf{C}) \propto P(L \mid I) \cdot P(L \mid \mathbf{C}) \tag{3.5}$$

Here the negative log likelihood of $P(L \mid I)$ is the same as the energy in conventional multi-label graphical model, and $P(L \mid \mathbf{C})$ is the model shape prior, defined as $P(L \mid \mathbf{C}) = \prod_{j \in V} P(L_j | \mathbf{C})$, where V is the set of all voxels. The shape prior of each individual voxel $P(L_j | \mathbf{C})$ is inversely proportional to the distance from the model. Let $i = L_j$, we have

$$P(L_j \mid \mathbf{C}) = P(L_j \mid \mathcal{C}_i) = \begin{cases} 1 & \text{if } \Phi_{C_i}(j) \geq 0, \\ 1 - \frac{\|\Phi_{C_i}(x_i)\|}{\|(\Phi_{C_i})\|_\infty} & \text{otherwise} \end{cases}$$

According to the above definition, voxels which are closer to a model are more likely to belong to the label of that particular model. This leads to the final energy for graphical model as

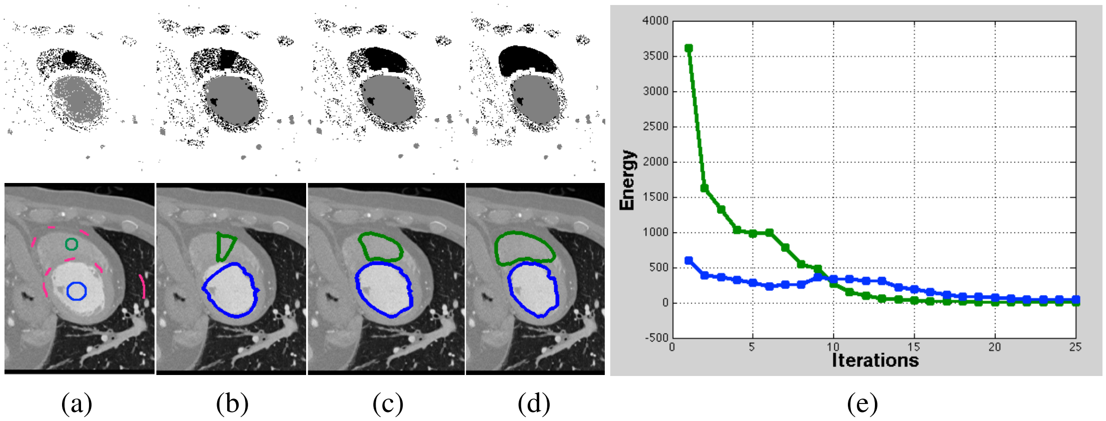


Figure 3.2: Iterative, model constrained estimation of labels. (a) User provided model initializations and background cues (strokes in magenta). (b-d) multi-phase graph cut results and states of the deformable models at consecutive iterations. In top row, gray color represents the label for blue model (left ventricle) and black color represents the label for green model (right ventricle), white label represents the background. In bottom row, deformable models at consecutive iterations are shown. (e) Plot of E_{ext} energy values computed at each iterations.

$$-\,log(P(L \mid I, \mathbf{C})) \propto E(L) = \sum_{j}^{N} \big(u_j(L_j) - log(P(L_j \mid \mathbf{C}))\big) + \sum_{\substack{p,q \in \mathcal{N} \\ L_p \neq L_q}} b_{pq} \tag{3.6}$$

In the above equation, $u_j(L_j)$ is the cost of assigning label L_j to j^{th} voxel and are computed as $\|I_j - \mu_i\|^2/\sigma_i$ where $i = L_j$, μ_i and σ_i are mean and standard deviation of the intensities inside the region enclosed by $\mathcal{C}_i$. b_{pq} is the typical binary term of MRF and is defined as

$$(|I_p - I_q|^2/\sigma) \times dist(p, q)^{-1} \tag{3.7}$$

For more details about construction of the graph and inference method we refer the reader to [49] and [43]. We solve the MAP inference problem given in Eq. 3.3 using α-expansion algorithm [43].

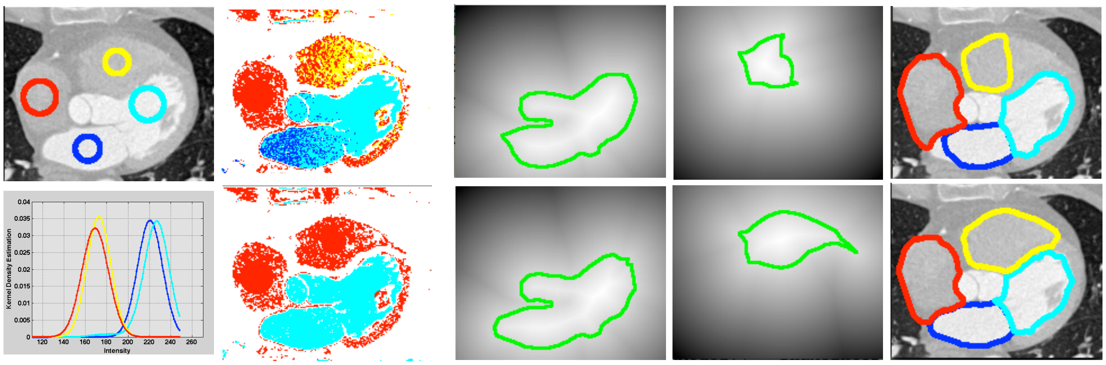
3.4.1 Alternating Energy Minimization Scheme

According to the energy function defined in Eq. 3.1, the models are deformed under smoothness constraints and the attraction force coming from the ROIs. The minimization problem can be achieved by using an alternating minimization scheme where we do coordinate descent and split it into two problems: 1) fix multi phase labeling conditioned on the given models and image data; 2) locally deform the models minimizing the other energy terms such as smoothness and image gradient etc. A single deformable model $\mathcal{C}^i$ is represented explicitly in terms of splines (2D) or meshes (3D) as in [1], [55] and deformed according to the deformable model dynamics explained there. Our segmentation process starts with initialization of the models $\{\mathcal{C}_1, ..., \mathcal{C}_m\}$ for the foreground objects and providing markings (seeds) for the background. Before starting the iterative process, a graph is constructed only once with the desired connectivity (can be 8 in 2D or 26 in 3D) using the initial models and the seeds. The unary and compatibility potentials along with the model shape constraints are computed for the graph cut. Then we calculate the labeling according to the minimization of Eq. 3.6. Once we obtain the labels for each pixel, we select the ROIs that intersect the models, and for each model, we compute driving forces using the associated ROI (see Eq. 3.2). We continue this alternating process iteratively and at each iteration after the models are deformed we update the parameters μ, σ and $P(L_j \mid \mathbf{C})$. We continue deformation in the same scheme until convergence.

In Fig. 3.2, we demonstrate this process for two foreground object segmentation. Fig. 3.2(a) shows the user initialized models for the foregrounds (blue and green) and strokes for the background (in magenta). Fig. 3.2(b-d) top row show the states of MRF labeling at consecutive steps. One can observe that, as the iterations continue not only the smoothness of the labels are enhanced but also the accuracy of the labeling gets better and better. Thus, E_{ext} provides more accurate driving force as the iterations continue. Fig. 3.2(e) shows the energy minimization process at each iteration which is calculated from Eq. 3.2. According to the plot, it is clear that the energy computed for each model is monotonically minimized and converges.

3.4.2 Implementation Detail

Multi phase graph cut method is not guaranteed to always return smooth and distinct region segments for each label, especially when target regions present similar/identical intensity properties. Due to this similarity, the calculated unaries could be in-distinctive and resulting labeling would not return structured segments. In such a case, the selected ROI per deformable model



(a) (b) (c) (d) (e)

Figure 3.3: Label unification and its effect to the segmentation: (a) Example kernel density estimate of the regions underneath the models. (b)Top row MAP-MRF labeling without unification (4 foreground labels). Bottom row, MAP-MRF labeling with unification (2 foreground labels). (c)-(d) Signed distance maps for two different models with (bottom) and without (top) unification.(e) Resulting segmentations obtained w/o unification.

would not be accurate enough to drive the model correctly. To tackle this problem we develop an online label unification method which adaptively identifies the labels of the models to be merged. We do this unification operation according to the Kullback-Leibler divergence (KL) between the kernel density estimated intensity distributions underneath the models. In Fig. 3.3(a), the estimated distributions for each model region is shown. In this scenario, according to KL score, the algorithm decides to unify the yellow model label with the red model label, and the light blue label with the dark blue label. With the unification of the labels target label size becomes 3 including the background. In (b), we compare the resulting labels of multi phase graph cut w/o using the unification. As seen in the bottom image, the labeled regions are more smooth relative to the top image. Note that the label set is automatically shrunk to 2 foreground labels but this does not mean that number of models are changed. The models always select the binary ROI that intersect with itself. Then, the distance map of the ROI is used in the E_{ext} term as shown in Eq. 3.2. In Fig. 3.3(c) and (d), we show the effect of our unification mechanism to the deformation force. As seen in the bottom image the distance map computed for right ventricle and the left atrium are more accurate. Also in Fig. 3.3(e) we compare resulting segmentations at convergence to demonstrate the effect of our unification process.

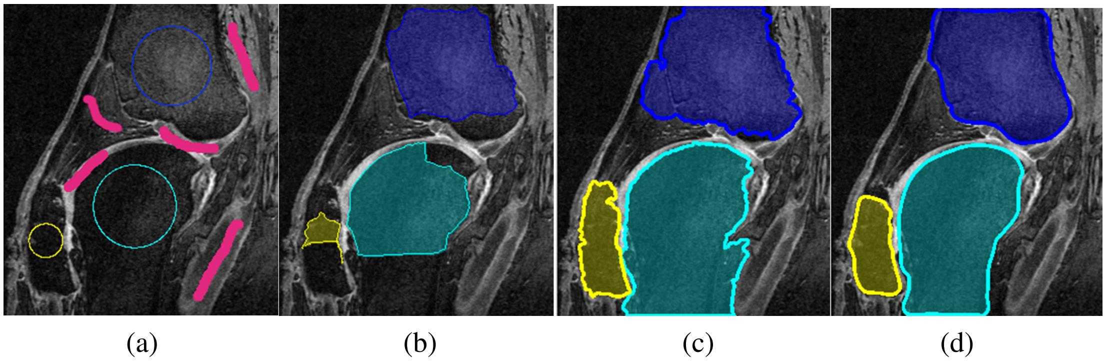


Figure 3.4: Experiment on knee joint MR image. (a) Three model initializations and background cues (strokes) for segmentation of tibia, femur and patella. (b) Result of [1]. (c) Result of [2]. (d) Result of our method.

Algorithm 1: Alternating minimization scheme.

```
Initialize the set of contours $\{\mathcal{C}_1, ..., \mathcal{C}_m\}$ and the background cues as strokes;
**while** *change in the states of* $\{\mathcal{C}_1, ..., \mathcal{C}_m\}$ *)* $\leq$ *thres* **do**
    · Calculate $\tilde{\mathbf{C}}$ using KL divergence KL $(\mathcal{C}_1, ..., \mathcal{C}_m)$;
    · Calculate $\alpha$-expansion parameters for $n+1$ labels. // $n$ is obtained after label
    unification;
    · Calculate region labeling **L** based on Eq.3.6, using$P(\mathbf{L} \mid I, \tilde{\mathbf{C}})$;
    · Calculate $\Phi_{\partial R_i}$ for each contour;
    · Deform contours $\{\mathcal{C}_1, ..., \mathcal{C}_m\}$ locally using traditional speed terms in addition to
    the GM based term;
**end**
```

With this alternating minimization scheme, we take advantage of the strengths of these two methods. At each iteration the models are updated locally with globally computed forces and the global parameters of the MAP-MRF are updated locally. Moreover as the models start getting close to the actual target object shape, the system reaches convergence very precisely.

3.5 Experiment & Results

We validate this proposed method on two multi organ segmentation applications. Our method is compared with two relevant approaches, 1) Metamorphs [1] which integrates texture information into deformable models, and 2) a graphical model coupling MRFs and deformable models [2]. They are evaluated on both MR and CT data sets whose ground truths were manually annotated by clinical experts. Our algorithm was implemented in MATLAB with C programming extensions. We tested the algorithm on a quad core (3.4 GHz) computer with 8Gb of memory.

3.5.1 Knee Joint Bone Segmentation

We segment knee joint bones femur, tibia and patella from 23 MR scans. The data scan protocol consists of 3D DESS scan with water excitation having 0.36x0.36x0.70mm voxel size. The qualitative comparisons are shown in Fig. 3.4. Due to the intensity heterogeneity inside the bone structures (particularly in regions close to the cartilage), [1] does not perform well and stuck in local minima as it uses the local online intensity modeling. Starting from the same initializations, our method converges to the final state for all 3 organs within 30 iterations ($\sim$6.5 sec/iter) while [1] stops at the final stage after 18, 23 and 38 iterations ($\sim$8 sec/iter) for patella, tibia and femur, respectively. For fair comparison, we tune deformation parameters (i.e. smoothness, image gradient, balloon) for method [1] to achieve the best performance and keep them exactly same for our method.

To compare with [2], we also carefully selected its parameters in order to obtain the best results. We observed that the Expectation Maximization approach in [2] performs better than [1] when updating the attraction force for the deformable models. However, it takes more iterations due to its narrow band limitation also running time per iteration takes $\sim$20 sec which is 3 times the running time of our approach. In addition, our method takes advantage of multi phase MRF labeling and the ROIs per models are estimated more accurately. Thus, our hybrid approach gets out of local minima and also avoid possible leakages towards muscle regions.

Segmentation of bone structures from Knee MR images is a critical process for several diagnoses like surgical knee replacement and cartilage segmentation. In both cases the analysis of the structural anatomy of the surface of femur, tibia and patella (knee cap) are very important. In the surgical operations, the damaged knee joint is replaced with artificial prostheses and these prostheses should ensure optimal outcome during and after surgery. Also bone-cartilage interface should be detected very accurately to measure the volume of cartilage tissue.

Organs	**Methods**	**Volume Overlap Error** [%]	**Avg. Surface Distance [voxel]**	**Iter Number till convergence**
femur	[1]	33.32±16.75	12.2±9.42	38
femur	[2]	8.6±2.81	6.34±1.99	66
femur	Ours	**7.34**±**2.75**	**4.2**±**2.42**	~30 for all
tibia	[1]	17.12±5.11	7.32±3.32	23
tibia	[2]	7.37±3.17	5.66±4.32	90
tibia	Ours	**6.27**±**2.22**	**3.3**±**0.57**	~30 for all
patella	[1]	28.12±12.31	7.1±6.19	18
patella	[2]	4.23±0.31	1.5±0.19	43
patella	Ours	**3.9**±**1.37**	**1.4**±**0.32**	~30 for all
LA	[1]	12.12±2.75	2.2±1.32	27
LA	[2]	11.89±1.85	2.09±1.2	31
LA	Ours	**7.12**±**2.21**	**1.9**±**1.45**	~50 for all
LV	[1]	11.32±3.75	4.4±3.42	45
LV	[2]	13.45±4.35	4.9±3.57	55
LV	Ours	**8.12**±**1.35**	**3.1**±**1.52**	~50 for all
RA	[1]	7.32±2.00	2.2±0.54	23
RA	[2]	6.13±2.03	2.00±0.54	30
RA	Ours	**5.88**±**2.33**	**1.92**±**0.32**	~50 for all
RV	[1]	9.32±2.75	3.2±1.42	45
RV	[2]	12.98±2.86	4.32±1.22	55
RV	Ours	**9.12**±**2.9**	**3.1**±**0.42**	~50 for all

Table 3.1: Quantitative comparisons of our method with two relevant approaches, 1) Metamorphs [1] which integrates texture information into deformable models, and 2) a graphical model coupling MRFs and deformable models [2]. We reported the mean and standard deviation of voxel distances between segmented surfaces and ground truth, and volume overlap errors in proportions for 23 MRI and 15 CT scans.

3.5.2 Cardiac Segmentation

We evaluated our algorithm on segmenting the cardiac structures such as Right Atrium (RA), Right Ventricle (RV), Left Ventricle (LV), Left Atrium (LA) from a set of 15 CT volumes.

The data scan protocol consisted of 3D CT scan with 1.0x1.0x1.0mm voxel size. The figure is shown in the supplementary materials due to page limitation. Compared to [1], our method performs slightly better in terms of avoiding leakages towards the heart muscle. In addition, the myocardium between LV and RV is identified better. For the LV case, the papillary muscles are nicely included into the segmentation owing to the smoothness factor of the graphical model and the parameter update scheme of the deformation. With our method, all models converge to the final state within 50 iterations (∼4.9 sec/iter). Compared to [2], our method achieves better average accuracy of all organs since the background is identified well within the multi-region labeling scheme. Most significant accuracy differences between [2] and our method are observed for the RV and LV cases, due to local minima problems.

As seen in Fig 3.5(c), there is overlapping region between Left Atrium and Left Ventricle. Also the papillary muscles inside Left Ventricle can not be enclosed inside the blood pool due to the different intensity profile. Accordingly, the walls of the Left Ventricle cannot be captured well. Similarly, due to the very low contrast, myocardium region between the ventricles can not be distinguished. It converges to the resulting segmentation in 36 iterations. In summary, [1] fails handling leakages and getting out of local minima. Fig. 3.5(d) is the result of [43]. As seen in the figure, although we tune the smoothing factor of the graphical model approach, it results in too many separate components yielding five different regions. Also separating Left Atrium from Left Ventricle is not possible with this method. One could use more strong smoothing factor to avoid spurious white pixels in the target object regions, but this effects other pixels labels badly since it is global labeling approach.

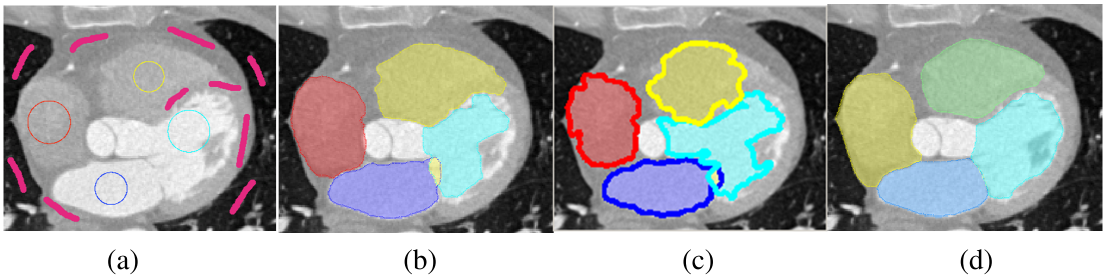


Figure 3.5: Quantitative comparisons on the heart CT image. (a) Four model initializations and background cues (strokes) for segmentation of Right Atrium, Right Ventricle, Left Ventricle and Left Atrium. (b) Results of [1]. (c) Results of [2]. (d) Results of our method.

Table 3.1 shows quantitative results of our method in both applications. We reported the

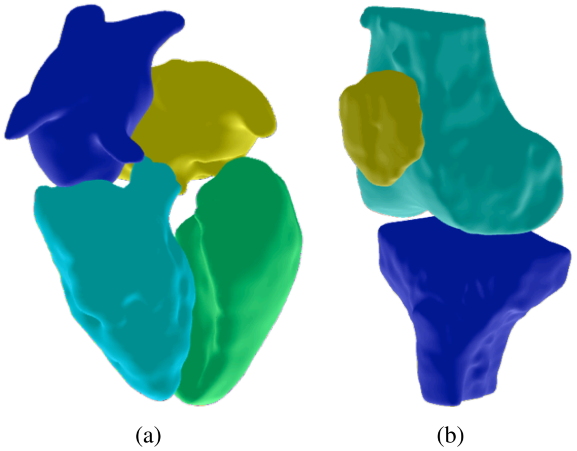


Figure 3.6: 3D visualization of results using our method; (a) Cardiac four chamber data. (b) Knee joint bones.

mean and standard deviation of voxel distances between segmented surfaces and ground truth, and volume overlap errors in proportions. We also visualize 3D results of our method in Fig. 3.6. In general, our method achieves more accurate results than the other two hybrid approaches, and is also more efficient.

3.6 Conclusions

In this chapter, we proposed a new hybrid multi object segmentation approach, in which deformable models and multi label graphical models are integrated into an alternating optimization framework. We integrate multi phase graph cut labeling into deformable model framework so that it provides the desired speed term for each deformable model to converge to the true boundary. We provide solutions for potential drawbacks of the two methods by combining the benefits of them to segment multiple objects efficiently and simultaneously using global and local constraints. We validated our method on medical images (MR, CT) and real-world images. As a future direction, we are currently working on speeding up the running time of our algorithm in 3D. We are planing to use a supervoxel approach which could reduce the graph cut processing time drastically. We also consider using conditional random fields model for a learning based multi object segmentation that might possibly use coupled prior information as well.

Chapter 4

Review on Neuron Reconstruction from Electron Microscopy Images

4.1 Introduction

Researchers have been working on mapping of circuit diagrams of nerve systems since 1950s. It has been argued that detailed maps of synaptic connectivity would be very helpful for understanding how the brain causes behavior and how brain malfunctions cause behavioral disorders [56]. With the recent developments in electron microscopy (EM) technology, people in the field of neuronal circuit reconstruction, now have the chance to work on very large scale and high resolution neuronal tissue volumes towards understanding the functionality of neuronal structures. The advancements in EM technology made the identification of dendrites, synapses and axons possible. In the last decade, a young topic named *Connectomics* [57], has shown significant progress. It provides new insights into the relation between the brain structures and its function [58]. Modeling this relationship is believed to provide deeper understanding for the principal reasons of serious brain diseases such as mental illnesses and learning disorders.

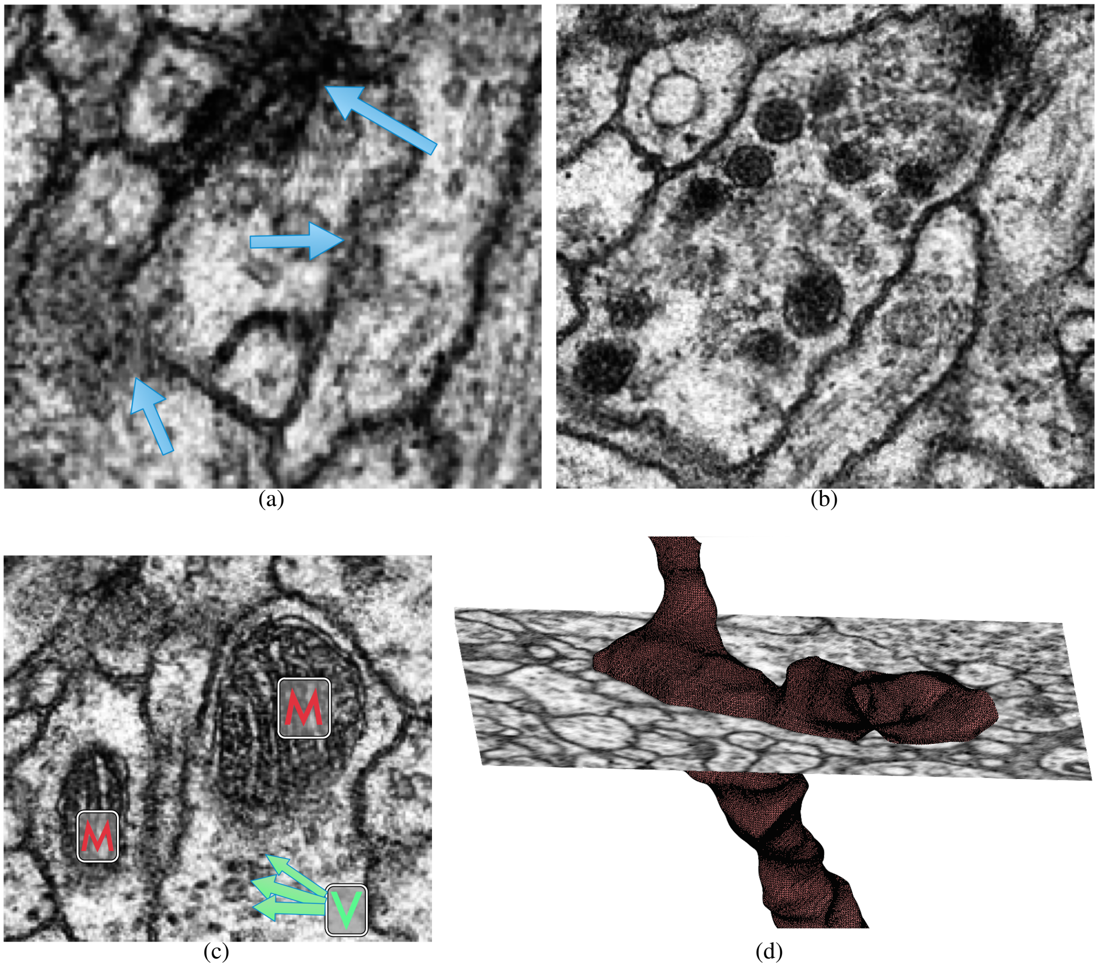
To reconstruct neuronal arbors, synapses, and glial cells (this means a complete wiring), the brain images has to be acquired at nanometer (nm) resolution []. There are two notable EM imaging techniques that can provide sufficient information for reconstruction of a nervous system: i) Serial section transmission electron microscopy (ssTEM) is the first step towards obtaining a complete neuron wiring diagram and provides potential of bringing a huge impact on the understanding of the whole nerve system by providing sufficiently high (synaptic) resolution in tractable amount of time. It can be used for reconstruction of a complete nervous system of small organisms such as *C. elegans* or *Drosophila* larva and delivers anisotropic volume with large slice thickness [8]. ii) Focused ion beam based serial section (FIBSEM) imaging is used to acquire sub-volume of adult mouse brain tissue at higher resolutions with isotropic resolution in three dimensions [9].

4.2 Challenges in Neuron Segmentation and Reconstruction

Neuron reconstruction from EM images relies on the accurate detection and segmentation of branchings and mergers of neuronal structures, thus small merge or split errors on membrane segmentation would make the results useless and the interpretation wrong [59, 60]. The decision for a membrane needs multiple levels of information which come from indicators such as gradient, texture, intensity and even prior knowledge. This challenges the computer vision and machine learning community to develop accurate and efficient techniques for neuron cell segmentation. Currently, there are remarkable progress in detection, segmentation and reconstruction of neuronal arbors; however, the state-of-the-art methods are not completely sufficient since the accuracy requirement is extremely high. Segmentation of EM images faces significant challenges:

- Diffused and low contrasted membranes (see Fig. 4.1(a)).
- Complex appearance within cells (see Fig. 4.1(b)).
- Similar membrane appearance for multiple structures (i.e., mitochondria and vesicles)(see Fig. 4.1(c)).
- High variation in shape of structures (elongated, twisted) (see Fig. 4.1(d)).
- Very large data size requires efficient methods in terms of space and computational complexity.

With the developments in computer processors and data storage systems, today, it is possible to acquire large image datasets in the GB-TB range. However, with data sets this size, manual analysis is no longer feasible. The current automated methods cannot solve all the above mentioned challenges [61, 62]. Therefore, manual or semi-automated interactive systems are still necessary in practice since *Connectomics* requires extremely accurate partitioning. In order to achieve a satisfying quality, human experts have to *proofread*, namely, manually correct the results of automated methods. Thus, semi-automated tools which can address large data sets with reduced user interaction and low complexity are also of huge interest [63, 64, 65, 66, 67].



(a)

(b)

(c)

(d)

Figure 4.1: Sample image patches illustrating the challenges of detection and reconstruction of neurons from electron microscopy images. (a) Weak, cluttered or even no local membrane evidence at the locations pointed by blue arrows; (b) Complex appearance inside the cytoplasm of neuron cells; (c) other organelles (i.e., mitochondria (red) and vesicles (green)) having membranes with similar appearance features; (d) High variation in the shape of neuron structures.

4.3 Previous Work

Existing attempts at automatic segmentation of neural tissue from EM images can broadly be divided into two categories. i) Pixel-wise classification based approaches try to identify cell membranes. ii) Region based approaches start with initial *superpixels* and merge small regions. Supervised learning methods have been focusing on neural network and decision forests based frameworks to achieve accurate boundary detection [68], [69, 70], [71].

In [68], a multi layer convolutional neural network is utilized to classify pixels as foreground and background. Their method presents two critical properties: i) the classification filter for each layer is obtained directly from the data, and ii) the multiple convolutions throughout the layers of the network provide an indirect filter effects. In their work the neural network contains huge number of parameters and therefore is computationally intensive and requires very large training set.

The approach of [70, 69] is based on a key concept of using nested artificial neural networks (ANN). To improve the membrane detection, the performance of ANN can be boosted by incorporating learned membranes from sequential sections and applying tensor voting based post-processing. Their method avoids the expensive computation of the filter banks which are traditionally used to extract pixel-wise features. However their final performance is affected very badly since they consider only local image feature/information.

In more recent works [72] and [73], a convolutional network is used to generate affinity graphs for image segmentation. An affinity graph is a graph whose nodes correspond to image pixels, and edges reflect the affinity between them. Graph-based segmentation algorithms solve the problem by constructing and then partitioning an affinity graph. The error measure of boundary detection is given by edge misclassification rate of affinity graphs; however this is not the only criteria directly renders the quality of segmentations produced by partitioning the affinity graph. In [72], different than previous works, a segmentation performance measure called *Rand Index* has been used in the cost function of affinity learning stage. By using a conventional graph partitioning algorithm they find the connected components of the thresholded affinity graph, and train an affinity classifier that directly minimizes the Rand index of clusters resulting from the graph partitioning. In [73], the affinity weights of a graph is learned by training a convolutional network. Here, the affinities are estimated by classifying large number of image features in a gradient descent procedure while adjusting network parameters. Thus this method needs long training times or fast GPU implementations in practice.

To reduce computation time and the memory consumption of optimization methods employed on affinity graphs, sparsification of the affinity graph has been used as a solution. Important examples of this approach can be given as [74, 75, 76, 77, 71, 64, 62, 78, 79, 80]. In these approaches the nodes of the graph represents *superpixels* (small regions) and the affinity is defined as the relationships between regions. These approaches can be divided into two categories based on the way the relationships are encoded. The first category [74, 75, 76, 77] utilize regular loopy graphs where spatially connected regions are linked to each other with undirected edges. These methods come up with approximate inference algorithms such as integer linear programming and loopy belief propagation. To overcome the errors in boundary misclassification, many of them employ procedures based on statistical learning or topology-preserving constraints. The second group [71, 64, 62, 78, 79, 80] actually use hierarchical representations of regions. The main difference of these approaches from the first category methods is based on the assumption that the hierarchy of regions can actually encode the topological constraints inherently and is content driven. If this assumption is true, than very efficient inference or classification algorithms can be employed on that hierarchical representation. These efficient algorithm are very effective when processing immense data.

We have mentioned that custom softwares have been designed for proofreading dense neuron reconstructions [63, 64, 65, 66, 67]. The common property of these approaches is that they all target reducing manual effort. The way they simplify the interaction process mostly rely on specific interface and visualization designs such as mapping 2D merge and split operations to real time 3D reconstructions or highlighting only few region of interest one at a time based on some defined priorities. These priorities are usually obtained by utilizing learning methods and represented as confidence measure for user consideration.

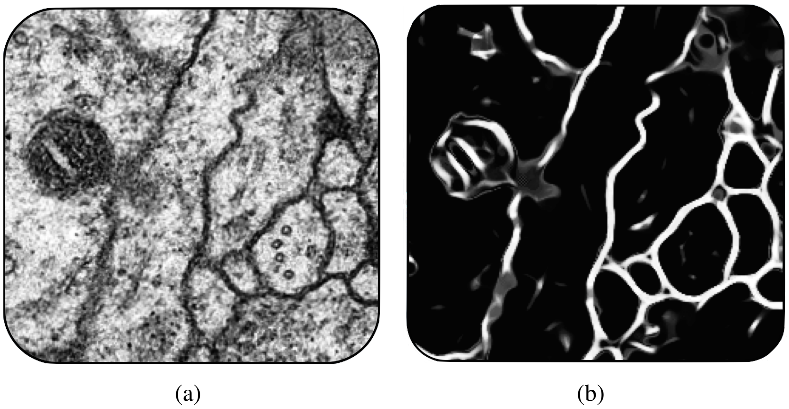
Chapter 5

Multi-Stage Decision Strategy for Neuron Segmentation

In order to extract useful information from EM images, such as segmentation, it is compulsory to characterize regions structurally, as well as contextually. For that reason, we propose a multistage decision mechanism that utilizes underlying differential geometric properties of objects in a biologically inherited framework. Consequently, we start with an initial feature selection procedure to select most relevant features to characterize distinct regions, such as membrane, cytoplasm and outliers. Similar to a topographic map, a random-forest classifier is employed to highlight mountain ridge like structures, e.g. membranes as well as plateaus, e.g. cytoplasm. In order to extract the underlying geometry of structures on this topographic map (especially membrane like structures), principal surface analysis is utilized. This unsupervised technique returns highly sparse yet accurate low dimensional representation of the data and especially characterizes membrane like regions. A task specific, second stage decision mechanism is employed to distinguish contextually different mitochondria and cell boundary membranes. This second stage learning/decision mechanism is based on the appearance, the initial topographic map with its low dimensional reconstruction and expert supervision on different types of membranes. Initial results on individual EM slices indicate that the proposed approach can successfully segment objects with minimal expert supervision and can potentially form a basis for a larger scale volumetric data interpretation.

5.1 Introduction

In 2d stack of ssTEM images, in order to reconstruct neuronal arbors, one requires to identify the boundary (cytoplasmic membrane) that encapsulates adjacent neurons. Identification of boundary from EM images is not a trivial task due to artifacts related to the chemical fixation process in the tissue preparation and the limitation in the section thickness as a result


(a)

(b)

Figure 5.1: Illustration of a random forest classifier performance on membrane prediction. Training is based on two classes: membrane vs. non-membrane. (a) Input image patch; (b) prediction result. Bright color means higher probability for being membrane.

of anisotropic resolution [Albert Cordona, personal communication]. Extracting the correct boundary of interest needs grouping relevant boundary components together which are scattered on image domain.

Training a classifier based on the prior knowledge of the shape and the appearance of the membranes is a common approach to highlight cell boundaries. Such a classifier is ideally expected to highlight only cytoplasmic membrane regions; however, other elements in a cell such as vesicles or mitochondria have their own inner/outer membranes, thus make the recognition task more challenging. Fig. 5.1 shows resulting response (b) of a trained random forest classifier on a sample image patch (a). Here the classifier is trained using two labels cytoplasm membrane vs. non-membrane. Notice how the membranes of other structures (mitochondria and vesicles) are equally well highlighted with the actual targeted membrane pixels.

For that reason, in this chapter, we propose a framework composed of two stage decision mechanisms that can identify accurate cytoplasmic membranes (we will call this as the cell boundary in the rest of the chapter). First we use random forest classifier to highlight all possible membrane regions. For each pixel, three membership probabilities are assigned to construct a multi channeled probability map of the image. Then, we compute a sparse representation of the image via ridge point detection method to find samples on the ridge of the probability map that is sufficiently dense to estimate the embedded underlying structure. Next, the sparse ridge points set is categorized into different classes based on their relative spatial distribution and appearance. The spatial distribution and appearance of ridge points are embedded in an affinity graph whose nodes are composed of ridge points and edges are determined via Delaunay triangulation. The categorization / classification of these nodes is utilized by a second stage classification followed by a two step label propagation processes in which the nodes are relabeled while maximizing the significancy level difference between each label. Finally, the cytoplasmic regions and their surrounding membranes are segmented on the image domain by morphological post processing.

5.2 Related Work

There have been efforts on bringing high level contextual information into consideration for accurate boundary delineation. Graph cut segmentation algorithm is one of the global optimization based solver of the affinity graphs via which, both local, nonlocal and contextual constraints can be incorporated into optimization. For example, in [81], the flux of the gradient vector field has been incorporated into graph cut approach as a solution to prevent gaps in segmentation of thin and elongated boundaries. As a more specific work to ssTEM images, we refer [82, 83] as instances which uses the flux of gradient vector field in segmentation of cytoplasmic membranes. They introduce perceptual grouping constraints to complete the gaps on the membranes. In particular, [83] combines flux of gradient vector field with probability output of a random forest classifier in a regular graph cut energy (cost) function. The solution found by graph cut optimization is global. The flux of gradient vector field is thought as the gap completion regularizer. In this work, choosing correct combination parameters balances the effect of different energy terms. This method is a solution to the local contrast variation in membrane pixels; however further manual improvement is needed since the gradient flux is reported to introduce a large amount of false positives when the image gradient is remarkably high at the undesired image regions in addition to the target segmentation borders. Also applying to large 3D volumes need too much computation power and memory.

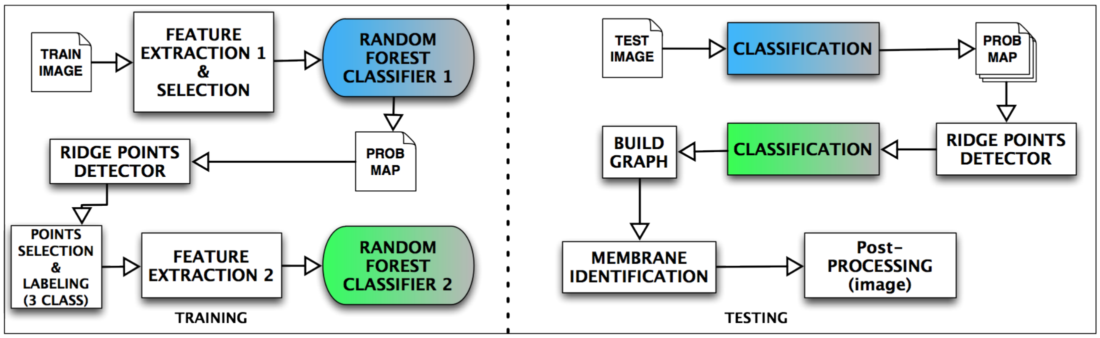


Figure 5.2: The block diagram of our workflow with its training and test parts.

5.3 Membrane Detection and Identification

In this section, we give the details of our approach and also summarize the overall flow via diagram shown in Fig. 5.2. Briefly, our approach is composed of training and testing sessions. In the training, two classifiers are built. The first classifier is employed on the pixel domain of the image while the second is on the intrinsic membrane manifold. Having the two classifiers prepared, given a test image, first a membrane detection, then the ridge point detection for low rank representation of the data, graph construction from low rank representation and the membrane identification on this graph is applied consecutively. Finally, morphology based post processing is utilized to obtain the ultimate segmentation.

In order to highlight membrane like structures in EM slices, we used random forest classifier [83]. We start with a feature selection procedure to select the most relevant features with respect to the groundtruth on a training set. Initial pool of features is a mixture of appearance, e.g. edge, min/max intensity and structural, e.g. curvature, filtering responses evaluated at different scales (in total 243 distinct features). We estimated the mutual information of a feature [84] with groundtruth and selected the first 4 features (3 different scales of anisotropic diffusion and Gabor filter response). The total number of the features are arbitrarily selected based on the visual inspection.

In order to extract accurate boundary from EM images, one needs to distinguish cell boundary from the rest of the elements. However, note that, any element that resamples the cell boundary in terms of appearance or structure will also be miss classified. Moreover, cell boundaries are not always smooth and might take arbitrary shapes, especially around synapses. Consequently, we approach the recognition task as a three class problem where we highlighted *i*) cell boundary, *ii*) cytoplasm, and *iii*) ambiguity regions in a training set. Ambiguities are locations where other membrane types such as mitochondria inner/outer membrane or synapse locations are present. In fact, the major challenge in boundary decision is to develop efficient and accurate algorithms for large volumes of data to categorize such ambiguity regions into cell membrane or cytoplasm. For that reason, we calculate the low rank membrane representation that accurately governs the underlying biological structure, yet provides sparse representation of the data.

Fig. 5.3-a shows the result of the random-forest classifier on a sample test image. Each color depicts the probability of having one class where red, green and blue represent cell membrane, cytoplasm and ambiguity regions correspondingly. In order to sparsely reconstruct a low rank representation of the data, we use the membrane probability map (red channel) and estimate the intrinsic structure of the membranes. Unlike previous approaches that partition the probability map into arbitrary regions, e.g. super-voxels, our sparse reconstruction of the data inherits the differential geometric properties of the cell tissue.

In order to obtain a sparse representation of the data, we utilized nonparametric principal curve projections [85, 86]. Intuitively, our goal is to find samples on the ridge of the probability map that is sufficiently sparse to efficiently analyze the data, yet dense enough to estimate the embedded underlying structure, i.e. curve. In fact, cell membranes are 2D surfaces embedded in 3D volume, however, we restrict ourself to the estimation of curves in 2D slices due to the high anisotropy in the data. For that purpose, we used the first and second derivatives of a ridge regression function jointly in order to find the ridge locations where gradient becomes orthogonal to the maximal curvature direction[1]. Fig. 5.3-b shows the overlay of the detected point locations on the ridge of the cell membrane probability map.

A crucial recognition step in membrane recognition is to analyze the data in the context and to classify ambiguities in the data into correct elements (cell membrane/cytoplasm). Pairwise

[1] Maximal curvature direction is defined as the eigenvector of the local Hessian matrix that has the largest absolute eigenvalue. For details see [85].

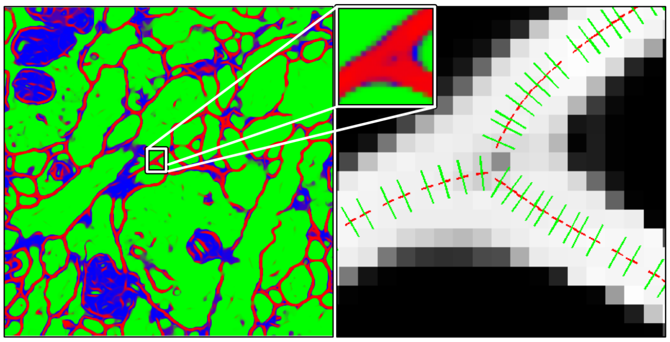


Figure 5.3: (a) Result of 1st stage random forest classifier on a test image. (b) A closer look to the detected point locations on the ridge of the cell membrane probability map. Note that the probability map is just the red channel of image in (a). Unit vectors in the tangent space (red) and the vectors (green) that are orthogonal to the tangents are also rendered.

similarities with the neighboring elements, appearance, and shape are necessary features to decide if an ambiguous location in the data is cell boundary or not. In order to identify different types of membranes and uncertainty regions, we classify the projected membrane points into three: isolated (L_1), transition (L_2) and inner samples (L_3) based on their cross-section probability profiles. This grouping is done via second stage random forest classifier which is trained on the probability profile along the normal direction of the tangent space of each projection point. Fig. 5.4 shows common uncertainty regions (dense blue regions) where the recognition task is not trivial. Fig. 5.4-a displays the 3 channel probability map obtained from the first stage random forest classifier. Orthogonal profiles of the probability values across isolated, transition and inner (cell, mitochondria outer and inner membranes with their associated numbers respectively) are overlaid on the probability map on the left. Similarly, membranes in the transition and inner regions of the synapse location are depicted on the right. In general, membranes have distinct profiles based on the spatial location in the cell. An isolated sample lies between cytoplasmic region, whereas a sample in the mitochondria or synapse is encapsulated by blue channel (uncertainty region). Lastly, a sample in transitional region is on the boundary between these two.

Motivated by the aforementioned observations, we formulate the membrane identification problem as a competitive label propagation, operating over a graph network which is well matched with our sparse representation.

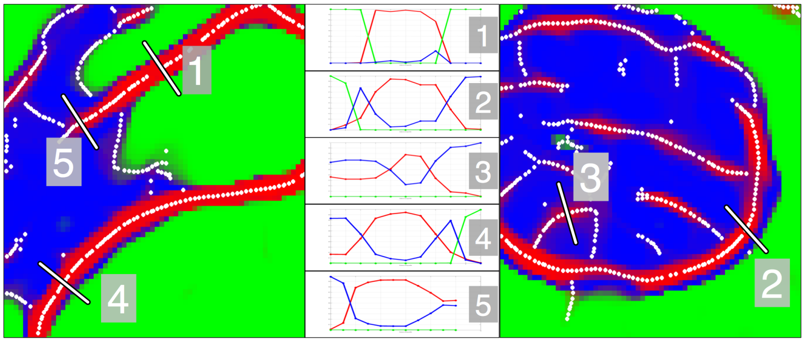


Figure 5.4: Illustration of feature vectors for the three classes: L_1 (1) partition cytoplasm into distinct clusters; L_2 (3, 5) samples are generally observed in uncertainty regions (dense blue); L_2 (2, 4) samples can be found between cytoplasm and synapse/mitochondria transition.

Let $\mathcal{V}$ be the set of projection points $L := L_1 \bigcup L_2 \bigcup L_3$ and E be the pairwise edges between them with constructed sparse graph $\mathcal{G} = \langle \mathcal{V}, E \rangle$. Edges between samples are determined by constrained Delaunay triangulation, where any edges that passes through high probability cytoplasmic regions are deleted from the graph. In the proposed competitive label propagation process, two classes: isolated (teal) and inner (yellow) propagate in the network and compete each other to occupy the transition (brown) class nodes. Starting from an initial state $L^{t=0} = \bigcup L_i^{t=0}$, $i = 1 \ldots 3$ and $\Phi(t = 0)$ the iterative competition model grows a dynamic front to span all unlabeled nodes ($j \in L_3$) in the graph to cover the whole data. The decision of an unclaimed label at iteration t is defined as the following optimization:

$$\begin{aligned} \tilde{L}_3^t(j) \quad &\Longleftarrow \quad \arg\max_k \big\{ |(\mathcal{N}_{k,t}(j))| \big\}, \quad k \in 1, 2 \\ \text{subject to} \quad & \mathcal{N}_{k,t}(j) \in L_1 \cup L_2 \\ & j \cap \mathcal{N}_{k,t}(j) \neq \emptyset \\ & \Theta(\mathcal{N}_{k,t}(j)) < \Phi(t) \end{aligned} \tag{5.1}$$

where optimization is formulated as majority voting in the graph. Here, $|(\mathcal{N}_{k,t}(j))|$ is the cardinality of the neighborhood of j having class label k at iteration t. First and second constraints indicate the propagating front of the competitive voting and the third constraint indicates the current state decision level $\Phi(t)$ at iteration t. In our competitive voting model, state decision level is a monotonically non-decreasing function and is a measure of anisotropy in the nodes. More clearly, as the front grows and occupies, unlabeled nodes that are intrinsically similar

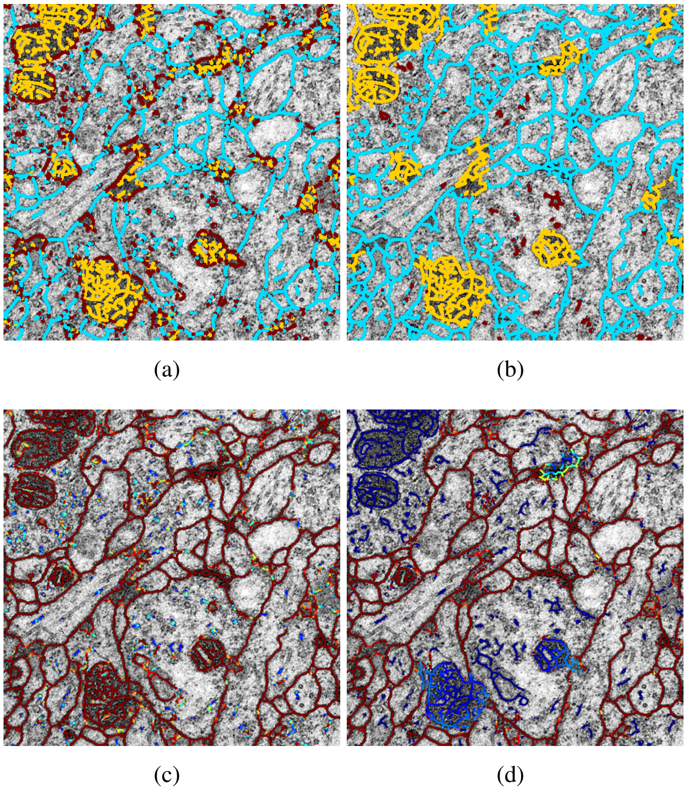


Figure 5.5: (a) A classification result via second-stage random forest classifier based on the features shown in Fig. 5.4. (b) Projection points with their label colormap. Points classified as cell boundary are shown in blue and the others in yellow. Red points are discarded since they have no valid connection to other points on the constrained graph. (c) Projection points rendered with their initial affinity colormap. (d) Projection points rendered with their adjusted affinities.

to the labeled instances are ideally favored. Estimated low rank representation of the data is utilized to align the tangent space of an unlabeled instance with a labelled one, hence highlight the local anisotropy in the graph. In order to align samples, dissimilarity/divergence of samples from the underlying structure is calculated as the total angle, $\Theta()$ between the aligned tangential spaces. The total angle is calculated as the sum of the interior angles between the tangential vectors and the edge between nodes. At a given decision level $\Phi(t)$, front continues to grow as long as there exits samples in the feasible set of the above optimization. After a single pass through all samples in L_3, $\Phi(t)$ level is relaxed with an arbitrarily selected percentage until iterative search process described above is finished.

Affinity Propagation and False Positive Removal The competitive label propagation method returns structurally meaningful elements and recognition task requires interpretation of them. For that reason, we analyze the morphology of elements, i.e. mitochondria and synapse, that are not returned as cell boundary. For this purpose, we employ a region growing algorithm that distinguish mitochondria from synapse using contextual information such as shape. The region growing is run on the sparse graph that we previously constructed. We constructed an affinity measure between nodes as a combination of likelihood and geometrical terms. Geometrical term is defined as the alignment of the tangential vectors between nodes. The assignment of likelihood term for cell boundary points (teal) is based on the red channel of the result of the first stage random forest classifier (see Fig. 5.5-c). The assignment of likelihood term for non-cell boundary (yellow) points is done by using an experimentally learned shape model of mitochondria structure that is obtained in an offline training processes. The shape model is composed of solidity and curvature [87] properties of the structure of interest. The shape information of regions under the yellow points are extracted from the binary image I_{bin}: $D_{t \to y} \geq D_{y \to b}$, where $D_{t \to y}$ is the geodesic distance map from teal points to yellow points and $D_{y \to t}$ is the opposite. For each connected component in I_{bin}, a decision tree is used to decide if it can be a mitochondria. At each node of the decision tree the priori learned range of solidity ($\mathcal{S}$) and curvature (κ) values are used to give a decision. For each yellow point enclosed in I_{bin}, the likelihood is computed as $\alpha = (1 - \mathcal{S} + \kappa) \cdot D_{t \to y}$. Among the leaf nodes of the tree, for the candidates that are likely to be mitochondria, a score is computed as follows: $\gamma = 1 - \mathcal{S} + \kappa$. For each yellow point enclosed in the candidate region, the likelihood is computed by weighting the score γ with the continuous normalized geodesic distance $\bar{D}_{t \to y} \in [0, 1]$ as: $\alpha = \gamma \cdot \bar{D}_{t \to y}$.

Before running the region growing, to avoid from capturing false positive non-cell boundary nodes, the affinity values for the nodes that are classified as non-cell boundary (yellow points) have to be adjusted since the affinities on mitochondria boundaries are very similar to cell boundaries (see Fig. 5.5-c). In this scheme, new affinities are computed based on a descriptor which is function of both high level shape information e.g. curvature and solidity, of the underlying regions of the non-cell boundary labeled (yellow) points. In Fig. 5.5-c and d, the merit of this likelihood adjustment scheme can be observed clearly. Accordingly, the points in the mitochondria regions are distinguished from synapse regions by adjusting new likelihood values.

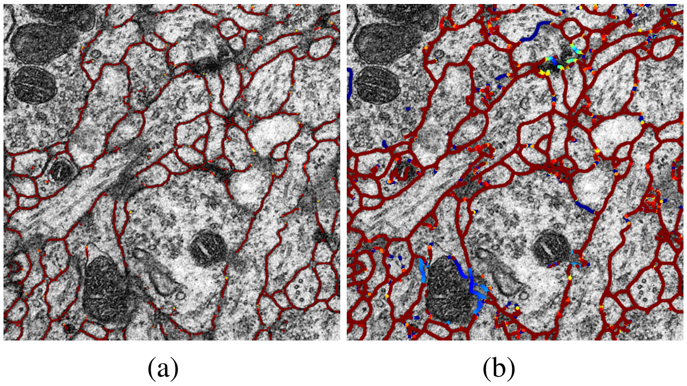


Figure 5.6: (a) Initial set of nodes that are chosen above a very large affinity threshold. Note that points are rendered with affinity colormap. (b) Final cytoplasmic boundary points obtained after region growing on the sparse graph.

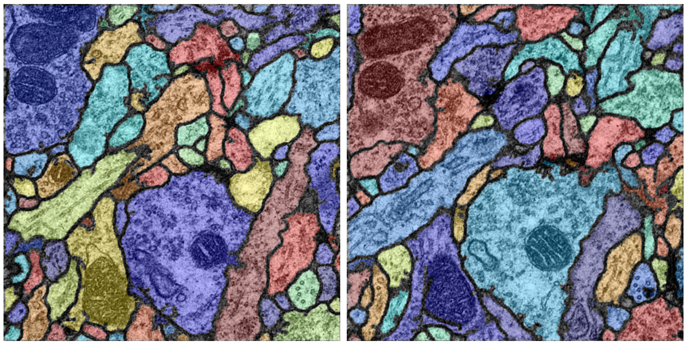


Figure 5.7: Final segmentation results for an example section from the test set.

Region growing starts from an initial set of nodes (see Fig 5.6-a) which are selected as the nodes having affinity greater than a very large threshold and iterates until convergence. The resulting cell boundary points are obtained as shown in Fig. 5.6-b. Given the final cell boundary points, segmentation of encapsulated neurons with morphological post processing steps is straightforward. Finally, we summarize and visualize the whole pipeline of our membrane identification based region partitioning approach in Fig. 5.8.

5.4 Experiment & Results

As part of the ”ISBI'12 Segmentation of neuronal structures in EM stacks” challenge, the method was evaluated on 30 sections from 2d stack of ssTEM images of Drosophila first instar larva ventral nerve cord. The microcube measures 2 x 2 x 1.5 microns approximately, with a resolution of 4x4x50 nm/pixel (see [88] for the details of the data set). We trained our

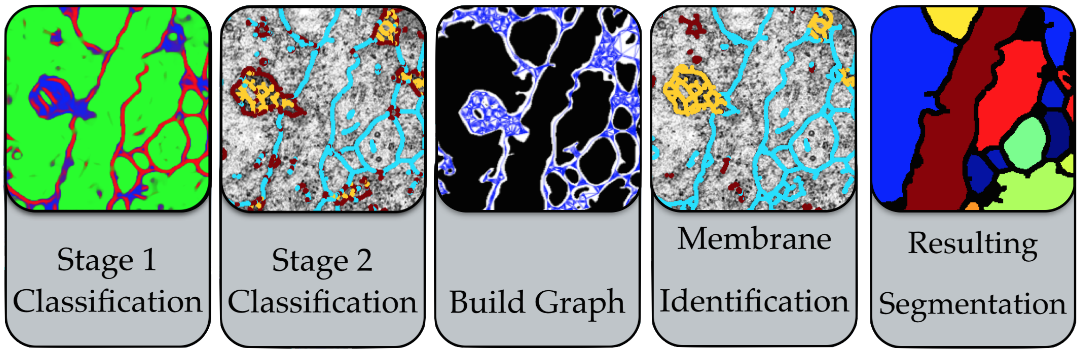


Figure 5.8: The visual summary of our framework.

classifiers on arbitrarily chosen 20 sections out of 30 training images. In order to evaluate the performance, Minimum Splits and Mergers Warping error, Rand error and the Pixel error were used (see [89] for the details of evaluation metrics). The performance of our method was reported as: $\%16.2303, \%0.1613$ and $\%10.939$ rates for Rand error, Warping error and Pixel error, respectively for all images in the test set. In Fig. 5.7, example segmentation results on two sections from the test set are shown.

5.5 Conclusions

We have underlined many challenging facts about the reconstruction of neuron structures from ssTEM images and proposed a two stage decision mechanism for segmentation of neuron membranes by using both low level (differential geometric) and high level contextual properties of biological elements. We defined a membrane identification problem which can be solved over a sparse connectivity graph and proposed an iterative competition based label propagation method for the membrane identification. We demonstrated results on 30 sections of the test data of ISBI'12 EM Segmentation Challenge and reported the accuracy of the proposed method. Future work includes extending current 2D label propagation into 3D with global optimization extensions.

Chapter 6

OpTree: Learning Adaptive Watersheds for Neuron Segmentation

In this chapter, we present a new algorithm for automatic and interactive segmentation of neuron structures from electron microscopy (EM) images. Different from our previous approach, this method aims a globally optimal solution to the image partitioning problem. We propose a learning based segmentation model providing both automated reconstruction and efficient interactive proofreading mechanism. More specifically, our method selects a collection of nodes from the watershed merging-tree as the desired segmentation. This is achieved by building a conditional random field (CRF) whose underlying graph is the merging-tree. The maximum a posteriori (MAP) prediction of the CRF is the output segmentation. Our automatic algorithm outperforms state-of-the-art methods. Both the inference and the training are very efficient as the graph is tree-structured.

The domain of neuron segmentation requires extremely high segmentation quality. Therefore, proofreading, namely, interactively correcting mistakes of the automatic method is a necessary module in the pipeline. Based on our efficient tree-structured graphical model, we develop an interactive segmentation framework which only selects uncertain locations for a user to proofread. The uncertainty is measured by the marginals of the graphical model. Only giving a limited number of choices make the user interaction very efficient. Based on user corrections, our framework modifies the hierarchical-tree and thus improves the segmentation globally.

6.1 Introduction

We propose a CRF based method whose underlying graph is constructed by the output of *watershed transform* [90]. The watershed transform partitions a given image into *superpixels* by simulating a water flooding of the landscape of a given scalar function, e.g. the gradient magnitude or the likelihood of each pixel being the boundary. These over-segmented regions usually

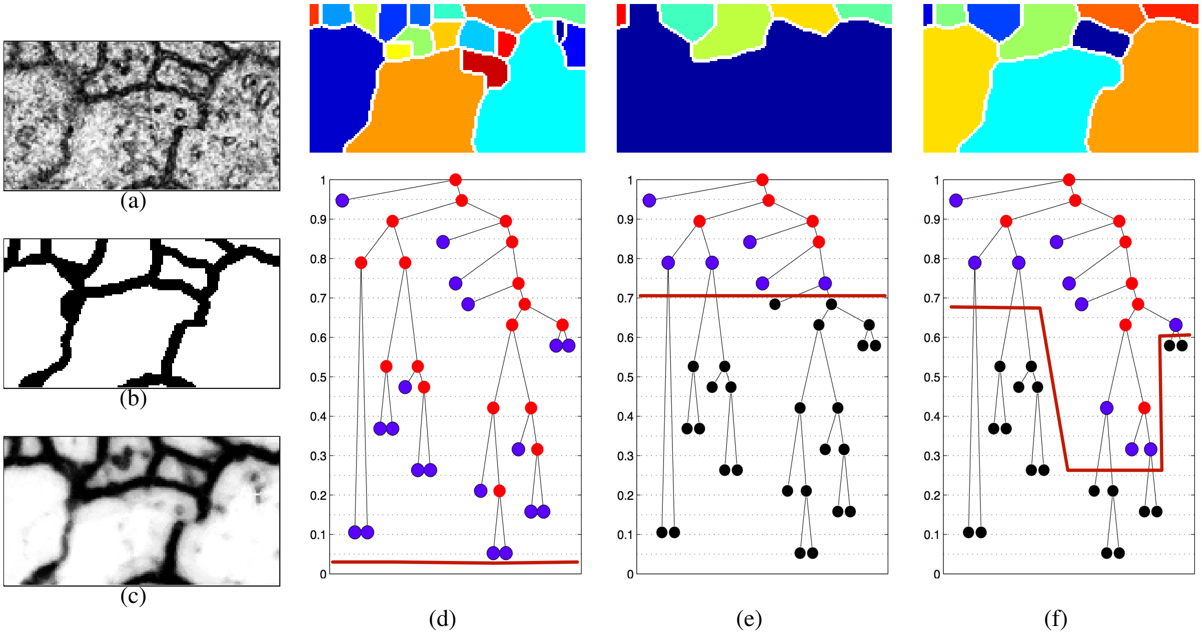


Figure 6.1: (a) The EM image patch; (b) the ground truth; (c) the boundary likelihood map (dark pixels have high values); (d) the watershed segmentation and its tree, built using the boundary likelihood map as the landscape function; (e) the watershed segmentation with a higher threshold; (f) the result of our algorithm.

form a representation of an image that is much more compact than the original pixel grid.

In order to mitigate the over-segmentation effect, one often merges neighboring segments when the minimal function value along the boundary between them (called the *saliency*) is below certain threshold. Considering all saliency thresholds, a hierarchical *merging-tree* is constructed [91] in which each leaf node is a segment of the original watershed and each non-leaf node is a merged segment. A *height* function can be assigned to each node according to the minimal saliency threshold at which it disappears (is merged with others). The watershed segmentation at any given threshold can be computed by cutting all tree nodes below the threshold and taking all leaf nodes of the remaining tree (see Figure 6.1(d) and 6.1(e)).

We use the watershed tree as the underlying graph of our graphical model. The MAP solution of the graphical model returns a subset of the nodes of this hierarchical tree that cuts the tree at different height levels or adaptively. In Fig. 6.1, the advantage of this adaptive cut is illustrated with an EM image segmentation scenario where watershed segmentation fails but ours doesn't. We show an EM image patch (a), its target segmentation (b), its boundary likelihood map (c) on which watershed is computed. The representations of the watershed merge tree for different cuts and their corresponding segmentation results (e), (d), (f) are also shown. Running watershed using a certain threshold usually leads to accurate segments at certain area yet over/under-segmentation at other areas (see Fig. 6.1(d) and 6.1(e)). However; our CRF-based learning algorithm finds a segmentation of higher quality by selecting different saliency thresholds at different areas of the image. Essentially, our algorithm learns from training data how to cut a hierarchical tree adaptively to achieve a better result. See Figure 6.1(f). Our automatic segmentation method is not only accurate, but also very efficient. The tree structure of the graphical model allows us to compute exact MAP inference and the marginals very fast. Our method outperforms state-of-the-art in automatic segmentation of 2D (ssTEM) [92] and 3D (FIBSEM) EM images.

6.2 Related Work

The watershed method and its variations have been used on EM images [64, 75, 76, 77]. They are graph based algorithms and in these approaches the nodes of the graph represents *superpixels* (watershed catchment basins) while the edges corresponds to the affinities defined as the relationships between adjacent superpixels. In these works, the problem is formulated as a labeling problem of the boundaries between adjacent small regions under certain topological constraints. In [75, 76], an image is represented as an adjacency graph of supervoxels. Edge weights indicate that how likely the incident regions should be merged. An optimal segmentation makes binary decisions for each edge so as to minimize the total cut weight, subject to the constraint of producing a topologically consistent solution. The problem is solved using multi-cut integer linear programming. In these two approaches the quality of the final segmentation depends critically on the edge weights. These edge weights are scalar functions of some features of the raw data (classifier output, e.g. random forest). So the final result is sensitive to the classifier performance. Recently, [77] extended this multi-cut approach with a structured learning paradigm to learn these weights using a cutting planes approaches. In an iterative learning scheme, a structured loss is used to compare the segmentations obtained from the current weights to the gold standard to train a structured SVM.

Besides the loopy graph representation, we emphasize that hierarchical tree based representation of superpixels is a popular tool in computer vision. It has been used in many image labeling tasks as proposed in [93, 94, 79, 95, 96]. The common point of these approaches is using the hierarchical representation as limiting factor on the solution space so that the inference tasks are easier to solve. Among these approaches [95], [79] and [96] are designed for image segmentations while others aims semantic labeling of the pixels.

In [95], the initial set of small regions are grouped into super-regions using a region graph. The super-regions in-turn form a region-graph that can be segmented again. Successively applied, a bottom-up tree of regions is computed. This tree-structure allows selection of the desired segmentation at any desired detail level. In [79], a hierarchical merge tree structure is used to represent the merging of multiple region hypotheses. They use supervised classification techniques to quantify the likelihoods of the hypotheses to be a good segmentation. They impose topological constraint as post processing (heuristics) to obtain valid segmentations. In [96], the authors address the requirement of post processing to solve a hierarchical segmentation problem. In order to avoid heuristics, they provide a framework called hierarchical cuts that formulates an optimization problem whose global minimum corresponds to segmentation. They optimize an energy defined as in graph cuts [47]. Their approach has similarities with our framework in terms of providing globally optimal solutions. However, there is no learning aspect of it.

Superpixel representation has also been used in agglomerative clustering algorithms as in[80]. In that paper, an active learning method is employed to learns how to merge watershed segments in a hierarchical scheme. Another learning approach [97] learns to construct a hierarchical tree so that the watershed cut at a certain threshold produces a high quality segmentation. There are many other learning based methods for Connectomics segmentation task. See [61] and references therein.

The ultimate goal of Connectomics is reconstructing the wiring diagram of brain at nanometer resolution. Such image data includes billion of nerve cells and their interconnections [98].

Extremely large size of image volumes requires semi-automatic segmentation approaches because the complex structure of the massive data makes the automatic detection and identification of branching of neurons very difficult. Powerful proofreading tools are crucial for enhancing segmentations efficiently and effectively. According to statistical analysis of existing proofreading tools in [99, 100], correction of automated results by hand requires 120 manual input actions per cubic micron of tissue. This suggests that, conventional interactive segmentation tools cannot be applied in this domain. One example tool specifically designed for *Connectome Project* is Raveler [101]. It is a stand-alone proofreading tool developed by Janelia Farm Research Institute for expert users specific to this domain. It has important visualization features towards making the interactive work easier and effective especially in 3D; however, more recent studies and projects [102, 99] have shown that proofreading will have to be crowdsourced in the future. This implies the importance of distributed settings by non-domain-experts and inexperienced users. In fact, this reminds us the potential of learning based interactive frameworks that could be easily used by non-expert users (i.e. people on the cloud). Our method that we explain in this chapter aims such a learning model using tree-structured CRFs which will be later shown to be a perfect match for efficient interaction. Our model is not the first one that exploits hierarchical trees. In [79], a hierarchical tree representation is employed similar to our approach.This method has been developed in parallel to our work and has certain similarities. Their model is also based on watershed merge tree and besides automatic segmentation, they also propose an extension for proofreading. However, there are clear differences between the two models. The main difference between [79] and our method is that their overall pipeline is composed of two parts: intra-section segmentation and inter-section reconstruction. Their segmentation method is designed to work in 2D images and it can provide very accurate results only when combined with a reconstruction step that needs information from adjacent slices. Our model is generic and can work in 2D and 3D images without any data type specification. Also, to acquire intra-section segmentation their method uses multiple hypotheses generation and supervised classification techniques on the merge tree to generate candidate regions. By evaluating the potentials of candidates and applying consistency constraints they resolve the final results. For inter-section reconstruction they use supervised linking procedures. Compared to our model, their framework includes too many parameters and requires fine tuning to achieve good results on a certain test image. On the other hand, our method is composed of a single elegant graphical model that is naturally parameterless and can be used in any dimension as features are provided.

6.3 Hierarchical Tree-Derived Segmentation

For any given image, we construct a hierarchical-tree by running watershed algorithm on a boundary likelihood map, namely, the likelihood of whether a pixel belonging to the boundaries between neuron cells (Figure 6.1(c) and 6.4(b)). More details can be found in Section 6.6. Next, we construct a tree-structured graphical model whose underlying graph is the same as the merging-tree. The problem of computing the optimal segmentation is transformed into the problem of computing the optimal labeling of this graphical model (Equation (6.3)). In order to achieve the transformation, we need to build a correspondence between segmentations and labelings.

Let $\mathcal{P} = \{p_1, \ldots, p_M\}$ be the set of all superpixels corresponding to the leaf nodes of the tree. We define a *segmentation* as the decomposition of the set of superpixels into a disjoint set of segments S. Formally, a segmentation is

$$S = \left\{ s_1, \ldots, s_m \mid s_i \subseteq \mathcal{P}, \forall i; \bigcup_{i=1}^{m} s_i = \mathcal{P} \right\} \tag{6.1}$$

As a special case, the segmentation is the original watershed segmentation when each segment s_i contains one single superpixel (see Figure 6.1(d)).

Let $\mathcal{S}$ be the space of all possible segmentations. For a given image I, we would like to find the most probable segmentation $\text{argmax}_{S \in \mathcal{S}} P(S|I)$, with a suitably defined posterior probability $P(S|I)$. Since $\mathcal{S}$ is too large to search through, we restrict the solution space to a smaller subset. Given a hierarchical tree, T, a segmentation S is T*-derived* if and only if each segment $s_i \in S$ is a node of T. We call such segmentations *tree-derived* and our algorithm search through the space of all T-derived segmentations, denoted by $\mathcal{S}_T \subseteq \mathcal{S}$.

The correspondence between tree-derived segmentations and the labelings of T is provided as follows. Let the label set $\mathcal{L} = \{-1, 0, 1\}$, represent whether each tree node is an under-segment, a segment or an over-segment. An under-segment node is the ascendant of a set of segment nodes. An over-segment node is the descendant of a segment node. For each labeling y, we take the set of zero-labeled nodes as the corresponding segmentation. In Figure 6.1(f), colors of the tree nodes represent a labeling. Red, blue and dark colors correspond to -1, 0 and $+1$ labels respectively. The corresponding segmentation is shown in the top row.

We are only interested in labelings that derive legit segmentations. Therefore we enforce certain restrictions on the labelings. For a labeling y and a node v, let $\Gamma_v(y)$ be the sequence of labels along the path from v to the root.

Tree-Segmentation Bijection Theorem. There is an one-to-one correspondence between $\mathcal{S}_T$ and the set of labelings $\mathcal{Y}_T$, such that for any labeling $y \in \mathcal{Y}_T$ and any leaf node v,
(1) $\Gamma_v(y)$ is monotonically non-increasing;
(2) The zero label appears exactly once in $\Gamma_v(y)$;
(3) The first label (label of v) cannot be -1 and the last label (label of the root node) cannot be $+1$.

Tree-Segmentation Bijection Theorem Proof. We call a subgraph of T a *rooted full subtree* T', if
(i) T' is a tree;
(ii) T' contains the root of T as its own root;
(iii) T' is *full*, namely, each internal node has two children within T'.

For a segmentation labeling, the leaf nodes of any subtree T' is labelled 0 and called *tree-cut*, while the root node and other internal nodes are labelled -1. It is straight forward to see the bijection between the set of segmentation labelings and the set of *tree-cut*s. Since, the subgraph using the union of the nodes of T with label -1 and 0 is a rooted full subtree, it remains to establish a bijection between the set of all rooted full subtrees, $\mathcal{T}_T$, and the space of tree-derived segmentations, $\mathcal{S}_\mathcal{T}$. Notice that any two nodes of T are either disjoint or nested, namely, one contains the other. The latter situation is true if and only if the bigger node is on the path from the smaller node to the root.

We construct a mapping Φ from $\mathcal{T}_T$ to $\mathcal{S}_T$, and then prove it is a bijection. Given a full subtree $T' \in \mathcal{T}_T$, we define $\Phi(T')$ as the *tree-cut* of T'. To see that $\Phi(T')$ belongs to $\mathcal{S}_T$, we need to show two things. First, any two elements of $\Phi(T')$ are disjoint. Otherwise, the bigger node must be on the path from the smaller node to the root, and thus cannot be a leaf node of T'. Second, for any superpixel v_i (a leaf node of T), there is one element of $\Phi(T')$ on the path from its corresponding node to the root, and thus contains v_i. Otherwise, the first node along the path that belongs to T' must be an internal node of T' with less than two children in T'. This is contradictory to the condition that T' is full.

It is straightforward to see Φ is injective. We conclude this proof by showing that it is also surjective. For any tree-derived segmentation $S \in \mathcal{S}_T$, we construct a subtree T' by taking the union of paths to the root from all elements of S. This T' is a subtree containing the root node. Each leaf of T', *tree-cut* belongs to S. At last, T' is full. Otherwise, there is an internal node of T' with a child $R \in T \backslash T'$. This region R is disjoint from any element of S, contradicting to the fact that S covers the image domain. □

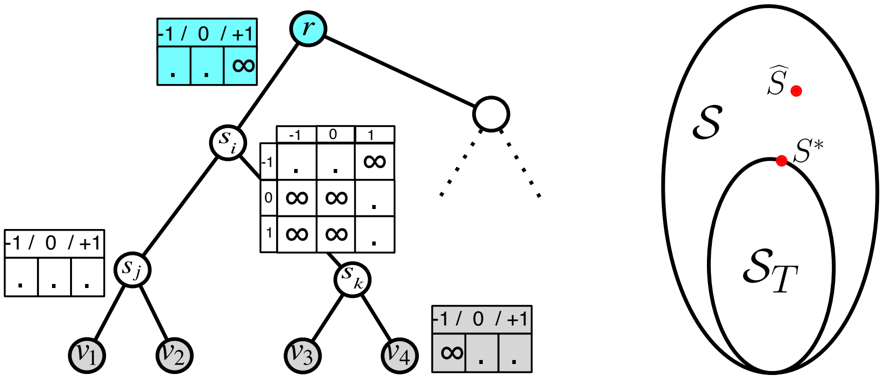
We call labelings within $\mathcal{Y}_T$ *segmentation labelings*. Conditions in Theorem 6.3 can be translated into restrictions on labels of nodes and edges. In particular, a labeling y is a segmentation labeling if and only if (1) the root has label $y_{root} \in \{-1, 0\}$; (2) any leaf node, v, has label $y_v \in \{0, 1\}$; (3) for any child-parent pair, (c, p), $y_c \geq y_p \geq y_c - 1$; (4) if $y_p = y_c$, $y_c \neq 0$. To satisfy the restrictions on the labels we enforce certain potentials to be infinity for the corresponding nodes and edges of the graphical model. In Fig.6.2(a), we illustrate the node/edge potential constraints for segmentation labelings.

6.4 CRF Inference and Training.

For any given image and hierarchical tree, we construct a graphical model. Let the posterior probability of a tree-derived segmentation $S \in \mathcal{S}_T$ be $P(S \mid I) = P(y \mid I, w)$, where y is the corresponding segmentation labeling of S. Given an observation I and a parameter vector w, the conditional probability of a labeling y is

$$P(y \mid I, w) = \exp(-E(y \mid I, w) - \log Z(I, w)), \tag{6.2}$$

where $Z(I, w) = \sum_{y \in \mathcal{Y}} \exp(-E(y \mid I, w))$ is the *partition function*.



(a)

(b)

Figure 6.2: (a) Constraints for segmentation labelings: The root node has infinite potential for label 1. The leaf nodes have infinite potential for label -1 since it only has labels 0 and 1 feasible. For any internal node s_i, and its child s_k in T, the edge (s_i, s_k) has only $(-1,-1)$, $(-1,0)$, $(1,1)$ and $(0,1)$ as feasible label combinations. So, an edge potential $\psi_{i,k}(y_i, y_k)$ is infinite for $\{(y_i = 0, y_k = 0), (y_i = 0, y_k = -1), (y_i = -1, y_k = 1), (y_i = 1, y_k = -1), (y_i = 1, y_k = 0)\}$ label configurations. (b) We find a tree derived segmentation that best approximates the ground truth segmentation $\widehat{S}$.

The energy is often defined as the negative inner product $E(y \mid I, w) = -\langle w, \phi(x,y)\rangle$, in which $\phi(x,y)$ is the concatenation of features of all nodes and edges. Note that the feature vector also depends on y. We compute the MAP, $\operatorname{argmax}_{y \in \mathcal{Y}_T} P(y \mid I, w)$ whose corresponding segmentation is the predicted segmentation. In prediction, one computes the *maximum a posteriori (MAP)* as,

$$\operatorname{argmax}_y P(y \mid I, w) = \operatorname{argmin}_y E(y \mid I, w) \tag{6.3}$$

and the *marginals* as,

$$P(y_i = \ell \mid I, w) = \sum_{y : y_i = \ell} P(y \mid I, w). \tag{6.4}$$

Since the graph is tree-structured, computing MAP and marginals can be solved exactly and efficiently using dynamic programming algorithms. For more general graphs, the inference tasks are NP-hard [7] and have to be solved approximately [7, 103, 104, 105] unless certain assumptions of the potential are made [47, 106].

Assuming the Markov properties, the energy E can be factorized into the summation of unary potentials and pairwise potentials,

$$E(y \mid I, w) = \sum_{i \in \mathcal{V}} E_i(y_i \mid I, w) + \sum_{(i,k) \in \mathcal{E}} E_{i,k}(y_i, y_k \mid I, w). \tag{6.5}$$

Each term corresponds to one type of factor and specifies different roles such as incorporating observations (features) into the model or enforcing a consistent labeling of the variables. For details of this representation and the design of the w parameter vector we refer the reader to [4] and UGM package by Mark Schmidt. For a node/edge, one can forbid a certain label/label combination by forcing the corresponding potential to be infinite. Thus, we can compute MAP and marginals only over the set of feasible labelings $\mathcal{Y}' \subseteq \mathcal{Y}$. This makes it possible to do training and inference only over $\mathcal{Y}'$, as we will do in this paper. Fig. 6.2(a) illustrates an example of node and edge potentials assignment for a constrained label configuration.

We train the parameter vector w from a given set of training data $\{(I^n, y^n)\}, n = 1, \ldots, N$. This can be achieved by minimizing the convex loss function defined as

$$loss(w) = \|w\|^2 + c \textstyle\sum_{n=1}^{N} [\log Z(I^n, w) - \langle w, \phi(I^n, y^n) \rangle]. \tag{6.6}$$

Since its convex, we can find the optimal w using gradient descent. The gradient can be computed efficiently as long as the partition function can be computed efficiently. Thanks to the tree structured factor graph that the partition function can be computed efficiently. We use the UGM package by Mark Schmidt for the w parameter learning [1].

Next we show how to compute the groundtruth labeling for each training data.

6.5 Computing the Optimal Tree-derived Segmentation.

For each training data, we are given a groundtruth segmentation $\widehat{S}$, which may not be tree-derived. In order to find a groundtruth labeling of this data, we find the tree-derived segmentation that best approximates $\widehat{S}$,

$$S^* = \text{argmax}_{S \in \mathcal{S}_T}\, score(S, \widehat{S}) \tag{6.7}$$

[1] `http://www.di.ens.fr/~mschmidt/Software/UGM.html`

(see Fig.6.2(b) for the intuition). For EM images, there are two popular score functions, the random index (RI) and the variation of information (VOI) [107]. We observe that both functions can be decomposed into a summation of scores of elements of S: $score(S) = \sum_{s_i \in S} f(s_i)$. Based on this observation, we provide a polynomial algorithm to compute S^* using dynamic programing. For each node of the tree, we compare whether it is better to keep this node as a single segment or to break it into several segments. If score of a parent node is larger than the score of its children it is kept as one of the selected nodes. We process all nodes in the post-order, namely, visit a node only if all its children have been visited. At last, the optimal cost of the root is the optimal tree-derived segmentation score. See Algorithm 2 for the psudocode.

Algorithm 2: Finding the Optimal Tree-Derived Segmentation Score

```
forall the $v \in \mathcal{V}$ do
    Compute $f(v)$;
Compute the post-order traversal of the tree, $\widehat{\mathcal{V}}$. A node is visited after its children;
forall the $v \in \widehat{\mathcal{V}}$ do
    if $v$ is a leaf node then
        $\hat{f}(v) = f(v)$;
        $S(v) = \{\{v\}\}$;
    else
        $\hat{f}(v) = \max\left(f(v), \sum_{u \in children(v)} \hat{f}(u)\right)$;
        if $f(v) > \sum_{u \in children(v)} \hat{f}(u)$ then
            $S(v) = \{\{v\}\}$;
        else
            $S(v) = \bigcup_{u \in children(v)} S(u)$;
return $S(root)$
```

6.6 Feature Types and Their Relation To Segmentation Tree

We now describe features for each node s_i of a hierarchical tree, given their corresponding image regions and the surfaces that separate adjacent nodes. We define two types of features: 1)*regional features* and 2) *boundary features*. The descriptions in this section is for 3D segmentation tasks. A 2D version can be defined similarly.

Regional Features:

For both image types (ssTEM 2D and FIBSEM 3D), we extract regional features using the

Table 6.1: Features.

No:	Regional Features
1	Volume of s_i.
2-33	32 bins histogram of classifier response values for the pixels of the region of s_i.
No:	Boundary Features
1-19	avg of profiles extracted from the gradient magnitude of the intensity.
20-38	std of profiles extracted from the gradient magnitude of the intensity.
39-57	avg of profiles extracted from the likelihood map.
58-76	std of profiles extracted from the likelihood map.

classifier output (i.e. likelihood map of being cell membrane) of the raw image. Note that such likelihood map is also used to generate watershed merging-trees. We use two different classifiers for two different data. For ssTEM images, we used the classifier provided by [108] as part of the ISBI 2011 Challenge. For FIBSEM image we used a random forest classifier provided in segmentation tool Ilastik [63]. These classifiers are trained using two class labels, cell membrane region vs non-cell membrane.

See Table 6.1) for the list of the components of the feature vector that we extracted for each node. The dimension of regional feature vector per node is 33.

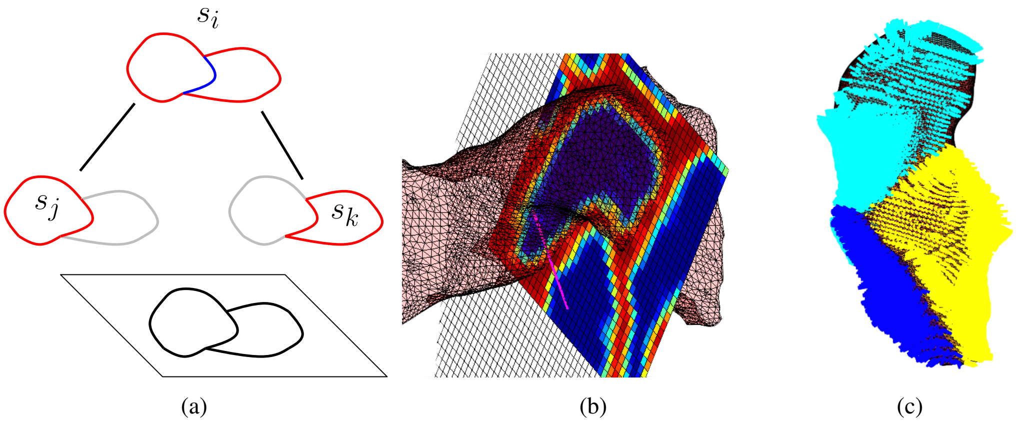


Figure 6.3: Boundary features. (a): external boundaries (red) and internal boundaries (blue), illustrated in 2D; (b) an example normal line segment at a boundary point (purple line), its profile is extracted using the likelihood map along the line (also shown); (c) normal lines of the whole external boundary, different colors correspond to boundary patches bewtween the node and different neighbors.

Boundary Features:

The boundary feature of a node depends on its *external boundary*, namely, the boundary of its region. The reason we call it external boundary will be obvious in Section 6.8. See

Fig. 6.3(a) for an illustration of external boundaries (red). Note that the external boundary of a node can be decomposed into boundary patches between different superpixels/supervoxels. [2] At every point of the boundary, we extract a normal line segment of a fixed length (19 pixels). Values of the likelihood map and intensity gradient along this line segment give a detailed description of this boundary point, called its *profile*. See Fig. 6.3(b) for an illustration. We extract profiles of every point on the boundary and use their statistics as the features. See Table 6.1 for more details.

6.7 Experiment & Results

In this section we validate our tree structured graphical model on two datasets. The first data set is the *Drosophila* first instar larva ventral nerve cord (VNC) and the second data set is *Drosophila melanogaster* larval neuropil. We apply our automated approach to both datasets and report quantitative evaluations comparing to the state-of-the-art methods. Holistic experiments demonstrate the necessity of structural constraints that are enforced in our graphical model.

In Section 6.9, we demonstrate our user interface for proofreading. We also show quantitative evidence that our new method improves the efficiency of manual correction.

Drosophila VNC Data Set: The *Drosophila* first instar larva ventral nerve cord (VNC) data set [92] contains 60 2D sections of ssTEM images, each of which has 512×512 pixels. The images were acquired using ssTEM at resolution of 4x4x50nm/pixel. This data set was used in ISBI 2012 EM Segmentation Challenge [109] with the ground truth 2D segmentation of 30 consecutive images as the training set and the other 30 consecutive images as the testing set. The ground truth is also publicly available for the training set.

For the 2D segmentation experiment, we trained our automatic algorithm with the 30 training images, tested on the 30 testing images and submitted our results to the ISBI 2012 EM Challenge server for evaluation. For tree extraction we used the boundary likelihood map from [108]. The graphical model training took 167 seconds and the MAP computation during testing per image was 0.05 seconds. Our method achieved the 2nd place in the competition out of the

[2] We used the open source software available at http://hci.iwr.uni-heidelberg.de/MIP/Software/cgp.php to extract boundaries.

Table 6.2: ISBI 2012 Competition Results

No.	Method	Rand Index Error	Warping Error	Pixel Error
1.	Human 1 vs Consensus	0.002	0.000005	0.001
2.	SCI [79]	0.018	0.000668	0.155
3.	Our approach	0.022	0.000807	0.110
4.	IDSIA [108]	0.050	0.000420	0.061
5.	Human 2 vs Consensus	0.029	0.000228	0.066
6.	Human 1 vs Human 2	0.030	0.000385	0.067

box in a single submission. We did not tune any parameter except the initial water level of the watershed transform. We found 0.02 is good water level choice to generate good enough over-segment regions. Qualitative examples are shown in Fig. 6.4.

In Table 6.7, we show our scores along with the results of two state-of-the-art methods and two human tracers. Note that our group name is ”optree-idsia” in the challenge website [109]. In Table 6.7, the first and the fifth entries are the errors between the human tracers and the consensus labels. The consensus labels are the ground truth (as for any regular submission). The sixth entry shows the errors between the two human tracers that segmented the test set independently. The labels from human 1 are considered ground truth and the labels from human 2 are the proposal. The second entry [79] in the table performs slightly better than our approach. This method uses adjacent slices when deciding the labels of each slice. In other words, they fix inconsistent labelings by introducing constraints in their formulation. However, this approach can only work for 2D stack images and cannot be used in a fully 3D approach. Note that, in ssTEM imaging the sectioning thickness is an important parameter and a limitation. Besides, our approach also has potential to be extended to take advantage of multiple image sections. The third entry [108] is the membrane detection method that we use as input for the tree extraction. Our approach improves the membrane detection results by over 2.8 percent.

To demonstrate the necessity of all structural constraints that we enforce in our algorithm, we ran a holistic experiment on the training set. We compared our method with three baseline approaches: i) using watershed transformation result at the best water level (WS); ii) using a classifier trained on node features for prediction of the node labels of the merging-tree (UC); iii) using a CRF based graphical model without the constraints defined in Theorem 6.3 (UNC).

Table 6.3: Baseline comparison.

	WS	NC	UNC	Optree	Optimal
Rand Index	0.05	0.14	0.10	**0.023**	0.015

In UC and UNC, we used post-processing to ensure the final result are legit segmentations (i.e. a pixel can get only one label). In the table on the right optree outperforms all baselines in the Adjusted Rand Index Error. We also present the optimal tree-derived segmentation result (Optimal), which is achieved when the groundtruth is known. This score is the theoretical upper-bound of our tree-derived segmentation.

Drosophila Neuropil Data Set: In this experiment we used a large 500x500x500 voxels isotropic volume generated by focused ion beam milling of Drosophila melanogaster larval neuropil, combined with scanning electron microscope imaging of the milled surface (FIBSEM) [3]. The microcube measures 2x2x1.5 microns approximately, with a resolution of 10 nm/pixel. Relative to the ssTEM data, FIBSEM has a smaller field of view, but yields isotropic resolution thus is compliant for a full 3D approach. We used the boundary likelihood map from “Ilastik” [63]. To reduce the watershed tree size, we removed all nodes with height less than 0.05. We divided the initial volume into 8 250x250x250 sub-volumes. We run experiments for 8 times. Each time we use one sub-volume for training and the rest for testing. Training took 465 seconds and MAP computation took 0.9 seconds on average. The average Rand Index score (one minus the Rand Index Error) is **0.9837**. Our method performs better than state-of-the-art [77, 75]. The optimal score of tree-derived segmentations (Optimal) is **0.9923**. There is still room for improvement over the optree segmentation result. In the next section, we show the result can be improved to the optimal via user interactions, as explained in Section 6.8.

6.8 Active Proofreading for Accurate Neuron Reconstruction

The domain of neuron segmentation requires extremely high segmentation quality. Therefore, proofreading, namely, interactively correcting mistakes of the automatic method is a necessary

[3] We would like to thank Harald Hess and C. Shan Xu at Janelia Farm Howard Hughes Medical Institute for providing the data.

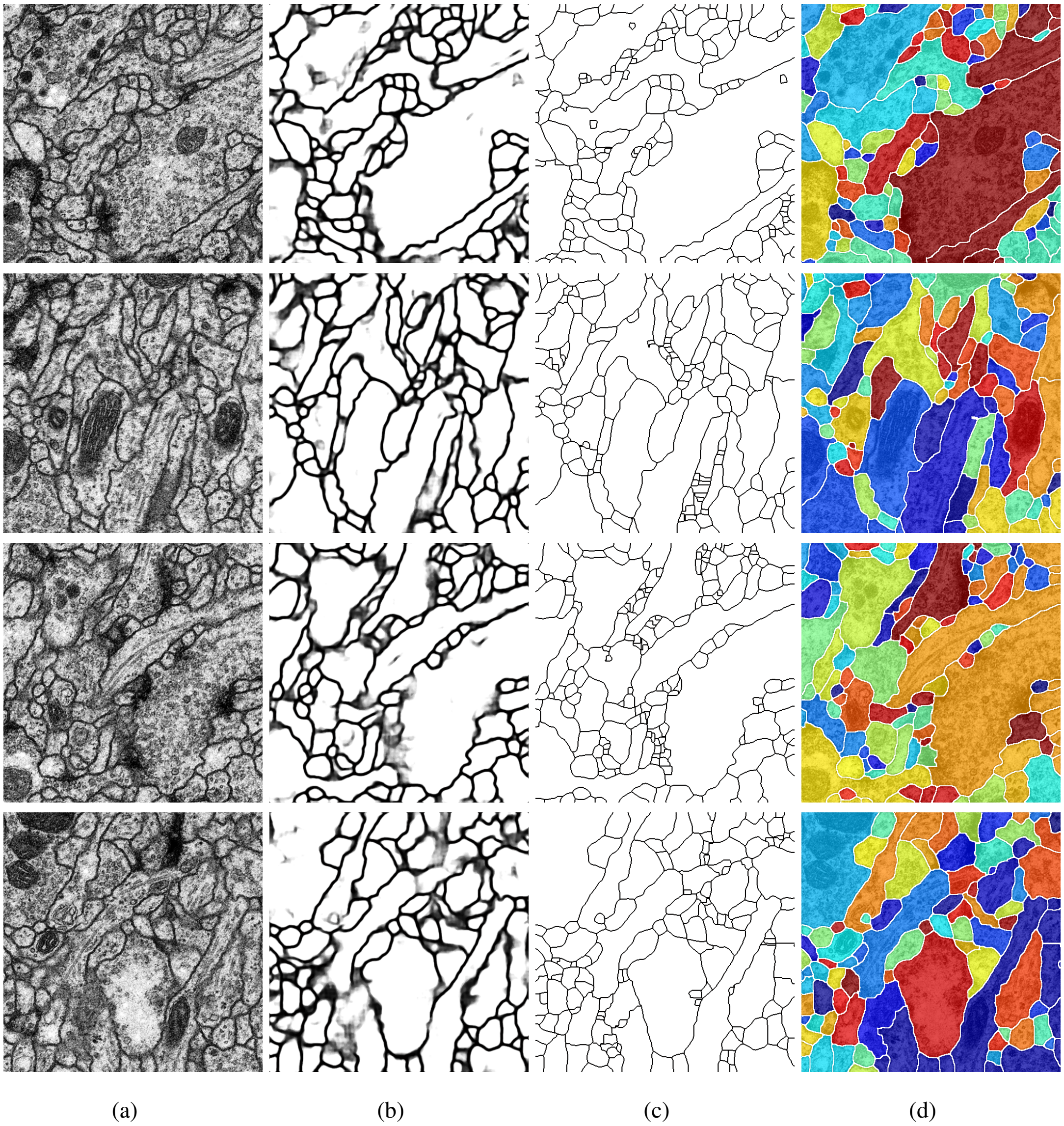


Figure 6.4: (a):Intensity image; (b) probability map obtained by a classifier; (c) initial superpixels obtained by applying watershed transform on probability map shown in (b); (d) our result

module in the pipeline. Due to the huge volume of the data, the proofreading step is always painfully inefficient. Based on our CRF model we explained in the previous Sections, we develop a novel interactive segmentation framework. Our new framework has two advantages.

1. Instead of the whole volume, we highlight only a few locations for a human expert to proofread. In particular, we highlight locations at which our model has the lowest confidence (cyan and yellow colored pixels in Fig. **??**). User input at these location will improve our model globally. Therefore, much fewer number of user input are required.

2. We develop an efficient algorithm to update our model after each user interaction. When a user fixes a mistake at one of these locations, our framework modifies the merging tree accordingly and recomputes the segmentation. The improvement to the segmentation is global. Together with an efficient tree-structured inference algorithm, our framework is fast enough for user interaction.

The uncertainty is measured by the marginals of the graphical model. With the hierarchy of the nodes in the tree structure, node marginals can be transformed to boundary marginals at no cost. The interactive system considers the boundary marginals to render the boundaries to be corrected by the user only if the confidence is low. Only giving a limited number of choices make the user interaction very efficient. Based on user corrections, our framework modifies the merging tree and thus improves the segmentation globally. Our experiments show that under the new framework, within fifteen user inputs, the segmentation is improved to the optimal quality, much faster than classical user interaction frameworks.

The workflow of our interactive segmentation algorithm is as follows. For a given test image, we compute an automatic segmentation using the algorithm presented in previous sections. Based on marginals of the graphical model, we suggest the user a few locations to proofread, in particular, the ones at which our model has the least confidence. When a user finds a mistake in the suggested locations, he/she clicks and corrects it. We modify the hierarchical tree accordingly and recompute the segmentation on the modified tree. The locations that have been corrected would not be highlighted anymore since the model is 100% sure about it. This process is repeated until the user is satisfied. Recall that to construct a graphical model, we need a parameter vector w. Throughout the user interaction, we use the same w which is learned in the training stage. See Figure 6.5 for the flowchart.

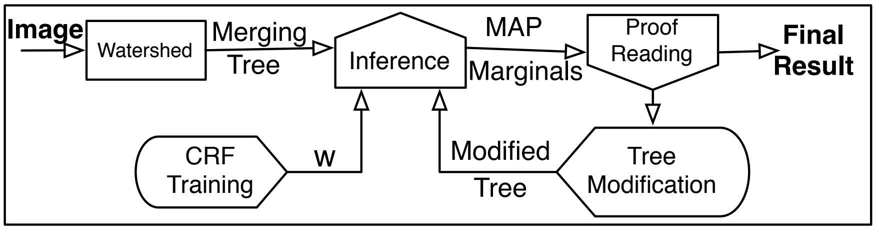


Figure 6.5: The flowchart of the interactive framework.

The advantage of our tree-structured interactive framework is two-folds. (1) It allows us to compute marginals efficiently. Based on marginals, we limit the attention of user on a small set of locations. This is very useful especially considering the huge size of the image. (2) After each user input, our algorithm improves the graphical model accordingly, and improve the segmentation globally. Depending on the correction scenario, fixing one location can result in multiple improvements at other locations due to the global optimization.

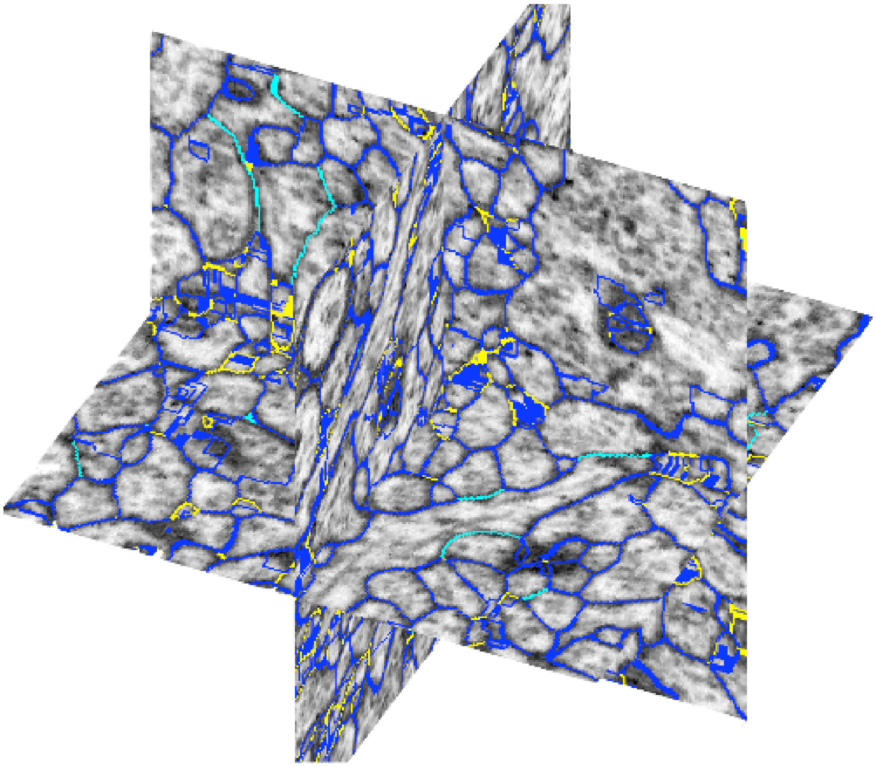


Figure 6.6: Boundaries a user have to proofread (blue) in a classical framework. Our system only highlights a small subset based on the marginals. Cyan boundaries are the candidates to be marked as split and yellow boundaries are the candidates to be markes as merge.

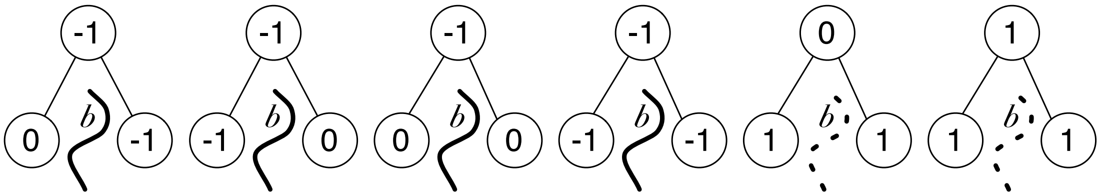


Figure 6.7: Transfer of node labels and marginals to internal boundary labeling and marginals. Dashed line represents a *merge* label while solid line is a *split*.

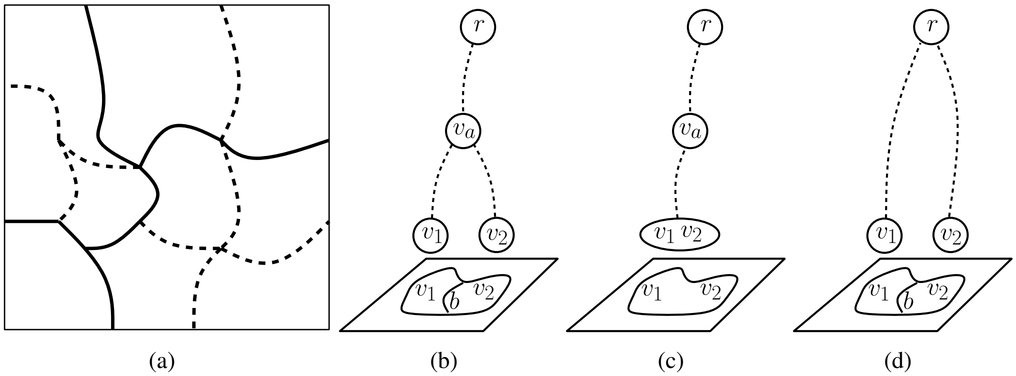


Figure 6.8: (a) in an example segmentation, there are 16 boundaries in the watershed transform, including 8 split (solid) and 8 merged (dashed); (b) a tree (nodes on the curved paths are not shown for clarity); (c) b is labeled merged; (d) b is labeled split.

The basic elements for a user to handle are the *elementary boundaries*, or in short *boundaries*, between superpixels. For a given segmentation, a boundary b is labeled *split* if the two adjacent superpixels belong to two different regions in the tree-derived segmentation. Otherwise, it is labeled *merged*. See Fig. 6.8(a) for an illustration. The task of segmentation is equivalent to finding an optimal split/merged labeling to all the boundaries. For a given node of the tree, we call the union of boundaries enclosing it as the *external boundary*, as we defined in Section 6.6. If it is not a leaf node, we call the intersection of the external boundaries of its two children its *internal boundary*. See Fig. 6.3(a) for an illustration.

Our CRF generates marginals for nodes of the tree. However, we can easily translate node marginals into marginals of whether each boundary being split or merged. For a given boundary b and its two adjacent superpixels/leaf nodes, v_1, v_2, we find the least common ancestor of v_1 and v_2, denoted by v_a. We call v_a the *containing node* of b, because b belongs to the interior boundary of v_a. (see Fig. 6.8(b)). The boundary b is split if and only if v_a is an undersegment, i.e. has label -1. Therefore, the probability of b being split is equal to the marginal

$P(y_{c_a} = -1|x, w)$. Figure 6.7 illustrates all possible labeling scenarios of the interior boundary of a given node.

At each iteration of the segmentation, we show the predicted segmentation and highlight the boundaries that we have low confidence of its corresponding label. This gives the user a small number of options to proofread. In Fig. 6.8, instead of all boundaries only the cyan boundaries (labeled merged, yet with low merged confidence) and the yellow boundaries (labeled split, yet with low split confidence). After the user corrects a boundary, the system will update the model accordingly and present the low marginal boundaries again.

We conclude this section by explaining how to update the tree structure according to user inputs. As we mentioned before there are two operations i) split and ii) merge that can be applied on a user selected boundary. Although these two operations are quite different, they mainly depend on enforcing new saliency values for the selected boundary. Recall the saliency values of boundaries decide when nodes are merged in the watershed algorith, and thus the structure of the merged tree. A naive algorithm is as follows. If the operation is merged (resp. split), we set its saliency value to zero (resp. infinitey). Afterwards, we recompute the merge tree. In the new tree, the specified boundary will have merged (resp. split) label for any possible tree-derived segmentation.

However, this operation is very expensive, and thus is not suitable for interative framework. In this section, we propose a new algorithm which only updates a small portion of the tree. This new algorithm is much more efficient than the native algorithm and thus can be used in practice.

Merge: When a user specifies a boundary to be merged, we merge the paths from the two adjacent leaf nodes to their lowest common ancestor node into a single path (see Fig.6.8(c)). Algorithm.3 presents how the subtree rooted by the lowest common ancestor of these two leaf nodes is restructured accordingly. We rebuild the subtree from bottom to top. During the process we use v_r to denote the root of the rebuilding tree. In the end, we replace the subtree rooted at v_a by the subtree rooted at v_r.

Split: When the user specifies a boundary to be split, we split the path from the least common ancestor of the split nodes (v_a) to the root into two paths. The nodes along the original path are assigned to either of the new paths, depending on their merging situation with other

Algorithm 3: Merge two nodes v_1 and v_2

```
$v_a \leftarrow$ the lowest common ancestor of $v_1$ and $v_2$;
$Q \leftarrow$ all nodes on the paths from $v_1$ and $v_2$ to $v_a$, except for $v_a$;
Sort $Q$ according to its height function (defined in Section 6.1);
$v_r \leftarrow \{v_1, v_2\}$;                ▷ $v_r$ is the root of the tree we are rebuilding;
$processed \leftarrow false$;                ▷ Bool array: whether a node has been processed;
$processed[v_1] \leftarrow true$;
$processed[v_2] \leftarrow true$;
for $v \in Q$ do
    $processed[v] = true$;
    $u \leftarrow$ the child of $v$ such that $processed[u] = false$;
    $w \leftarrow$ the child of $v$ such that $processed[w] = true$;
    $v \leftarrow \{v_r, u\}$                  ▷ $v$ is the parent of $v_r$ and $u$;
    $v_r \leftarrow v$                          ▷ update the root $v_r$;
Replace $v_a$ with $v_r$ in the original tree;
```

nodes. In Fig. 6.8(d), we illustrate this intuition with a toy example. Algorithm.4 presents the pseudocode. We build two new trees, rooted at v_{r1} and v_{r2} from bottom to top. In the end, these two roots merged into the new root. In order to ensure that these v_1 and v_2 are always split, we enforce an extra constraint that the new root could only have label -1.

Algorithm 4: Split two nodes v_1 and v_2

```
$v_a \leftarrow$ the lowest common ancestor of $v_1$ and $v_2$;
$Q \leftarrow$ the sequence of nodes on the path from $v_a$ to $root$;
$v_{r1} \leftarrow v_1$;                 ▷ the roots of the trees we are rebuilding;
$v_{r2} \leftarrow v_2$;                 ▷ the roots of the trees we are rebuilding;
for $v \in Q$ do
    $v_s \leftarrow$ the sibling of $v$;
    $f_1 \leftarrow$ the minimum of saliency values of all boundaries between $v_s$ and $v_{r1}$;
    $f_2 \leftarrow$ the minimum of saliency values of all boundaries between $v_s$ and $v_{r2}$;
    $i \leftarrow \mathrm{argmin}_j\, f_j$            ▷ the subtree to which $v_s$ should be merged to;
    $v_{ri} \leftarrow \{v_{ri}, v_s\}$;
$root \leftarrow \{v_{r1}, v_{r2}\}$;
```

Lastly, these operations can be applied sequencially and operation at one step does not hurt the results of previous ones. The graphical model keeps the history of the constraints, and ensures that at any iteration the predicted tree-derived segmentation satisfies all of the previous contsraints. Moreover, the improvement (or fixation) after a single operation is not limited to the processed nodes. At one operation, spliting or merging of two nodes affects other tree nodes in the vicinity and this improves the graphical model accordingly. Since the prediction is global among all tree nodes editing one location can fix many other locations in the image domain. We will show such cases in Section 6.9.

6.9 Experiment & Results

Our interactive system suggests a few boundaries for users to proofread at each iteration. Users judge by observation whether a boundary is mislabeled and correct it. We run experiments to show how this method could improve the efficiency of proofreading. We simulate a user interaction process on a particular 250^3 subvolume on which our automatic method has the worst score. We start from the automatic algorithm result and correct one mislabeled boundary at each iteration. At each iteration, the simulated user (robot) selects one mislabeled boundary based on certain strategy. The merging tree is modified accordingly. To illustrate the usefulness of marginals in the interaction, we implemented two interaction strategies (1) always select the mislabeled boundary with the highest marginal; (2) randomly select a mislabeled boundary.

In Figure 6.10(a), we compare the results of the two strategies. Correcting high-marginal boundaries (red curve) clearly improves the results much faster than correcting randomly selected boundaries (blue curve). The former takes about 14 iterations to reach the accuracy > 98.7 while the latter takes 39 iterations to reach the same accuracy. We also show the optimal result (black curve) of tree-derived segmentation as a theoretical upper bound. We also compare our interactive approach with method by [75] via using interaction strategy 1 where the mislabeled boundaries with highest marginal are selected. In this method the marginals are obtained by training a random forest classifier for predicting the *split* contour probability. The probabilities are used as the edge weights of a graphical model and they are never updated during the interaction process. Each iteration took around 6 7 seconds without warm start property. We used used same set of features for a fair comparison. As seen in the graph, our method reaches to its upper bound level with less iterations compared to [75] (green curve) using the two interaction strategies.

The advantage of our interactive segmentation framework is twofold. First, the interaction is very fast thanks to the efficient model updating algorithm and inference algorithms. In average, each iteration takes around 1.9 second.

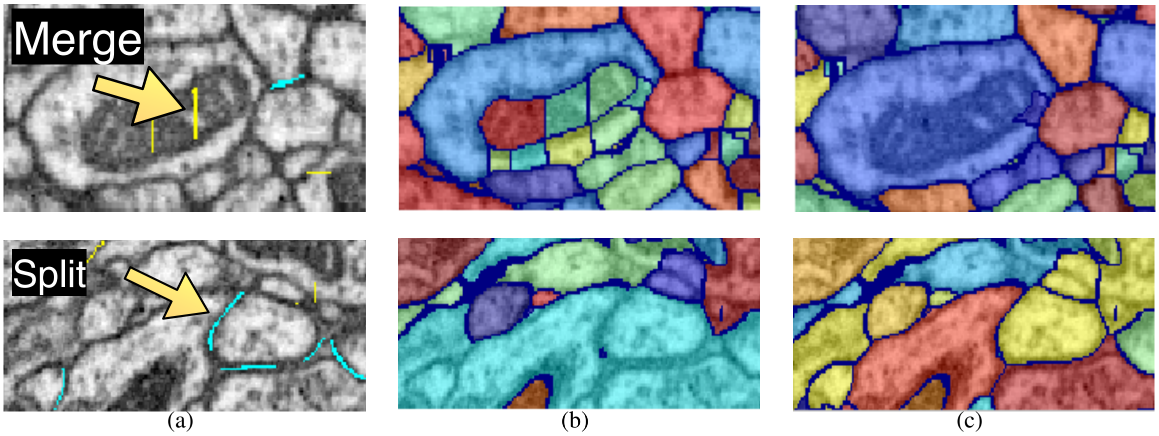
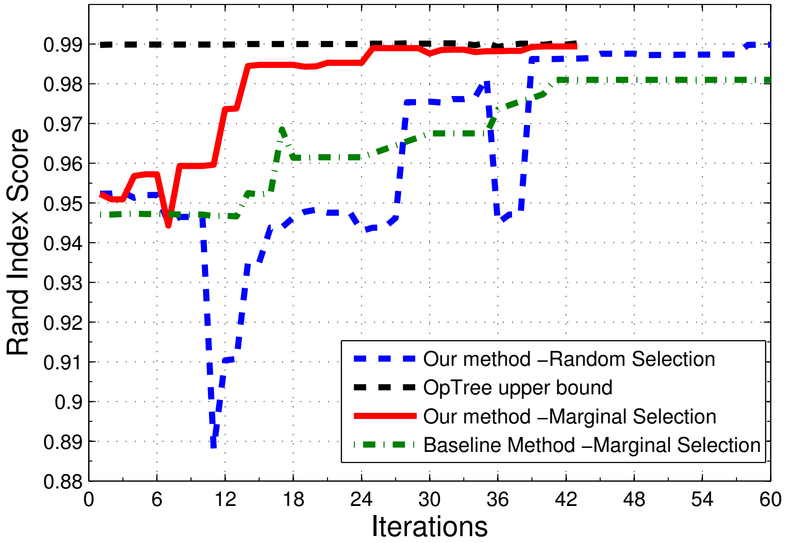


Figure 6.9: (a): our system only show boundaries with low marginals (yellow if labeled split, cyan if labeled merged); (b): before user input; (c): after user input, many boundaries are fixed.

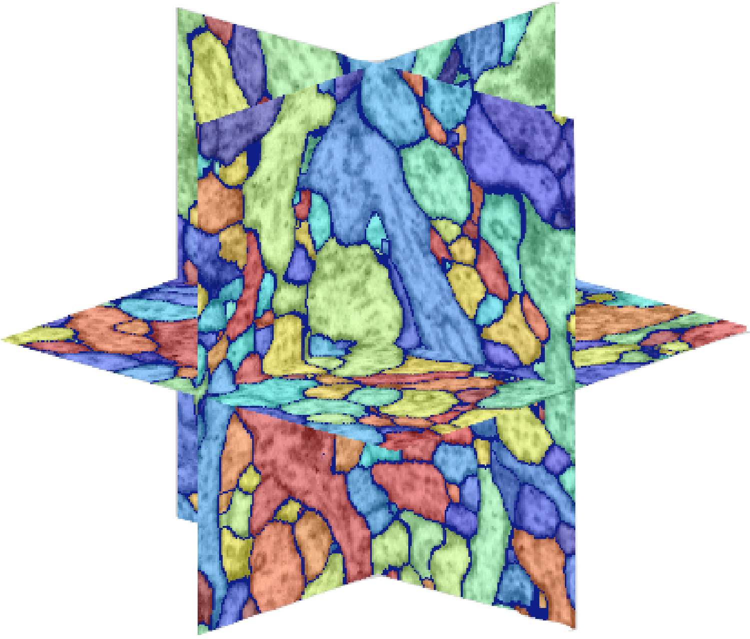
Second, marginals which are recomputed after every user input enables users to focus on the mislabeled contour segment with the most improvement of the score (See Figure 6.8). In Figure 6.9, we illustrate how one user input can improve the segmentation globally. After a selected boundary (yellow arrow) is corrected, many other boundaries are automatically corrected (Figure 6.9(a), 6.9(b) and 6.9(c)).

6.10 Conclusion

This paper presents a CRF-based algorithm for neuron segmentation. The tree-structured graphical model allows us to compute accurate segmentation of 500^3 dataset within a second. Furthermore, we develop an interactive segmentation framework that takes advantage of the marginals of the graphical model. The new framework improves the segmentation to the optimal quality within a small number of user inputs.



(a)



(b)

Figure 6.10: (a) Interaction Simulation; (b) 3D Result.

Chapter 7

Conclusions

In this dissertation, we proposed new methods for 3D organ segmentation from MR and CT images (macroscopic), 2D cellular region segmentation and 3D neuron structure extraction from EM images (microscopic). The first part of our contributions is about combining graphical models and deformable models into a collaborative framework for multi organ segmentation from CT and MR images. We presented a collaborative energy minimization framework where the deformable models and graphical models are combined together. In an iterative optimization scheme, two different models are improved in an alternating scheme towards more accurate segmentation. The overall framework takes advantage of these two segmentation paradigms. The final results are smooth, natural looking and can capture details while the optimization scheme is enhanced with globalization aspect of this hybrid framework. The proposed hybrid method is designed to satisfy the following requirements: 1) segmenting multiple objects simultaneously, 2) being robust to local high and low contrast challenges in cardiac CT images and knee MR images. While satisfying these requirements and having shown success in these applications, we also address the fact that decoupling the two model are decoupled in the minimization scheme. Current minimization scheme is based on alternating optimization (i.e. coordinate descent) and does not theoretically guarantees convergence. However, in practice we experienced that compared to conventional gradient descent based minimization scheme it performs better. As future work, we suggest using *supervoxel* representation instead of pixels for speeding up in 3D processing. Also, CRFs can be employed to learn the parameters of the graphical model when training data is available. Lastly, as in conventional Deformable Model approaches, use of shape prior in addition to graphical model based term could help improve the convergence rate of the minimization scheme.

The second part of our contributions is about designing a multi stage decision strategy for neuron segmentation from electron microscopy images. We propose a two step classification mechanism to identify cell membranes between adjacent cells. We solve this contextual characterization problem using a graph based local optimization scheme. Once the membranes are identified the cells can be separated well, despite the diffused and confusing membrane appearance. We showed promising results and the method is important in terms of scalability to very large Terabyte data; however, the general approach includes heuristics and depends on local optimization on a sparse graph. As we presented in Chapter 6 more global solutions are necessary for more robust segmentation results.

The third contribution of this dissertation is to create a tree structured CRF model to segment neuron structures from 2D and 3D electron microscopy images in a learning fashion. We use a region (*superpixel*) based approach where the relationships between the regions are encoded in a hierarchical tree. We employ efficient and exact belief propagation method to solve the inference problem. We show that the accuracy of the results are comparable to the state of the art. Last but not the least, we propose a method to extend the learning model into an active *proofreading* framework using the marginals of the graphical model as the confidence level of segmentation. With this feedback mechanism, we provide an interface where the user can reach globally improved accurate results with very few efforts compared to other baseline approaches.

References

[1] X. Huang, D. Metaxas, and T Chen. Metamorphs: Deformable shape and texture models. In *Computer Vision Pattern Recognition*, volume 1, pages 496–503, 2004.

[2] Rui Huang, V. Pavlovic, and D.N. Metaxas. A graphical model framework for coupling MRFs and deformable models. In *Computer Vision Pattern Recognition*, volume 2, pages II–739 – II–746 Vol.2, 2004.

[3] Chenyang Xu and Jerry L. Prince. Active contours, deformable models, and gradient vector flow, 2014. `http://www.iacl.ece.jhu.edu/static/gvf/`,[accessed: 11.15.2014].

[4] Sebastian Nowozin and Christoph H Lampert. *Structured learning and prediction in computer vision*, volume 6. Now publishers Inc, 2011.

[5] Level set representation, 2012. `http://www.engr.uconn.edu/~cmli/DRLSE/`,[accessed: 11.15.2014].

[6] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, August 2006.

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[8] Kevin L Briggman and Davi D Bock. Volume electron microscopy for neuronal circuit reconstruction. *Current Opinion in Neurobiology*, 22(1):154 – 161, 2012.

[9] Graham Knott, Herschel Marchman, David Wall, and Ben Lich. Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. *The Journal of Neuroscience*, 28(12):2959–2964, 2008.

[10] Mustafa Gökhan Uzunbaş, Chao Chen, Shaoting Zhang, Kilian M Poh, Kang Li, and Dimitris Metaxas. Collaborative multi organ segmentation by integrating deformable and graphical models. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 16(Pt 2):157–164, 2013.

[11] E Bas, M G Uzunbas, D Metaxas, and E Myers. *Contextual grouping in a concept: a multistage decision strategy for EM segmentation*. Proc. of ISBI, 2012.

[12] Mustafa Gökhan Uzunbaş, Chao Chen, and Dimitris Metaxas. Optree: a learning-based adaptive watershed algorithm for neuron segmentation. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 17(Pt 1):97–105, 2014.

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Int. J. Computer Vision*, volume 1, pages 321–331, 1988.

[14] Tian Shen, Hongsheng Li, and Xiaolei Huang. Active Volume Models for Medical Image Segmentation. *IEEE Trans. Med. Imaging ()*, 30(3):774–791, 2011.

[15] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic Active Contours. *International Conference on Computer Vision*, pages 694–699, 1995.

[16] A Yezzi Jr, S Kichenassamy, and A Kumar. A geometric snake model for segmentation of medical imagery. *Medical Imaging*, 1997.

[17] D Adalsteinsson and J A Sethian. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269–277, May 1995.

[18] Zheng Ying, Li Guangyao, Sun Xiehua, and Zhou Xinmin. Geometric active contours without re-initialization for image segmentation. *Pattern Recognition*, 42(9):1970–1976, September 2009.

[19] Chenyang Xu and Jerry L Prince. Gradient Vector Flow: A New External Force for Snakes. *CVPR*, pages 66–71, 1997.

[20] D MUMFORD and J SHAH. Optimal Approximations by Piecewise Smooth Functions and Associated Variational-Problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, July 1989.

[21] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on Image Processing ()*, 10(2):266–277, 2001.

[22] Luminita A. Vese, Tony F. Chan, Tony, and F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50:271–293, 2002.

[23] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, pages 1065–1076, 1962.

[24] Timo Kohlberger, M. Gökhan Uzunbaş, Christopher Alvino, Timor Kadir, Daniel O. Slosman, and Gareth Funka-Lea. Organ segmentation with level sets using local shape and appearance priors. Medical Image Computing and Computer-Assisted Intervention '09, pages 34–42. Springer-Verlag, 2009.

[25] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active Appearance Models. *IEEE Trans. Pattern Anal. Mach. Intell. ()*, 23(6):681–685, 2001.

[26] Andy Tsai, Anthony Yezzi, William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W Eric Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE transactions on medical imaging*, 22(2):137–154, February 2003.

[27] S Dambreville and Y Rathi. A framework for image segmentation using shape models and kernel space shape priors. *Pattern Analysis and . . .* , 2008.

[28] Joshua B Tenenbaum. Mapping a Manifold of Perceptual Observations. *NIPS 1997*, 1997.

[29] Ahmed M Elgammal and Chan-Su Lee. Separating Style and Content on a Nonlinear Manifold. *CVPR*, pages 478–485, 2004.

[30] Y Freund and R E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. *Computational learning theory*, 1995.

[31] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research ()*, 9:1871–1874, 2008.

[32] G I CHIOU and J N HWANG. A Neural-Network-Based Stochastic Active Contour Model (Nns-Snake) for Contour Finding of Distinct Features. *Ieee Transactions on Image Processing*, 4(10):1407–1416, October 1995.

[33] Yongmei Wang and Lawrence H Staib. Boundary Finding with Correspondence Using Statistical Shape Models. *CVPR*, pages 338–345, 1998.

[34] M Rousson and N Paragios. Shape priors for level set representations. *European Conference on Computer Vision—ECCV 2002*, 2002.

[35] J Kim, M Çetin, and A S Willsky. Nonparametric shape priors for active contour-based image segmentation. *Signal Processing*, 2007.

[36] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Comparing Active Shape Models with Active Appearance Models. *BMVC*, pages 1–10, 1999.

[37] Mustafa Gokhan Uzunbas, Octavian Soldea, Devrim Unay, Mujdat Cetin, Gozde Unal, Aytul Ercil, and Ahmet Ekin. Coupled Nonparametric Shape and Moment-Based Intershape Pose Priors for Multiple Basal Ganglia Structure Segmentation. *Ieee Transactions on Medical Imaging*, 29(12):1959–1978, December 2010.

[38] Daphne Koller and Nir Friedman. Probabilistic Graphical Models - Principles and Techniques. *MIT Press 2009*, 2009.

[39] Matthias Gallé. Review of bayesian reasoning and machine learning by David Barber. *SIGACT News ()*, 45(2):27–29, 2014.

[40] S Geman and D Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE transactions on pattern analysis and machine intelligence*, 6(6):721–741, June 1984.

[41] R Kindermann and J L Snell. Markov random fields and their applications, 1980.

[42] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal Computer Vision*, 70(2):109–131, 2006.

[43] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[44] Frank R Kschischang, Brendan J Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory ()*, 47(2):498–519, 2001.

[45] D M GREIG, B T PORTEOUS, and A H SEHEULT. Exact Maximum a-Posteriori Estimation for Binary Images. *Journal of the Royal Statistical Society Series B-Methodological*, 51(2):271–279, 1989.

[46] Rodney J. Baxter. *Exactly solved models in statistical mechanics*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], London, 1982.

[47] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[48] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Trans. on Medical Imaging*, 20(1):45–57, 2001.

[49] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147 –159, feb. 2004.

[50] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *Int. J. Computer Vision*, 22(1):61–79, 1997.

[51] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

[52] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. of Computer Vision*, 50:271–293, 2001.

[53] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[54] Xinjian Chen, Jayaram K. Udupa, Ulas Bagci, Ying Zhuge, and Jianhua Yao. Medical image segmentation by combining graph cuts and oriented active appearance models. *IEEE Trans. Image Processing*, 21(4):2035–2046, 2012.

[55] Laurent D. Cohen and Isaac Cohen. Finite element methods for active contour models and balloons for 2d and 3d images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:1131–1147, 1991.

[56] Jeff W Lichtman and Joshua R Sanes. Ome sweet ome: what can the genome tell us about the connectome? *Current Opinion in Neurobiology*, 18(3):346–353, June 2008.

[57] Olaf Sporns, Giulio Tononi, and Rolf Ktter. The human connectome: A structural description of the human brain. *PLoS Comput Biol*, 1(4):e42, 09 2005.

[58] Moritz Helmstaedter and Partha P Mitra. Computational methods and challenges for large-scale circuit mapping. *Current Opinion in Neurobiology*, 22(1):162 – 169, 2012.

[59] V. Jain, B. Bollmann, M. Richardson, D.R. Berger, M.N. Helmstaedter, K.L. Briggman, W. Denk, J.B. Bowden, J.M. Mendenhall, W.C. Abraham, K.M. Harris, N. Kasthuri, K.J. Hayworth, R. Schalek, J.C. Tapia, J.W. Lichtman, and H.S. Seung. Boundary learning by optimization with topological constraints. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2488–2495, June 2010.

[60] Winfried Denk Moritz Helmstaedter, Kevin L Briggman. High-accuracy neurite reconstruction for high-throughput neuroanatomy, 2011.

[61] Viren Jain, H Sebastian Seung, and Srinivas C Turaga. Machines that learn to segment images: a crucial technology for connectomics. *Current Opinion in Neurobiology*, 20(5):653 – 666, 2010.

[62] V Jain, S C Turaga, K L Briggman, and M Helmstaedter. Learning to Agglomerate Superpixel Hierarchies. *NIPS*, 2011.

[63] Christoph Sommer, Christoph Straehle, Ullrich Koethe, and Fred A. Hamprecht. ilastik: Interactive learning and segmentation toolkit. In *8th IEEE Int. Symposium on Biomedical Imaging*, 2011.

[64] Dmitri B Chklovskii, Shiv Vitaladevuni, and Louis K Scheffer. Semi-automated reconstruction of neural circuits using electron microscopy. *Current opinion in neurobiology*, 20(5):667–675, 2010.

[65] C Straehle, U Koethe, G Knott, K Briggman, W Denk, and F A Hamprecht. Seeded watershed cut uncertainty estimators for guided interactive segmentation. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 765–772. IEEE, 2012.

[66] V Kaynig, Amelio Vazquez-Reina, S Knowles-Barley, Mike Roberts, Thouis R Jones, Narayanan Kasthuri, Eric Miller, Jeff Lichtman, and Hanspeter Pfister. Large-Scale Automatic Reconstruction of Neuronal Processes from Electron Microscopy Images. *Arxiv*, March 2013.

[67] Ting Liu, Cory Jones, Mojtaba Seyedhosseini, and Tolga Tasdizen. A modular hierarchical approach to 3D electron microscopy image segmentation. *Journal of neuroscience methods*, 226:88–102, April 2014.

[68] V Jain, J Murray, F Roth, S Turaga, V Zhigulin, K Briggman, M Helmstaedter, W Denk, and H Seung. Supervised learning of image restoration with convolutional networks. *International Conference on Computer Vision*, pages 1–8, 2007.

[69] Elizabeth Jurrus, Antonio R C Paiva, Shigeki Watanabe, James R Anderson, Bryan W Jones, Ross T Whitaker, Erik M Jorgensen, Robert E Marc, and Tolga Tasdizen. Detection of neuron membranes in electron microscopy images using a serial neural network architecture. *Medical Image Analysis*, 14(6):770–83, 2010.

[70] Elizabeth Jurrus, Shigeki Watanabe, RichardJ. Giuly, AntonioR.C. Paiva, MarkH. Ellisman, ErikM. Jorgensen, and Tolga Tasdizen. Semi-automated neuron boundary detection and nonbranching process segmentation in electron microscopy images. *Neuroinformatics*, 11(1):5–29, 2013.

[71] B Andres, U Köthe, M Helmstaedter, W Denk, and F Hamprecht. Segmentation of sbfsem volume data of neural tissue by hierarchical classification. *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pages 142–152, 2008.

[72] Srinivas C. Turaga, Kevin L. Briggman, Moritz Helmstaedter, Winfried Denk, and H. Sebastian Seung. Maximin affinity learning of image segmentation. *CoRR*, 2009.

[73] Srinivas C Turaga, Joseph F Murray, Viren Jain, Fabian Roth, Moritz Helmstaedter, Kevin Briggman, Winfried Denk, and H Sebastian Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Comput*, 22(2):511–38, 2010.

[74] Amelio Vazquez-Reina, Michael Gelbart, Daniel Huang, Jeff Lichtman, Eric Miller, and Hanspeter Pfister. Segmentation fusion for connectomics. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 177–184. IEEE Computer Society, 2011.

[75] Bjoern Andres, Thorben Kroeger, Kevin L Briggman, Winfried Denk, Natalya Korogod, Graham Knott, Ullrich Koethe, and Fred A Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *European Conference on Computer Vision, 2012*, pages 778–791. Springer, 2012.

[76] Bjoern Andres, Ullrich Koethe, Thorben Kroeger, Moritz Helmstaedter, Kevin L Briggman, Winfried Denk, and Fred A Hamprecht. 3D segmentation of SBFSEM images of neuropil by a graphical model over supervoxel boundaries. *Medical Image Analysis*, 16(4):796–805, 2012.

[77] Thorben Kroeger, Shawn Mikula, Winfried Denk, Ullrich Koethe, and FredA. Hamprecht. Learning to segment neurons with non-local quality measures. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2013*, pages 419–427. Springer, 2013.

[78] Bjrn Andres, Ullrich Kthe, Moritz Helmstaedter, Winfried Denk, and FredA. Hamprecht. Segmentation of sbfsem volume data of neural tissue by hierarchical classification. volume 5096 of *Lecture Notes in Computer Science*, pages 142–152. Springer Berlin Heidelberg, 2008.

[79] Ting Liu, Cory Jones, Mojtaba Seyedhosseini, and Tolga Tasdizen. A modular hierarchical approach to 3d electron microscopy image segmentation. *Journal of Neuroscience Methods*, 226(0):88 – 102, 2014.

[80] Juan Nunez-Iglesias, Ryan Kennedy, Toufiq Parag, Jianbo Shi, and Dmitri B Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PLOS ONE*, 8(8):e71715, 2013.

[81] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. *International Conference on Computer Vision 2005*, 1:564 – 571 Vol. 1, 2005.

[82] Nhat Vu and B.S Manjunath. Graph cut segmentation of neuronal structures from transmission electron micrographs. *ICIP*, pages 725–728, 2008.

[83] V Kaynig, T Fuchs, and J.M Buhmann. Neuron geometry extraction by perceptual grouping in sstem images. *Computer Vision Pattern Recognition*, pages 2902–2909, 2010.

[84] Hanchuan Peng Hanchuan Peng, Fuhui Long Fuhui Long, and C Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[85] Erhan Bas and Deniz Erdogmus. Principal curves as skeletons of tubular objects - locally characterizing the structures of axons. *Neuroinformatics*, 9(2-3):181–191, 2011.

[86] Umut Ozertem and Deniz Erdogmus. Locally defined principal curves and surfaces. *J. Mach. Learn. Res.*, 12:1249–1286, 2011.

[87] Xiao Chen He and Nelson H C Yung. Corner detector based on global and local curvature properties. *Optical Engineering*, 47(5), 2008.

[88] Albert Cardona, Stephan Saalfeld, Stephan Preibisch, Benjamin Schmid, Anchi Cheng, Jim Pulokas, Pavel Tomancak, and Volker Hartenstein. An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLoS Biol*, 8, 2010.

[89] V Jain, B Bollmann, M Richardson, D.R Berger, M.N Helmstaedter, K.L Briggman, W Denk, J.B Bowden, J.M Mendenhall, W.C Abraham, K.M Harris, N Kasthuri, K.J Hayworth, R Schalek, J.C Tapia, J.W Lichtman, and H.S Seung. Boundary learning by optimization with topological constraints. *Computer Vision Pattern Recognition*, pages 2488–2495, 2010.

[90] Serge Beucher and Fernand Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical Morphology in Image Processing*, 1992.

[91] Laurent Najman and Michel Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *Pattern Analysis and Machine Intelligence*, 18(12):1163–1173, 1996.

[92] Albert Cardona, Stephan Saalfeld, Stephan Preibisch, Benjamin Schmid, Anchi Cheng, Jim Pulokas, Pavel Tomancak, and Volker Hartenstein. An integrated micro-and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLoS biology*, 8(10), 2010.

[93] Sebastian Nowozin, Peter V Gehler, and Christoph H Lampert. On parameter learning in CRF-based approaches to object class image segmentation. In *European Conference on Computer Vision*, pages 98–111. Springer, 2010.

[94] Victor S Lempitsky, Andrea Vedaldi, and Andrew Zisserman. Pylon model for semantic segmentation. In *NIPS*, volume 24, pages 1485–1493, 2011.

[95] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition IEEE Conference on*, pages 2141–2148. IEEE, 2010.

[96] B Ravi Kiran and Jean Serra. Global–local optimizations by hierarchical cuts and climbing energies. *Pattern Recognition*, 47(1):12–24, 2014.

[97] Srinivas C. Turaga, Kevin L. Briggman, Moritz Helmstaedter, Winfried Denk, and H. Sebastian Seung. Maximin affinity learning of image segmentation. In *NIPS*, pages 1865–1873, 2009.

[98] Jeff W. Lichtman and Winfried Denk. The big and the small: Challenges of imaging the brains circuits. *Science*, 334(6056):618–623, 2011.

[99] D Haehn, S Knowles-Barley, M Roberts, J Beyer, N Kasthuri, J W Lichtman, and H Pfister. Design and Evaluation of Interactive Proofreading Tools for Connectomics. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2466–2475, December 2014.

[100] Seymour Knowles-Barley, Mike Roberts, Narayanan Kasthuri, Dongil Lee, Hanspeter Pfister, and Jeff W Lichtman. Mojo 2.0: Connectome annotation tool. *Frontiers in Neuroinformatics*, (60).

[101] Janelia Farm. Raveler, 2014. `https://openwiki.janelia.org/wiki/display/flyem/Raveler`,[accessed: 11.15.2014].

[102] Jinseop S Kim, Matthew J Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F Behabadi, Michael Campos, Winfried Denk, H Sebastian Seung, and Eye-Wirers. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–+, 2014.

[103] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1568–1583, 2006.

[104] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf optimization via dual decomposition: Message-passing revisited. In *Computer Vision, ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[105] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf energy minimization and beyond via dual decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):531–552, 2011.

[106] Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, volume 2, pages 939–946. IEEE, 2005.

[107] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[108] Dan Ciresan, Alessandro Giusti, Juergen Schmidhuber, et al. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems 25*, pages 2852–2860, 2012.

[109] Ignacio Arganda-Carreras, Sebastian Seung, Albert Cardona, and J. Schindelin. Segmentation of neuronal structures in em stacks challenge, 2012. `http://brainiac2.mit.edu/isbi_challenge/leaders-board`,[accessed: 04.15.2014].