UNIVERSITY OF CALIFORNIA
Santa Barbara

Image Segmentation with Semantic Priors: A Graph Cut Approach

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Nhat Bao Sinh Vu

Committee in Charge:

Professor B. S. Manjunath, Chair

Professor Steven Fisher

Professor Michael Liebling

Professor Kenneth Rose

Professor Ambuj Singh

September 2008

The Dissertation of
Nhat Bao Sinh Vu is approved:

Professor Steven Fisher

Professor Michael Liebling

Professor Kenneth Rose

Professor Ambuj Singh

Professor B. S. Manjunath, Committee Chairperson

August 2008

Image Segmentation with Semantic Priors:

A Graph Cut Approach

Copyright © 2008

by

Nhat Bao Sinh Vu

To my parents and teachers

Acknowledgments

Although this thesis encompasses several years of research, its culmination is due in large part to many more years of support and encouragement from a very special group of people, and it gives me great pleasure to acknowledge them here. First, I would like to thank Professor Manjunath, my advisor and mentor, for his unwavering confidence in my abilities and research decisions. He welcomed me into the Vision Research Lab (VRL), supported me financially with NSF IGERT grant #DGE-0221713 and NSF ITR grant #ITR-0331697, and provided me with an interdisciplinary environment in which to explore a wide range of topics from media arts to retinal biology.

Second, I would like to thank my colleagues in the VRL for their friendships and for the many stimulating discussions and research collaborations that we have shared. They were never hesitant in offering advice and assistance, whether it be on selecting the fastest compiler or in choosing the most scenic hikes. In addition, my UCSB experience has been incredibly enriching thanks to my friends and apartmentmates Luke and Phu, my tri-tip grilling sidekick Mike, and my many intramural sport teammates, especially my basketball co-star Elisa.

Next, I would like to thank my teachers, who have shown great patience and enthusiasm in educating me from the time I started the fourth grade, when I barely spoke a word of English, to my senior year of high school, when I scarcely kept my mouth silent. In particular, I would like to thank Mrs. Menard for pointing me in the right

direction by introducing to me the concept of academic research. Additionally, I would like to thank the many professors both at WashU and UCSB who have made my search for knowledge and understanding both easy and enjoyable.

Most importantly, I would like to thank my biggest supporters—my family. My sister Thu has been my academic trailblazer, and her accomplishments are great examples from which I can emulate. My parents, to whom this thesis is dedicated, have always encouraged me to pursue a higher education. Their lifelong sacrifices have afforded me the opportunities to attend college and graduate school and continue to be a major source of motivation for everything I do in life. My best friend Jen has been my truest companion ever since our first year of college when we embarked on this journey together. Her passion for science can only be rivaled by her love of life, and it is this contagious combination that has motivated me to persevere through this thesis work.

Nhat Vu
August 2008
Santa Barbara, CA

Curriculum Vitæ

Nhat Bao Sinh Vu

Education

September 2008	Doctor of Philosophy in Electrical and Computer Engineering, University of California, Santa Barbara.
June 2005	Master of Science in Electrical and Computer Engineering, University of California, Santa Barbara.
May 2003	Bachelor of Science in Electrical Engineering, Washington University in St. Louis, Missouri.

Participation

2003 – 2008	Trainee in the NSF Interactive Digital Multimedia IGERT Program, University of California, Santa Barbara.
2005 – 2008	Graduate Student Researcher in the NSF ITR Center for Bio-image Informatics, University of California, Santa Barbara.

Selected Publications

Nhat Vu and B. S. Manjunath, "Graph Cut Segmentation of Neuronal Structures from Transmission Electron Micrographs." In *Proc. IEEE International Conference on Image Processing*, October 2008.

Nhat Vu and B. S. Manjunath, "Shape Prior Segmentation of Multiple Objects with Graph Cuts." In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

Nhat Vu, Pratim Ghosh, and B. S. Manjunath, "Retina Layer Segmentation and Spatial Alginment of Antibody Expression Levels." In *Proc. IEEE International Conference on Image Processing*, September 2007.

Jiyun Byun, Nhat Vu, Baris Sumengen, and B. S. Manjunath, "Quantitative Analysis of Immunofluorescent Retinal Images." In *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2006.

Abstract

Image Segmentation with Semantic Priors: A Graph Cut Approach

Nhat Bao Sinh Vu

Image segmentation is the partitioning of an image into meaningful regions or pixel groups and is a necessary prerequisite for many higher level computer vision tasks, such as object recognition, scene interpretation, and content-based image retrieval. However, the segmentation problem is inherently ill-posed due to the large number of possible partitionings for any single image. Much effort in image segmentation research is devoted to making the problem more tractable by constraining the solution space using prior information. Commonly, the optimality criteria used to compute a preferred partitioning are formulated base on measures that account for contour smoothness, regional coherence, and visual homogeneity.

In this thesis, we present a set of novel image segmentation algorithms that utilize high-level semantic priors available from specific application domains. These priors are incorporated into the segmentation framework to further constrain the results to a more semantically meaningful solution space. Our algorithms are formulated using Random Field models and employ combinatorial graph cuts for efficient optimization. For many instances, they guarantee the globally optimal solutions, and our experiments demonstrate that the algorithms are applicable to a wide range of segmentation tasks.

Contents

Acknowledgments v

Curriculum Vitæ vii

Abstract viii

List of Figures xii

1 Introduction 1
- 1.1 Segmentation with Prior Information . . . 3
- 1.2 Semantic Priors . . . 4
  - 1.2.1 Shape Priors . . . 5
  - 1.2.2 Multiview Priors . . . 6
  - 1.2.3 Region Topology Priors . . . 8
  - 1.2.4 Presegmentation Priors . . . 10
- 1.3 Why Combinatorial Graph Cuts? . . . 12
- 1.4 Summary of Contributions . . . 13
- 1.5 Organization of Thesis . . . 16

2 Random Fields and Graph Cuts in Image Segmentation 18
- 2.1 Image Segmentation Overview . . . 19
- 2.2 MAP Estimation for Discrete Models . . . 23
  - 2.2.1 Random Fields . . . 23
  - 2.2.2 Markov Random Fields . . . 24
  - 2.2.3 Conditional Random Fields . . . 27
  - 2.2.4 CRFs for Image Segmentation . . . 29
- 2.3 Energy Minimization . . . 31
  - 2.3.1 Submodular Functions . . . 31

2.3.2 Energy Minimization Algorithms . . . 33
2.4 Graph Cuts . . . 37
2.4.1 Graph Structure for Image Segmentation . . . 38
2.4.2 Finding the Mincut by Maximum Flow . . . 45

3 Segmentation of Serial Electron Micrographs 47
3.1 Segmentation of Neuronal Structures from Transmission EM . . . 49
3.2 Segmentation of 2D EM Images . . . 53
3.2.1 Related Works on 2D EM Segmentation . . . 53
3.2.2 Interactive 2D Segmenation . . . 54
3.2.3 Pairwise penalty using image intensity . . . 56
3.2.4 Results with Intensity . . . 58
3.2.5 Unary penalty using flux . . . 63
3.3 Segmentation of Serial EM Stack . . . 68
3.3.1 Related Works . . . 69
3.3.2 3D Neighborhood Graph . . . 71
3.3.3 Label Propagation . . . 73
3.3.4 Results . . . 76
3.4 Conclusion . . . 80

4 Segmentation Using Shape Prior 83
4.1 Related Works . . . 85
4.2 Segmentation Energy . . . 87
4.3 Shape prior model . . . 89
4.3.1 Shape distance . . . 89
4.3.2 Shape penalty . . . 91
4.3.3 Affine invariant shape alignment . . . 92
4.4 Shape prior segmentation . . . 93
4.4.1 Multiphase graph cuts . . . 94
4.4.2 Iterative Segmentation with Shape Prior . . . 98
4.4.3 Data model . . . 102
4.5 Multiple prior shapes . . . 103
4.6 Experiments . . . 104
4.7 Conclusion . . . 114

5 Globally Optimal Nested Layer Segmentation 116
5.1 Nested Region Topology . . . 118
5.1.1 Label Adjacency Constraint . . . 120
5.2 Related Works . . . 121
5.3 Graph Representation of $\mathcal{M}_s^2$ Functions . . . 124

5.3.1 Boolean Transformation $\mathcal{M}_s^2 \rightarrow \mathcal{F}_s^2$ . . . 125
5.3.2 Encoding Unary Multi-label Variables . . . 126
5.3.3 Encoding Pairwise Multi-label Variables . . . 129
5.4 Minimizing $\mathcal{M}_s^2$ with Label Constraint . . . 132
5.4.1 Boolean Encoding with Adjacency Constraint . . . 132
5.4.2 Submodularity of $\mathcal{M}_s^2$ with Label Constraint . . . 134
5.4.3 Label Adjacency Constraint Graph . . . 135
5.5 Experiments . . . 139
5.5.1 Clique Potentials . . . 140
5.5.2 Image Features . . . 142
5.5.3 Segmentation Workflow . . . 143
5.5.4 Label Adjacency Validation . . . 145
5.5.5 Comparison with Other Algorithms . . . 146
5.5.6 Results . . . 150
5.6 Conclusion . . . 157

6 Towards Bioimage Analysis **159**
6.1 Interactive Editing . . . 160
6.1.1 Edit Energy . . . 162
6.1.2 Two-Label Case . . . 165
6.1.3 Multi-Label Case . . . 168
6.2 Analysis of the Outer Nuclear Layer . . . 172
6.2.1 Local Thickness and Density Measurements . . . 175
6.2.2 Analysis Results . . . 177
6.3 Spatial Analysis of Antibody Expression Levels . . . 180
6.3.1 Expression Level Correspondence . . . 182
6.3.2 Preliminary Biological Analysis . . . 184
6.4 Conclusion . . . 186

7 Conclusion and Future Outlook **187**
7.1 Future Directions . . . 188
7.1.1 Hierarchical Layer Segmentation . . . 188
7.1.2 Higher Order $\mathcal{P}^n$ Potts Potential . . . 189
7.1.3 Layer Segmentation with Probabilistic Nesting . . . 191
7.2 Conclusion . . . 192

Bibliography **193**

List of Figures

1.1 Segmentation of a walking sequence using only intensity. 5
1.2 Example of serial stack prior. 7
1.3 Nested region topology examples. 9
1.4 Electron micrograph. 11

2.1 Graph construction to optimize submodular binary second order energy. 40
2.2 Simplified graph for optimization of binary second order energy. . . . 41
2.3 Graph construction for optimizing the $\mathcal{P}^n$ Potts clique potential. . . . 44

3.1 Adjacent slices from serial EM. 50
3.2 Mutual prior images. 51
3.3 Example 1: Sensitivity to σ_x. 59
3.4 Example 2: Sensitivity to σ_x. 60
3.5 Example 1: Input marking sensitivity. 62
3.6 Example 2: Input marking sensitivity. 63
3.7 Gradient field flux. 64
3.8 Modulated unary flux costs. 66
3.9 Results using flux. 67
3.10 Segmentation using 3D neighborhood graph. 72
3.11 Higher order clique maps. 77
3.12 Example of the clique costs. 78
3.13 Label propagation results. 79
3.14 Segmentation result for serial stack. 80

4.1 Shape representations. 90
4.2 Regions for two labelings $\mathbf{y}^1$ and $\mathbf{y}^2$. 95
4.3 Multiphase graph cuts example. 98
4.4 Segmentation of a hand with shape prior 101

4.5 Shape templates. . . . 105
4.6 Robust initializations . . . 106
4.7 Segmentation of occluded leaf with increasing noise levels. . . . 107
4.8 Segmentation of a guitar body . . . 109
4.9 Silhouette templates used for multishape prior segmentation. . . . 110
4.10 Segmentation of a walking sequence. . . . 111
4.11 Segmentation of two leaves along with estimated images. . . . 112
4.12 Another example of two object segmentation. . . . 112
4.13 Segmentation of three objects. . . . 113

5.1 Triple junction example. . . . 117
5.2 Examples of layer nesting. . . . 118
5.3 Unary variable encoding graph and infeasible cut. . . . 127
5.4 Pairwise variable encoding graphs. . . . 131
5.5 Edges enforcing label adjacency constraint. . . . 136
5.6 Nested layer segmentation graph. . . . 137
5.7 Nested layer segmentation workflow. . . . 144
5.8 Label adjacency validation. . . . 145
5.9 Comparison with Ishikawa method. . . . 147
5.10 Maximum flow and energy plot for Ishikawa method. . . . 148
5.11 Comparison with the $\alpha\beta$-swap algorithm. . . . 150
5.12 Segmentation of retinal cross section 1. . . . 151
5.13 Segmentation of a longitudinal section of the jejunum. . . . 152
5.14 Segmentation of a cross-section of the jejunum. . . . 153
5.15 Segmentation of immunofluorescence image of retinal cross section. . . . 154
5.16 Segmentation of 3D phantom. . . . 155
5.17 Segmentation of 3D MRI data. . . . 156

6.1 Example of segmentation editing for two labels. . . . 167
6.2 Results of segmentation editing for two labels. . . . 168
6.3 Example of segmentation editing for three labels. . . . 170
6.4 Results of segmentation editing for four-label case. . . . 171
6.5 Retina detachment. . . . 173
6.6 Retina images stained with TOPRO. . . . 174
6.7 ONL thickness and cell density profiles. . . . 176
6.8 Thickness and density values. . . . 177
6.9 Thickness and density for mosaiced images. . . . 179
6.10 Retinas stained with rod opsin and GFAP. . . . 180
6.11 Segmentation results for rod opsin and GFAP stained retinas. . . . 182
6.12 Slicing of retinal layers into sublayers. . . . 183

6.13 Antibody expression levels. 185

7.1 Hierarchical layer segmentation of Ascidian. 189
7.2 Multilabel higher order Potts clique graph. 190

Chapter 1

Introduction

Image segmentation is the partitioning of an image into smaller regions composed of pixels that share similar characteristics or properties. It is an essential prerequisite for many fundamental computer vision tasks, such as object recognition, scene interpretation, and content-based image retrieval. Image segmentation plays an increasingly important role as various scientific disciplines become more reliant on image data and as image acquisition devices become cheaper and more accessible. For example, in the biomedical domain, image segmentation is vital for fast, accurate, and reproducible information extraction from large image datasets, the analysis of which would otherwise require extensive manual effort. Image segmentation if also heavily utilized in other scientific disciplines, such as the geosciences, psychology, and marine biology. Despite its wide use, the segmentation task is often one of the main time bottlenecks in

many information extraction pipelines, even when automatic algorithms are used. This is because developing robust pixel grouping criteria that facilitate semantically meaningful segmentations remains a major challenge, and existing algorithms often require intensive manual input and editing of the results.

This thesis presents a set of image segmentation algorithms that utilize prior information available from domain knowledge. These information priors are incorporated into the segmentation framework to reduce the inherently ill-posedness of the segmentation problem and constrain the results to a more semantically meaningful solution space. The set of priors includes information about an object's shape, the topology of the image regions, the spatial relationships among objects in a serial image stack, and even a previous segmentation result for which further editing is needed. The proposed segmentation algorithms are formulated in the discrete domain using Random Field models, and for many instances these algorithms guarantee globally optimal solutions with respect to the Random Field energy. For a majority of these algorithms, the development is motivated mainly by challenges stemming from bioimage analysis, but they can be readily applied to image data from a wide variety of applications. Using graph cuts for optimization, these algorithms are computationally efficient and are easily extensible to segmentation of N-dimensional image data.

In this chapter, we briefly discuss the utility of prior information in some state-of-the-art segmentation methods and impact it has had on improving the segmentation

results. Then we highlight several key challenges in image analysis that motivate our work and discuss the justification supporting our use of combinatorial graph cuts for optimization. Finally we provide a summary of our major contributions and briefly outline the organization of the dissertation.

1.1 Segmentation with Prior Information

Image segmentation is an inverse problem, where given one or very few observed images, the task is to infer the spatial region layout that generated these images. Since both the space of all images as well as the space of all possible partitionings are extremely large, image segmentation is an ill-posed problem given the small number of observations. For example, a low resolution three channel color image of size 32×32 resides in a space of 3072 dimensions. Using 8 bits to encode the values for each dimension, there are a total of nearly 10^{7400} possible images. This number is extremely large especially if we consider that a human in a 100 years only see a total of 10^{11} frames (at 30 frames/second) [108]. Despite this overwhelmingly large image space, it is well known that the spatial layout of the image regions is not random but highly correlated [103, 109], and thus the images that we do encounter are often relegated to a much smaller, and to a certain extent, more manageable subspace.

Generally speaking, all image segmentation algorithms utilize some form of prior information based on visual grouping or perceptual organizing principles [95, 119]. The Gestalt laws of proximity, similarity, closure, continuity and symmetry are often integrated into the mathematical formulation of the segmentation cost functional. To control the smoothness of the segmented regions, penalties for long contour length and high contour curvature are added [55, 16, 18]. These penalties are often weighted by the support of image information, such as using the intensity gradient to improve detection and localization of edges [120, 57]. Conversely, to facilitate segmentation of thin structures, *i.e*. to encourage longer contours, the criterion is to maximize the flux of the gradient field[112]. Along the same line, pixels are grouped according to their visual coherence, which is often measured by intra-regional similarity and inter-regional dissimilarity [101]. For example, the segmentation can penalize region groupings that are inhomogeneous in intensity [18] or texture [84].

1.2 Semantic Priors

While the syntactic priors above act to control the intrinsic form of the resulting image regions, they prove to be inadequate for segmenting more challenging images, such as those with low signal-to-noise ratio, high background clutter, or significant occlusions of the desired object. Recent research efforts are more focused on incorporating

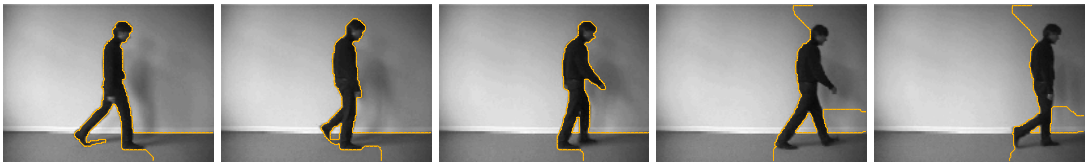


Figure 1.1. Segmentation of a walking sequence using only intensity information. Uneven illumination of the scene leads to progressively worse results. Video sequence obtained from [99].

higher-level semantic priors into the segmentation framework to better address these challenges. In this section, we discuss several of these priors that are relevant to our work, including shape priors, region topology priors, multiview priors, and presegmentation priors. We would like to point out that there are other types of higher-level priors, such as appearance modeling [23], but we do not discuss them here in detail.

1.2.1 Shape Priors

Prior knowledge of an object's shape often enhances our ability to delineate the object from its surroundings, especially when the object is occluded, in a cluttered scene, or undergo changes in illumination. For example, figure 1.1 shows several frames from a video of a person walking across a room. As the person moves closer to the right, the uneven illumination causes the person's grayscale value to be similar to that of the wall and floor. As the results show, using intensity information alone will lead to incorrect segmentation.

Shape prior information is typically added to the segmentation cost as a term penalizing deviations between the segmentation and the prior shape. A low cost is incurred when the resulting contour closely matches that of the prior shape, while a high cost is incurred when the segmentation does not resemble the prior shape. The addition of a shape prior model has proven to produce accurate results even in the presence of strong image noise and large object occlusions [46, 24, 71, 93, 111]. This is especially true for segmentation tasks involving medical or biological image data, where noise is inherent in the imaging process [71, 52]. Despite the popularity of shape prior segmentation algorithms in the continuous domain–parametric snakes and level sets–the use of shape information is still limited in the discrete domain and has only recently been introduced into graphical models [30, 39, 98]. In chapter 4, we introduce a novel shape prior segmentation algorithm based on graph cuts that offer several advantages over both existing continuous and discrete methods.

1.2.2 Multiview Priors

For many image analysis applications, the segmentation algorithms are primarily designed to operate only on a single image at a time. However, there are situations where the image dataset contains multiple images of the same or similar objects at varying viewpoints. Instead of segmenting the objects in each image independently, the information redundancy available from the other images should be exploited to improve the

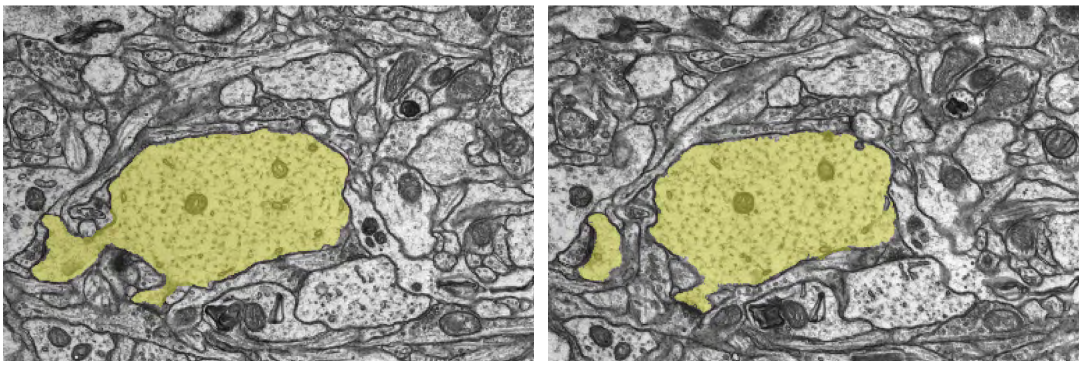


Figure 1.2. Adjacent images from a serial EM image stack. Note the large branching that has occured for the highlighted object.

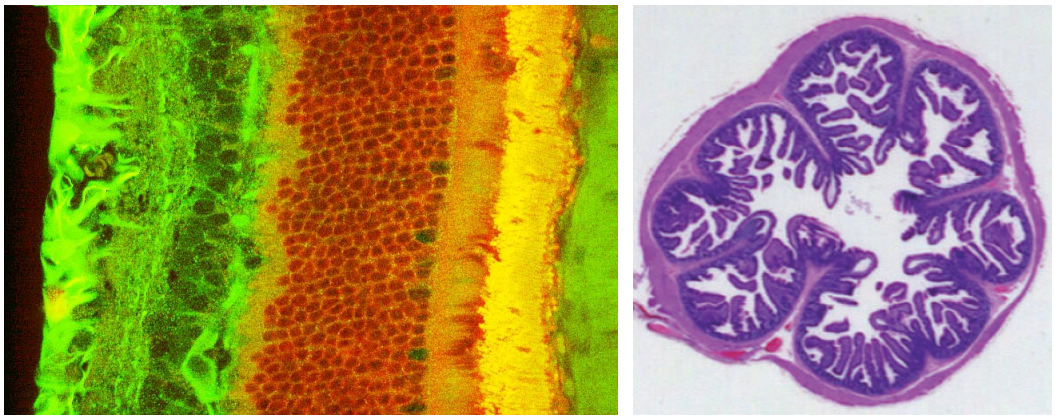
segmentation accuracy. In [123], Yezzi *et al*. propose to simultaneously segment and register two images by iteratively mapping the contour from one image onto the other image during curve evolution. The method is able to segment two images from different modalities, *e.g*. MRI and CT, but the two segmentation must have nearly identical shapes. Riklin-Raviv *et al*. [89] proposed a similar framework where the segmentation from one image is used as a shape prior for the other image. One shape can differ from the other by having a missing or extra part, and the segmentation can fill in or exclude the shape difference. However, the overall shape must be nearly identical up to a projective transformation. Using the visual appearance of regions from two images, such as matching the regions' histograms [91] or pairwise pixel similarity [2], have also shown to improve segmentation.

There are several examples from the biomedical domain where multiple images of an object are used for segmentation and 3D reconstruction. In cryo-electron micrograph

(cryo-EM) experiments, up to several thousand nearly identical, but randomly arranged, macromolecules are imaged to generate a collection of all possible viewpoints, which are then used to reconstruct the 3D structure [31]. Multiple view information about an object can also come from the 2D cross sections obtained through serial sectioning of the 3D object. An example of this type of data is the serial EM image. Figure 1.2 shows two adjacent images from a serial EM stack. Note that the 2D contours of the highlighted object exhibit high variability in shape and topology, even though these are consecutive images in the stack. The previous methods ([123, 89, 91, 2]) would fail to correctly segment these two contours due to the large shape differences between the contours and the lack of distinctive visual feature separating the object of interest from background objects. In chapter 3, we present a segmentation algorithm capable of handling the high shape variations between contours in adjacent images. The algorithm is not limited to operating on two images at a time, but is able to segment the 2D contours from the entire serial image stack.

1.2.3 Region Topology Priors

Images of biological specimens, ranging in scale from whole organisms to smaller tissue sections, often exhibit a consistent spatial arrangement of neighboring anatomical regions. Figure 1.3 shows two examples of images that have consistent layer ordering. These spatial layer relationships are consistent when imaging similar structures



(a) Retina cross section (b) Jejunum cross section

Figure 1.3. Examples of images exhibiting nested region topologies.

from very different specimens. Not surprisingly, the layer topology directly reflects the anatomical structures being imaged and consequently is constrained to have the same spatial layout for specimens of the same species.

Image segmentation methods able to take advantage of the consistent spatial relationships among neighboring structures should produce significantly better results. Algorithms enforcing topology constraints have proven successful in separating two adjacent objects sharing a very thin boundary [49, 48, 50]. Methods that incorporate spatial information about neighboring objects have also outperformed those methods using image data alone [122]. For dealing with images exhibiting nested region topologies, Chung and Vese [20] proposed a multilayer level set framework, where the region interfaces are embedded onto the different levels of a single level set function. How-

ever their method suffers from sensitivity to initialization and does not guarantee the globally optimal solution. Ishikawa [51] proposed a graph cut framework that, in theory, could also be used to segment images with nested layers. However in practice, this algorithm is limited by numerical errors and inefficiency in the implementation. In chapter 5, we propose a graph cut algorithm algorithm that employ the nested layer topology prior to compute the globally optimal segmentation.

1.2.4 Presegmentation Priors

User interaction is an indispensable part of image segmentation when the image signal-to-noise ratio (snr) is low or when the target object is occluded, has missing parts, or is located in a cluttered background. These scenarios are frequently encountered in biomedical image analysis because of several factors: imaging systems are inherently noisy, tissue organization causes occlusions, and neighboring objects have similar visual characteristics. Figure 1.4 shows an example of an EM image, where most structures have similar visual characteristics. The thin dark boundaries are the main visual cue that can be used to segment the structures.

User interaction can also be useful when a given segmentation result is imperfect and requires further editing. Most automatic or semiautomatic algorithms produce segmentations that are locally or globally optimal for given a cost function. Commonly, the cost function is a weighted sum of metrics defined using the priors discussed above,

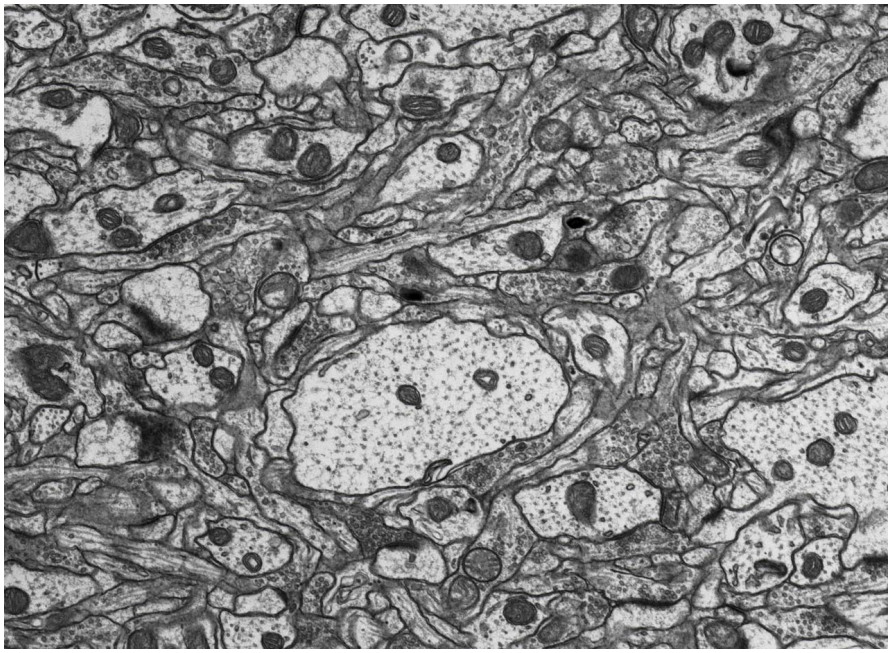


Figure 1.4. EM image of a rat brain tissue.

and the weights of the various terms are usually set by the user. Even when the globally optimal segmentation, with respect of the cost function, is found, there is no guarantee that the result correctly corresponds to what the user may have expected. Parameter tuning and learning can help diminish this discrepancy, but doing so may be infeasible or time consuming. The are several algorithms that allow for editing of the segmentation results. Using parametric snakes, Carlbom *et al*. [15] allow the user to interact with the contour during the curve evolution process to correct for errors. Boykov and Jolly [7] proposed a graph cut algorithm that allows the user to select a small set of object and background pixels for both segmentation and editing. In chapter 6, we present an

interactive editing algorithm that is very efficient and provide visually intuitive results. Given a presegmentation prior computed from a previous segmentation, the user can make edits quickly with minimal interaction.

1.3 Why Combinatorial Graph Cuts?

The segmentation algorithms presented in this thesis use graph cuts as the main optimization tool, and we provide several justifications for our choice. Image segmentation using combinatorial graph cuts has proven to be an attractive alternative to traditional segmentation algorithms such as parametric snakes [106, 55] and implicit active contours [77, 16, 56, 18]. Whereas the latter techniques iteratively minimize continuous energy functionals by numerical schemes using finite approximations, graph cuts minimizes discrete energy functions exactly by combinatorial optimization on graphs. Because graph cuts does not use approximations or iterative gradient descent, the algorithm does not suffer from wrongly converging to spurious local minima, which often maligns continuous methods. Furthermore, graph cuts is guaranteed to find the global minimum for certain binary or two class segmentation problem [45]. For the multi-label problem, graph cuts is proven to converge to within a small factor of the global minimum [10, 65], while no such guarantee can be made for continuous methods. Note

that our work deals with combinatorial graph cuts, and not spectral graph methods such as Normalized Cuts [101].

Graph cuts also share many attractive properties with continuous methods such as variational level sets. Like level sets, graph cut methods exhibit topological flexibility due to the implicit representation of the curve. The graph cut framework can also incorporate boundary and regional constraints [63], as well as prior information such as object shape [39]. Much like the level set formulation, graph cuts is easily extensible to N-dimensional problems [11]. Moreover, graph cut algorithms are inherently stable, unlike many gradient descent methods that require careful design of the time step to maintain stability. Although the computational efficiency of graph cuts is comparable to continuous methods [9], graph cuts often outperform continuous methods that require small time steps.

1.4 Summary of Contributions

The overall contribution of this thesis work is the development of novel segmentation algorithms that utilize higher-level semantic priors to constrain the space of feasible solutions and improve segmentation results. The algorithms are formulated using fandom field models and the random field energies are minimized with graph cuts. The contributions of our work are summarized below.

- **Segmentation of serial EM image stacks:** We developed a set of segmentation algorithms to facilitate the 3D reconstruction of neuronal structures from serial EM image stacks. The algorithms are efficient and allow quick, interactive user selection of the object of interest. The first algorithm segments planar contours from a single 2D image by minimizing an energy function defined using the image intensity and the flux of the intensity gradient field. The second algorithm uses the information redundancy from adjacent slices in the serial image stack to segment all the planar cross-sections of the entire object. The segmentation result of one image slice is used as a geometric prior to constrain the segmentation of the adjacent image slice. This method is more computationally efficient than performing full 3D segmentation and can cope with large deformations in the object's shape between adjacent images.

- **Affine invariant segmentation of multiples object using shape priors:** We developed a new graph cut segmentation algorithm using a shape distance metric that is both symmetrical and obeys the triangle inequality. This shape distance is commonly used in level sets, but has not been extended to graph cuts. To simultaneously segment multiple objects, we developed a multiphase graph cut approach to handle object overlap, where a pixel can have multiple object memberships (labels). This is fundamentally different from the traditional multiway cuts in that the latter techniques can only assign one label per pixel. The multi-

phase graph cuts has a level set counterpart, but is more computationally efficient both in terms of memory requirements and speed of convergence. Our shape prior energy can also incorporate multiple shape priors, which is necessary when there is variability in the shape. Moreover, our algorithm can segment noisy, occluded objects under affine transformation without the need to estimate transformation parameters by gradient descent, as is commonly done in level set literature.

- **Globally optimal nested layer segmentation:** We developed an algorithm to solve the problem of segmenting images with known nested region topology. Using a label adjacency constraint on the pairwise pixel labeling, we show that the multi-label problem, generally NP-hard, can be solved exactly with graph cuts. Our graph construction is a significant improvement in terms of size and numerical stability over an existing method [51] that can, in theory, be used to solve the nested layer segmentation problem but in practice, encounters several implementation issues. Our method is also straightforwardly extensible to 3D and higher dimensional image data.

- **Interactive editing of a presegmentation:** We developed an interactive editing method, formulated as an energy minimization using graph cuts. Given a pre-segmentation result and a set of user edit markings, the algorithm can efficiently compute a new segmentation. The segmentation energy is defined using both the

image data and the presegmentation prior. The geodesic distance from a pixel to the user marking is used as an intuitive measure of the amount of presegmentation refinement required. Our algorithm is more efficient than methods that do not use a presegmentation prior, and offers a viable alternative to more time consuming tasks such as parameter tuning and learning.

- **Bioimage analysis applications:** We developed two information extraction and analysis framework using the layer segmentation results for confocal images of retina cross sections. Given the segmentation, the first method computes the layer thickness and photoreceptor nuclei density in the Outer Nuclear Layer. The second method computes the spatial distribution of glial fibrillary acidic protein (GFAP) and Rod Opsin antibody expressions across the retinal layers. These analysis methods provide quantitative metrics of retinal restructuring during detachment experiments.

1.5 Organization of Thesis

The remainder of this thesis is organized into 6 chapters. Chapter 2 provides the background information on Random Field models and graph cuts that are necessary for the development of the algorithms in subsequent chapters. An overview of several state of the art segmentation algorithms are also provided. Chapter 3 describes the set of

algorithms developed for segmentation of serial EM image stacks. Chapter 4 presents the shape prior segmentation algorithms, including the multiphase graph cuts and the extension to multiple prior shapes. The nested layer segmentation algorithm is detailed in chapter 5, along with a comparison that shows the implementation problems associated with the Ishikawa method [51]. Chapter 6 discuss topics that are more closely related to bioimage analysis and is divided into three main sections. The first section describes the presegmentation editing algorithm, which can be considered the final step in the segmentation process before information extraction and analysis. The second section describes the ONL thickness and nuclei density computation, and the last section describes the calculation of the antibody distribution. Chapter 7 concludes this dissertation by discussing potential research directions.

Chapter 2

Random Fields and Graph Cuts in Image Segmentation

This chapter provides an overview of image segmentation, focusing mainly on the use of Random Fields to model the segmentation problem and the graph cut algorithms used to obtain solutions to this problem. Image segmentation has a long history, and providing a comprehensive survey of the existing literature is beyond the scope of this thesis. Instead we briefly discuss some current state-of-the-art segmentation techniques and then turn our focus to Random Field models and graph cuts. The details provided here are necessary for understanding the algorithms presented in subsequent chapters.

2.1 Image Segmentation Overview

Early image segmentation approaches often rely on a series of heuristics in their algorithm to compute a suitable segmentation [42], and although some early methods are still in use, for the most part they have been replaced in the computer vision research community by more principled statistical methods. The statistical formulation models the segmentation problem as a maximization of a posterior probability, and the optimal solution is found my minimizing the associated cost functional. Current state-of-the-art algorithms can be divided into two groups according to how they model the spatial domain of the image. The pioneering works of Geman and Geman [40] and Besag [4] represent the image pixels as discrete nodes in a graph, while those of Mumford and Shah [82] and Zhu and Yuille [125] model the image as a continuous subset in $\mathbb{R}^2$ ($\mathbb{R}^3$ for 3D).

In the continuous framework, typically an energy functional is defined to measure the "goodness" of a particular segmentation and then using variational calculus techniques, a gradient descent is performed to find the segmentation with the lowest energy. The parametric active contour or snake model [55, 106] uses a deformable contour with an associated energy functional defined as a sum of terms that capture the intrinsic and extrinsic properties of the contour. Intrinsic properties include the contour length, elasticity, and stiffness, and extrinsic properties are defined base on the image data, mainly

the intensity gradient. The snake is iteratively deformed during each iteration to decrease its energy until convergence is reached. However, the parametric representation limits the object boundary to be a single contour, and it is difficult to incorporate region based metrics into the energy definition.

To overcome these restrictions, the level set framework was proposed whereby the deformable contour is embedded onto the zero level set of a continuous Lipschitz function [83, 77]. During gradient decent, the embedded contour is implicitly evolved according to deformations in the level set function. As a result, the topological changes in the contour occur automatically during the gradient descent without the need to keep track of the contour points. Secondly, the notion of regions, *e.g*. inside and outside, is well defined for the level set, which allows for the straightforward addition of region based terms in the energy functional. Among the popular level set methods are the geodesic active contours [16, 56] and variational level sets [124, 18]. Although most active contour algorithms deal with only the two label problem, there are variants, such as the multiphase level sets [113], that can handle the multi-label case.

A general limitation of continuous methods is that, since these algorithms rely upon gradient descent to find the optimal solution, they are prone to converge onto local minima. In practice, good initializations are required in order to ensure satisfactory results. Recently, Juan *et al*. propose to incorporate a stochastic element into the curve evolution to help avoid premature convergence [54]. Nonetheless, there are no guarantee for

finding the global optimal solution. The numerical implementation of active contour methods also present its own challenges. Since the contour or level set must be discretized, the iterative gradient descent can become unstable, and careful attention must be paid to control the update step size. As a consequent, the algorithm can take a large number of iterations to converge.

In the discrete formulation, the image pixels are represented as discrete nodes in a graph, and the node connectivity (via edges) indicates the dependency among the pixel nodes. This graphical representation is referred to as the Random Fields model, and we discuss it in more details starting in section 2.2.1. Unlike the continuous formulation, the segmentation is now viewed more as a pixel labeling problem, where the labels indicate the region membership. The goal is then to find the labeling that minimizes a discrete energy function. Despite the elegant statistical formulation of early models [40, 4], the lack of efficient optimization algorithms has limited their wide use. Recently there has been renewed interest in the Random Field models due to the development of efficient graph cut algorithms for optimization [7, 9]. We defer the discussion on graph cuts to section 2.4.

There are also notable semi-discrete models that have recently gained in popularity. The Normalized Cuts algorithm of Shi and Malik [101] uses a generalized eigenvalue method to approximately solve a graph-partition problem. Their image partitioning or cut criterion measures both the total dissimilarity between the different pixel groups as

well as the total similarity within the groups. For the two label problem, the eigenvector corresponding to the second smallest eigenvalue is thresholded to yield two regions. For multiple labels, the authors proposed to either use a recursive thresholding for this eigenvector or use the all of the top eigenvectors as indicator vectors for clustering and pruning. One setback of the Normalized Cuts is that the eigenvalue problem can become intractable for typical images and some resizing are usually required.

The Random Walker (RW) algorithm proposed by Grady [43] is another state-of-the-art segmentation algorithm that operates on graphs. For each label, the method requires the user to interactively mark a small set of pixels indicating the regions that should have that label. Then the RW algorithm computes the probability that a random walker starting at an unlabeled pixel will reach one of the marked pixels. The computation involves solving a linear system of equations formulated using electrical circuit analogies. The unlabeled pixel is assigned the label of the marked pixel that has the greatest RW probability. The RW algorithm can handle the multi-label case with moderate increase in computational resources. Similar to the Normalized Cuts, powerful linear systems solvers become necessary for large image sizes or densely connected graphs.

2.2 MAP Estimation for Discrete Models

In this section, the background information on random field models, particulary Markov Random Fields and Conditional Random Fields, are provided. The algorithms presented in this thesis rely upon the random field formulation to model the segmentation task, as well as the subsequent optimization methods used to perform the segmentation.

2.2.1 Random Fields

The Random Field model [72] consists of an undirected graph $\mathcal{G}(\mathcal{V}_{RF}, \mathcal{E}_{RF})$ composed of a set of vertices or sites $\mathcal{V}_{RF} = \{1, \ldots, N\}$ and a set of undirected edges $\mathcal{E}_{RF}$ connecting neighboring sites in $\mathcal{V}_{RF}$. Typically $\mathcal{V}_{RF}$ is the set of pixels $\mathcal{P}$ in the image, and we will use $\mathcal{V}_{RF}$ and $\mathcal{P}$ interchangeably. The neighborhood is usually specified by a 4- or 8-connectivity of the pixels for a 2D image and 6- or 26-connectivity for a 3D image. A site q is a neighbor of p if they are connected by an edge $(p, q) \in \mathcal{E}_{RF}$, and the set of neighbors of p is denoted $\mathcal{N}_p$. Furthermore, the sites $\mathcal{P}$ has an associated random field $\mathbf{Y} = \{Y_p : p \in \mathcal{P}\}$, where each random variable Y_p takes one of K values from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_K\}$. An edge $(p, q) \in \mathcal{E}_{RF}$ connecting sites p and q indicates a dependency between Y_p and Y_q. In addition, a clique c is defined as a set of sites, such that $\forall p, q \in c$, $p \in \mathcal{N}_q$ and $q \in \mathcal{N}_p$, and the associated random variables

$\mathbf{Y}_c = \{Y_p, p \in c\}$ are dependent on each other. That is, each site in the clique is connected to all other sites, and a clique if often referred to as a fully connected subgraph of $\mathcal{G}$.

The joint event $\{Y_1 = y_1, \ldots, Y_N = y_N\}$, with $y_p \in \mathcal{L}$, is called a configuration or realization of the random field $\mathbf{Y}$. For convenience, the joint event is denoted $\mathbf{Y} = \mathbf{y}$, where $\mathbf{y} = \{y_p \ : \ p \in \mathcal{P}\}$. We will refer to $\mathbf{y}$ as a labeling of the sites $\mathcal{P}$. Then image segmentation becomes an estimation or inference problem where the image $\mathbf{x}$ is an observation of some underlying random field $\mathbf{Y}$, and the goal is to find the maximum *a posteriori* (MAP) estimate of the underlying field given the observation, *i.e.* we seek a labeling $\mathbf{y}^*$ such that

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}} \Pr(\mathbf{y}|\mathbf{x}), \tag{2.1}$$

where the space of all possible labelings is denoted $\mathcal{Y} = \mathcal{L} \times \mathcal{L} \times \cdots \times \mathcal{L}$ (N times) $= \mathcal{L}^N$ and has cardinality K^N.

2.2.2 Markov Random Fields

The most popular random field model is the Markov Random Field (MRF) first introduced into the vision community by the seminal works of Geman and Geman [40] and Besag [4]. The MRF model offers a principled framework to incorporate local contextual constraints in image segmentation. Using Bayes's rule, the posterior is

expressed as [72]

$$\Pr(\mathbf{y}|\mathbf{x}) \propto \Pr(\mathbf{x},\mathbf{y}) = \Pr(\mathbf{x}|\mathbf{y})\Pr(\mathbf{y}). \tag{2.2}$$

The term $\Pr(\mathbf{x}|\mathbf{y})$ is the likelihood of an observation $\mathbf{x}$ given $\mathbf{y}$ and captures the dependency of the labels on the observation. In the case of image denoising, this term can be interpreted as the model for sensor noise. The term $\Pr(\mathbf{y})$, which indicates the probability of a particular labeling $\mathbf{y}$ among all labelings in $\mathcal{Y}$, is a spatially varying prior that can be expressed as a locally dependent MRF.

The MRF is a random field that obeys the following two properties with respect to the neighborhood system $\mathcal{N} = \{\mathcal{N}_p : p \in \mathcal{P}\}$:

$$\text{Positivity:} \qquad \Pr(\mathbf{y}) > 0, \ \ \forall \mathbf{y} \in \mathcal{Y}, \tag{2.3a}$$

$$\text{Markovian:} \qquad \Pr(y_p|\mathbf{y}_{\mathcal{P}\setminus\{p\}}) = \Pr(y_p|\mathbf{y}_{\mathcal{N}_p}), \ \ \forall p \in \mathcal{P}. \tag{2.3b}$$

Here, for simplicity we denoted $\Pr(\mathbf{Y} = \mathbf{y})$ as $\Pr(\mathbf{y})$, $\Pr(Y_p = y_p|\cdot)$ as $\Pr(y_p|\cdot)$, and $\mathbf{y}_{\mathcal{P}\setminus\{p\}} = \{y_p : p \in \mathcal{P} \setminus \{p\}\}$. Equation (2.3a) ensures that all configurations $\mathbf{y} \in \mathcal{Y}$ are probable, while equation (2.3b) simply states that a realization y_p at site p is only dependent on the realizations at the neighbors of p specified by the set $\mathcal{N}_p$.

Using the Hammersley-Clifford theorem [47, 3], the prior $\Pr(\mathbf{y})$ can be expressed as a Gibbs distribution

$$\Pr(\mathbf{y}) \propto \exp\left(-\sum_{c\in\mathcal{C}} V_c(\mathbf{y}_c)\right), \tag{2.4}$$

where $\mathcal{C}$ is the set of all the cliques in the graph and $V_c(\mathbf{y}_c)$ is a clique potential that describes the energy of a particular labeling $\mathbf{y}_c = \{y_p : p \in c\}$ for a clique $c \in \mathcal{C}$.

Although the Gibbs distribution is a convenient representation for the prior term, the presence of the data likelihood term $\Pr(\mathbf{x}|\mathbf{y})$ makes the MAP estimation problem difficult in general. Commonly, to make the problem more tractable, the likelihood term is approximated by assuming that the observation $\mathbf{x} = \{x_1, \ldots, x_N\}$ is conditionally independent given the labels, *i.e.*

$$\Pr(\mathbf{x}|\mathbf{y}) = \prod_{p \in \mathcal{P}} \Pr(x_p|y_p). \tag{2.5}$$

Then the posterior is given by

$$\Pr(\mathbf{y}|\mathbf{x}) \propto \exp\left(\sum_{p \in \mathcal{P}} \log \Pr(x_p|y_p) - \sum_{c \in \mathcal{C}} V_c(\mathbf{y}_c)\right). \tag{2.6}$$

The conditional independence assumption implies that equation (2.6) is also an MRF [72], which facilitates the MAP estimation by allowing the use of existing inference tools for MRFs. Then the MAP estimate of $\Pr(\mathbf{y}|\mathbf{x})$ is the minimizer of

$$-\log \Pr(\mathbf{y}|\mathbf{x}) = E(\mathbf{y}) = \sum_{p \in \mathcal{P}} V_p(y_p) + \sum_{c \in \mathcal{C}} V_c(\mathbf{y}_c), \tag{2.7}$$

where $V_p(y_p) = -\log \Pr(x_p|y_p)$. Here we use $E(\mathbf{y})$ to denote the MRF energy, and while the observation $\mathbf{x}$ is not explicitly included in the argument, it is understood that the log likelihood term is observation dependent. We defer the discussion on energy minimizations until section 2.3.

Despite its popularity, the MRF model has several limitations [69]. First, it is well known that the pixels in natural images are not conditionally independent but exhibit strong correlations [103]. Yet the conditional independence assumption for the likelihood term, for the sake of computational tractability, does not accurately reflect the nature of the pixel features and fails to capture the long range spatial dependencies among pixels. Second, the prior term $\Pr(\mathbf{y})$, which models the interaction among the labels, does not depend on the data observation $\mathbf{x}$ and consequently limits the potential to incorporate any data dependency in the label interactions. Traditionally, the prior term is used to encourage smoothness of the labels so that neighboring pixels are more likely to be assigned similar labels [40]. However, the smoothness constraint becomes problematic at discontinuities such as edges, causing over smoothing.

2.2.3 Conditional Random Fields

To remedy the shortcomings of MRFs, Lafferty *et al*. [70] proposed the Conditional Random Field (CRF), which directly models the posterior distribution $\Pr(\mathbf{y}|\mathbf{x})$ as a Gibbs field. In contrast to the MRF where the posterior is maximized by maximizing the joint distribution $\Pr(\mathbf{x}, \mathbf{y})$ (see equation (2.2)), the CRF model is a discriminative framework which does not explicitly model the prior $\Pr(\mathbf{y})$. The CRF model has been subsequently extended to 2D lattices [68] and has been shown to outperform the traditional MRF model for many vision tasks [102, 60, 69].

In this section, we review the CRFs formulation of Lafferty *et al.* [70]. Given an observation $\mathbf{x}$, $\mathbf{Y}$ is a conditional random field (CRF) if, when conditioned on $\mathbf{x}$, the random variables Y_p obey the following two properties with respect to the neighborhood system $\mathcal{N}$:

$$\text{Positivity:} \qquad \Pr(\mathbf{y}|\mathbf{x}) > 0, \quad \forall \mathbf{y} \in \mathcal{Y}, \tag{2.8a}$$

$$\text{Markovian:} \qquad \Pr(y_p|\mathbf{x}, \mathbf{y}_{\mathcal{V}-\{p\}}) = \Pr(y_p|\mathbf{x}, \mathbf{y}_{\mathcal{N}_p}), \quad \forall p \in \mathcal{P}. \tag{2.8b}$$

These two conditions are analogous to the conditions in equation (2.3) for the MRF, with the exception that the probabilities are all conditioned on the observation $\mathbf{x}$. The CRF can be considered a Markov random field (MRF) globally conditioned on the observation $\mathbf{x}$ [69].

Again by the Hammersley-Clifford theorem, the posterior distribution $\Pr(\mathbf{y}|\mathbf{x})$ over the labelings of the CRF can be expressed as a Gibbs distribution

$$\Pr(\mathbf{y}|\mathbf{x}) \propto \exp\left(-\sum_{c\in\mathcal{C}} V_c(\mathbf{y}_c|\mathbf{x})\right), \tag{2.9}$$

where $\mathcal{C}$ is the set of all cliques, and $V_c(\mathbf{y}_c|\mathbf{x})$ is the potential function of the clique c given the observation $\mathbf{x}$. The CRF energy is given by

$$E(\mathbf{y}|\mathbf{x}) = \sum_{c\in\mathcal{C}} V_c(\mathbf{y}_c|\mathbf{x}), \tag{2.10}$$

and the the most probable or MAP labeling $\mathbf{y}^*$ of the CRF is given by

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\mathcal{Y}} \Pr(\mathbf{y}|\mathbf{x}) = \arg\min_{\mathbf{y}\in\mathcal{Y}} E(\mathbf{y}|\mathbf{x}). \tag{2.11}$$

We will generally refer to the MAP estimation as an energy minimization problem.

Though the dependency of the clique potentials on $\mathbf{x}$ seems to be a trivial difference between the previous model, the CRF offers several key advantages over the MRF formulation. First the CRF unary potential defined over a single site can be a function of the entire observation $\mathbf{x}$, whereas the unary potential, *i.e*. log-likelihood term, in the MRF model is a function of only the data at that single site. This enables the use of discriminative classifiers to compute the unary potential [69]. Second, the pairwise and possibly higher order clique potentials, which model the interaction among the labels, can include a data dependency. For example, the observed data can be used to provide support for assigning similar labels to neighboring sites. In fact it has been shown that by modulating the pairwise potential in the MRF model with the image intensity gradient, a significant improvement in labeling accuracy can be seen at discontinuities such as edges [7, 5]. For a more in depth comparison of MRF and CRF models, we refer the reader to [69].

2.2.4 CRFs for Image Segmentation

As we have mentioned, the image segmentation task can be posed as a pixel labeling problem modeled by a random field. The set of sites $\mathcal{V}_{RF}$ is composed of the image pixels $\mathcal{P}$, and the label set $\mathcal{L}$ denotes the different regions in the image. The goal is to assign labels $y_p \in \mathcal{L}$ to all pixels $p \in \mathcal{P}$ such that the CRF energy in equation (2.10) is

minimized. Typically, the clique potentials are defined over unary and pairwise sites, so that the CRF energy becomes

$$E(\mathbf{y}|\mathbf{x}) = \sum_{p \in \mathcal{P}} V_p(y_p|\mathbf{x}) + \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} V_{pq}(y_p, y_q|\mathbf{x}), \tag{2.12}$$

where $V_p(y_p|\mathbf{x})$ and $V_{pq}(y_p, y_q|\mathbf{x})$ are the unary and pairwise clique potentials or costs, respectively. Higher order cliques are also possible [65, 58], but the minimization can become computationally expensive. In this work, we will make use of the higher order $\mathcal{P}^n$ Potts potential cliques proposed by Kohli *et al*. [58].

In equation (2.12) the unary cost for assigning label y_p to pixel p is often taken to be the negative log-likelihood of the class conditional density $\Pr(y_p|x_p)$, but can be the output of any discriminant classifier [69]. This data association cost favors pixel labelings that have the highest likelihood of belonging to a particular label given the observation. The pairwise cost for assigning label pair $\{y_p, y_p\} \in \mathcal{L}$ to neighboring pixel pair $\{p, q\} \in \mathcal{P}$ is often defined as a contrast sensitive function, which encourages neighboring sites with similar attributes to have the same label. We will discuss the unary and pairwise costs more thoroughly in the subsequent chapters when their details become necessary.

2.3 Energy Minimization

As note in section 2.3, the labeling space $\mathcal{Y} = \mathcal{L}^N$ is very large with cardinality $|\mathcal{Y}| = K^N$, where K is the number of labels and N is the number of sites or pixels in the image. In general, the solution for equation (2.11) is NP-hard [65], and only approximations can be obtained. In this section, we review popular algorithms used to solve equation (2.11). However, we start by introducing the concept of submodular functions of discrete variables, a principle necessary to understand how the form of the clique potential affects the minimization of equation (2.10).

2.3.1 Submodular Functions

The submodularity of a function of discrete variables plays an analogous role to that of the convexity of a function of continuous variables. It is well known that submodular functions can be minimized in polynomial time [6], especially for the case of binary labels [45, 65]. Thus it is important to determine the submodularity of a function so that an optimal solution can be found efficiently.

Before we state the results concerning submodular functions, we define the projection of a function.

Definition 2.3.1 *A projection of a function $f : \mathcal{L}^n \rightarrow \mathbb{R}$ on m variables is a function $f^s : \mathcal{L}^m \rightarrow \mathbb{R}$ which is obtained by fixing the values of $n - m$ arguments of $f(\cdot)$. Here s refers to the set of variables whose values have been fixed* [58].

For example, $f^s(v_3, v_4, \ldots, v_n) = f(a, b, v_3, v_4, \ldots, v_n)$ is the projection of a function $f(v_1, v_2, \ldots, v_n)$ onto the first two variables, and a and b are fixed constants. Using definition 2.3.1, a submodular function of binary variables is defined below.

Definition 2.3.2 *All functions of one binary variable are submodular. A function f of two binary variables is submodular if and only if*

$$f(0,0) + f(1,1) \leq f(0,1) + f(1,0). \tag{2.13}$$

A function $f : \mathcal{L}^n \rightarrow \mathbb{R}$ is submodular if and only if all its projections on 2 variables are submodular [65, 58].

Thus for a multi-label function with more than two variables, we can determine its submodularity by verifying whether its projections on all possible variable pairs are submodular.

Schlesinger and Flach [97] present the following conditions to determine submodularity of a multi-label function. Let $\mathcal{L}$ be an ordered label set such that any label pair in $\mathcal{L}$ has an "above/below" relationship. For any label $l_i \in \mathcal{L}$, we denote l_{i+1} to be the lowest label above l_i. Then a function $f : \mathcal{L}^2 \rightarrow \mathbb{R}$ is submodular if

$$f(l_i, l_j) + f(l_{i+1}, l_{j+1}) \leq f(l_{i+1}, l_j) + f(l_i, l_{j+1}) \tag{2.14}$$

for all pairs $\{l_i, l_j\}$ in the label set. Note that the notion of submodularity for multi-label functions is dependent on having an ordered label set. However, Schlesinger [96] proposed a technique to find an ordering, if one exists, for which the function becomes submodular, and thus we can consider the notion of submodularity for multi-label functions independent of an ordered label set. We will use the notation $\mathcal{M}_s^k$ to denote the class of submodular functions of k multi-label variables and $\mathcal{F}_s^k$ to denote the class of submodular functions of k binary variables.

2.3.2 Energy Minimization Algorithms

Several popular methods, most notably graph cuts, have been used to minimize the energy in equation (2.12). There are few special cases where the energy can be minimized exactly: 1) when $\mathcal{L} = \{0, 1\}$ and $V_{pq}(y_p, y_q|\mathbf{x})$ is submodular, *i.e.* it is in the class $\mathcal{F}_s^2$ [45, 65], and 2) when $\mathcal{L}$ is an ordered set and $V_{pq}(y_p, y_q|\mathbf{x})$ is submodular [51, 97]. However, the majority of vision problems do not fall into these categories and thus only approximate solutions can be found [10, 105]. Below, we review some of the most common energy minimization algorithms and defer the in-depth discussion on graph cuts to the next section.

Move Making Algorithms

Move making algorithms refer to iterative techniques where starting from an initial estimate, the labeling is updated at each iteration to reduce the energy in equation (2.10). Convergence to the final solution is reached when no further energy reduction can be made. Geman and Geman [40] used simulated annealing to compute the MAP estimate. At each step, a label change is proposed for a randomly chosen site, and the move is accepted according to some probability that is dependent of the change in the energy and the current annealing temperature. The temperature schedule is designed to accept the majority of label changes in the at the onset, even when they increase the energy. As the simulation progresses, the temperature is slowly decreased so that only moves that reduce the energy are likely to be accepted.

For most image segmentation applications where the number of pixels is large, simulated annealing can be very slow even for binary labels, and using "practical" temperature schedules can lead to unsatisfactory results [45]. To expedite the computation, Besag [4] proposed a form of simulated annealing with zero temperature called Iterated Conditional Modes (ICM). At each iteration, the method makes a greedy move by changing a single label that yield the largest decrease in energy. Though convergence is guaranteed and occurs very rapidly, ICM is extremely sensitive to the initial estimate and usually converges to one of many local minima [105].

Recently two move making algorithms, α-expansion and $\alpha\beta$-swap [10], based on combinatorial graph cuts have gained wide acceptance in the vision community, due in large part to their efficiency and guarantees of convergence to strong local minima. Moreover, they have proven to be superior to the earlier forms of move making algorithms and are better than message passing algorithms for segmentation task, both in accuracy and efficiency [105].

For each iteration of the α-expansion algorithm, a move is made to expand the set of pixels labeled α, and the label expansion that results in the maximum decrease in the energy for all labels is kept. Convergence is reached when there are no further expansion moves for any labels that can decrease the energy. In each iteration of the $\alpha\beta$-swap algorithm, the two pixel sets with labels α and β are selected, and a swap move is made to interchange the labels α and β among these pixels. The swap move for a pair of labels among all possible pairs which decreases the energy by a maximum amount is selected. As before, convergence is reached when there are no swap moves that can further decrease the energy. For the binary label case, only a single iteration of either the expansion or swap algorithm is required. In both of these algorithms, the optimal move is computed efficiently at each iteration using graph cuts. We will discuss graph cuts in section 2.4.

One major difference between the expansion and swap moves and the standard moves of simulated annealing and ICM is that the label changes in each iteration are

not limited to a single site. Instead, large subgroups of pixels can have their label changed after each expansion or swap move. As a result of these large moves, both the α-expansion and $\alpha\beta$-swap algorithms are more successful at avoiding shallow local minima and converge to strong local minima. Kolmogorov and Zabih [65] showed that the expansion algorithm can only be used for minimizing an energy where V_{pq} satisfies, for all labels α β and γ,

$$V_{pq}(\alpha,\alpha) + V_{pq}(\beta,\gamma) \leq V_{pq}(\alpha,\gamma) + V_{pq}(\beta,\alpha). \tag{2.15}$$

Similarly, the swap algorithm can be used if for all labels α and β,

$$V_{pq}(\alpha,\alpha) + V_{pq}(\beta,\beta) \leq V_{pq}(\alpha,\beta) + V_{pq}(\beta,\alpha). \tag{2.16}$$

Though these conditions seemingly limit the types of energies that can be minimized, most common pairwise potentials in segmentation applications can be defined to satisfy these requirements.

Message Passing Algorithms

Although the minimization algorithms presented in this thesis fall under the category of move making algorithms, we would like to briefly mention several popular message passing algorithms for minimizing the random field energy. The interested reader can find a more in depth review of some of these algorithms in [105]. Message passing algorithms compute the minimum energy label by passing messages between the sites

in the random field. The max-product belief propagation (BP) [85] algorithm passes messages forward and backward along the rows and columns of the site lattice and chooses the label with the highest belief for each site. The BP algorithm has strong convergence guarantees, similar to the expansion and swap algorithms, if it converges. However BP can become stuck in an infinite loop switching between labels that have equal min-marginals. Similar to BP, the tree-reweighted message passing algorithm (TRW) [62, 66] propagates messages between the sites. The label computation for a particular site is dependent on the max-marginal of a set of trees that include that site. The TRW computes the lower bound on the energy, but does not stop when this bound is reached. In practice, the labeling with the lowest energy to date is kept and returned when the algorithm terminates. Another variant of the message passing algorithms is the dual decomposition method in [117], where the authors proposed a tree-relaxed linear program method and a tree-reweighted max-product message-passing method to solve the MAP problem. These methods attempt to compute tight upper bound of the MAP configuration when possible.

2.4 Graph Cuts

In this section, we briefly describe some necessary background on graph cuts, which is commonly used to minimize equation (2.10). The graph cut algorithm uses a weighted

directed graph $G(\mathcal{V}, \mathcal{E})$ composed of a set of nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ connecting the nodes. The node set includes two special terminal nodes, the source s and the sink t, and the remaining nodes are considered neighborhood nodes. A directed edge $(u, v) \in \mathcal{E}$ connects node $u \in \mathcal{V}$ to node $v \in \mathcal{V}$ and has nonnegative capacity or weight w_{uv}. Note that because the graph is directed, $(u, v) \neq (v, u)$.

A subset of edges $\mathcal{E}_c \subset \mathcal{E}$ is called an st-cut if the terminal nodes are completely separated in the induced graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E} \setminus \mathcal{E}_c)$. That is there are no forward paths from terminal s to terminal t when all edges in the cut are removed. Hence, the cut partitions the nodes into disjoint subsets $\mathcal{S}$ and $\mathcal{T}$ where $s \in \mathcal{S}$ and $t \in \mathcal{T}$. In our convention, an edge $(p, q) \in \mathcal{E}$ from node u to node v is in the cut if $u \in \mathcal{S}$ and $v \in \mathcal{T}$. For simplicity, we will refer to the st-cut simply as a cut. The cost of the cut is the sum of all the edge weights in $\mathcal{E}_c$. For a given graph, the minimum cost cut (mincut) can be found by solving an equivalent maximum flow (maxflow) problem [35], which we will discuss shortly.

2.4.1 Graph Structure for Image Segmentation

The goal of using graph cuts for minimizing equation (2.10) is to construct a graph such that: 1) there is a one-to-one mapping between cuts in the graph and labelings of the image pixels, and 2) the mincut cost is equal to the minimum energy (up to a constant). In this section, we describe the graph construction presented in [65] for

minimizing the submodular CRF energies of binary variables and defer the discussion on minimizing the multi-label energy to chapter 5. Typically, the boolean energies encountered in image segmentation are limited to unary and pairwise cliques–first and second order terms–to keep the optimization tractable. We begin by describing the graph construction for this type of energy.

The node set $\mathcal{V}$ is normally composed of the pixel set and the terminal nodes, *i.e.* $\mathcal{V} = \{\mathcal{P}, s, t\}$. We will refer to the pixel nodes in $\mathcal{P}$ as neighborhood nodes (n-nodes) and use the notations p and q to denote these nodes. Let $\theta_{p;i}$ be the cost of label assignment $y_p = i$ for $i \in \{0, 1\}$ so that the unary potential in equation (2.12) can be reexpressed as

$$V_p(y_p|\mathbf{x}) = \theta_{p;0}\overline{y}_p + \theta_{p;1}y_p, \tag{2.17}$$

where $\overline{y}_p = 1 - y_p$. We can use a graph node p to encode the the boolean variable y_p by assigning $y_p = 0$ if $p \in \mathcal{T}$ and $y_p = 1$ if $p \in \mathcal{S}$ after a cut. In this way, we can use N graph nodes to encode the binary labeling $\mathbf{y} = \{y_p : p \in \mathcal{P}\}$, and any cut on the graph will result in a corresponding labeling. To ensure that the mincut corresponds to the lowest energy labeling, we must construct an equivalent graph with appropriate edge weights. Notice if $\theta_{p;1} - \theta_{p;0} \geq 0$, the potential in equation (2.17) can be written as $V_p(y_p|\mathbf{x}) = (\theta_{p;1} - \theta_{p;0})y_p + \theta_{p;0}$, and minimizing this potential is equivalent to finding the mincut for the graph in figure 2.1(a). Obviously for this graph, the mincut will have

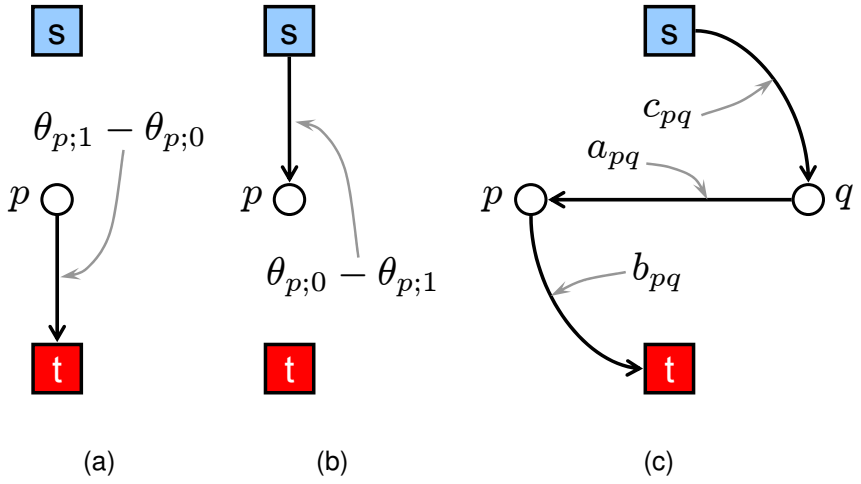


Figure 2.1. Graphs (a) and (b) are used to encode the unary potential V_p. Graph (c) encodes the submodular pairwise potential V_{pq}. The edge weights are indicated in the figure.

$p \in \mathcal{T}$ and thus $y_p = 0$. If instead $\theta_{p;1} - \theta_{p;0} < 0$, then $V_p(y_p|\mathbf{x}) = (\theta_{p;1} - \theta_{p;0})\overline{y}_p + \theta_{p;1}$, which can be minimized by finding the mincut for the graph in figure 2.1(b).

Similarly, the second order boolean energy can be reexpressed as

$$\begin{aligned} V_{pq}(y_p, y_q|\mathbf{x}) &= \theta_{pq;00}\overline{y}_p\overline{y}_q + \theta_{pq;01}\overline{y}_p y_q + \theta_{pq;10} y_p\overline{y}_q + \theta_{pq;11} y_p y_q \\ &= a_{pq}\overline{y}_p y_q + b_{pq} y_p + c_{pq}\overline{y}_q + d_{pq}, \end{aligned} \tag{2.18}$$

where

$$a_{pq} = \theta_{pq;01} + \theta_{pq;10} - \theta_{pq;00} - \theta_{pq;11} \tag{2.19a}$$

$$b_{pq} = \theta_{pq;10} - \theta_{pq;00} \tag{2.19b}$$

$$c_{pq} = \theta_{pq;01} - \theta_{pq;11} \tag{2.19c}$$

$$d_{pq} = \theta_{pq;00} + \theta_{pq;11} - \theta_{pq;10}. \tag{2.19d}$$

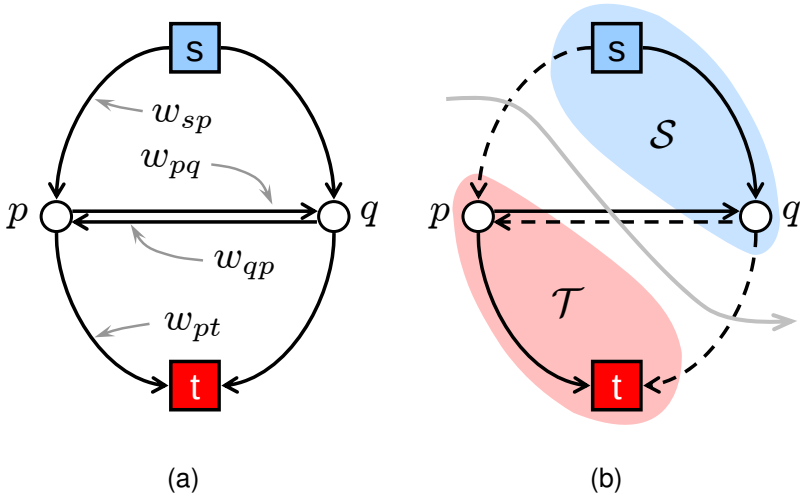


Figure 2.2. (a) Simplified graph used for representing the submodular binary second order CRF energy in equation (2.12). The edge weights are indicated in the figure. (b) Example of an st-cut, where $p \in \mathcal{T}$ and $q \in \mathcal{S}$ correspond to the labelings $y_p = 0$ and $y_q = 1$, respectively. Edges in the cut are shown as dashed arrows.

and $\theta_{pq;ij}$ is the cost of the label assignment $\{y_p = i, y_q = j\}$. The graph structure with two nodes p and q in figure 2.1(c) is used to encode this energy. Note that since d_{pq} is a constant, we ignore this term in the minimization and it is not included in the graph weight. According to this graph structure, the cost of cutting edge (p, t), or equivalently the label assignment $\{y_p = 1, y_q = 1\}$, is b_{pq}. Likewise, the cost of cutting (s, q), equivalently $\{y_p = 0, y_q = 0\}$, is c_{pq}. Finally, the cost of cutting edge (q, p) or label assignment $\{y_p = 0, y_q = 1\}$ is a_{pq}. The additive property of graphs [65] allows the graphs in figure 2.1 to be combined, where directed edges connect the same node pair are added.

Table 2.1. Edge weight assignments for graph in figure 2.2(a).

Edge	Weight
(s, p)	$w_{sp} = \theta_{p;0}$
(p, t)	$w_{pt} = \theta_{p;1}$
(p, q)	$w_{pq} = \theta_{pq;10}$
(q, p)	$w_{qp} = \theta_{pq;01}$

For the pairwise potentials used in this work, the cost $V_{pq}(i, i) = 0$ and we use a slightly more intuitive, but less compact, graph representation in figure 2.2(a). The weight assignments are given in table 2.1. Figure 2.2(b) shows an example of a cut on this graph corresponding to the labeling $\{y_p = 0, y_q = 1\}$. The unary terminal edges (s, p) and (q, t) are in the cut, and they contribute $w_{sp} + w_{qt}$ to the cut cost. The neighborhood edge (q, p) is in the cut, but not edge (p, q). This is because our convention dictates that an edge $(p, q) \in \mathcal{E}$ from node p to node q is in the cut if $p \in \mathcal{S}$ and $q \in \mathcal{T}$.

Recently, Kohli *et al*. [58] introduced a higher order clique (HOC) potential of arbitrary clique size called the $\mathcal{P}^n$ Potts potential. This HOC potential can capture long range interactions among the pixels in a large image patch and has proven to be very powerful in modeling image features such as texture [94, 59]. We use the HOC potential in this work and briefly describe it here. The $\mathcal{P}^n$ Potts potential for a clique c with

size $|c| = n$ is defined as

$$V_c(\mathbf{y}_c) = \begin{cases} \gamma_i & \text{if } y_p = i, \forall p \in c, \\ \gamma_{\max} & \text{otherwise.} \end{cases} \tag{2.20}$$

Here, $\mathbf{y}_c = \{y_p : p \in c\}$, and equation (2.20) simply states that if all the pixels in clique c are assigned the same label i, then a total cost of γ_i is incurred. However, if the pixels in c have mixed label assignments, then a cost of $\gamma_{\max} > \gamma_i$ is incurred. For the two label case, the HOC potential can be express as

$$V_c(\mathbf{y}_c) = \begin{cases} \gamma_0 & \text{if } y_p = 0, \forall p \in c, \\ \gamma_1 & \text{if } y_p = 1, \forall p \in c, \\ \gamma_0 + \gamma_1 & \text{otherwise.} \end{cases} \tag{2.21}$$

The graph representation of the Potts HOC potential requires the use of two auxiliary nodes c_s and c_t for each HOC clique c. The graph structure is shown in figure 2.3(a) and the edge weight assignments are given by

$$w_{sc} = \gamma_0 \tag{2.22a}$$

$$w_{ct} = \gamma_1. \tag{2.22b}$$

Consider the the cuts shown in figures 2.3(b) and 2.3(c), where the blue region indicates nodes in the source set $\mathcal{S}$ and the red region indicates nodes in the sink set $\mathcal{T}$. Figure 2.3(b) corresponds to the case $y_p = 1, \forall p \in c$ and edge (c_t, t) with weight w_{ct}

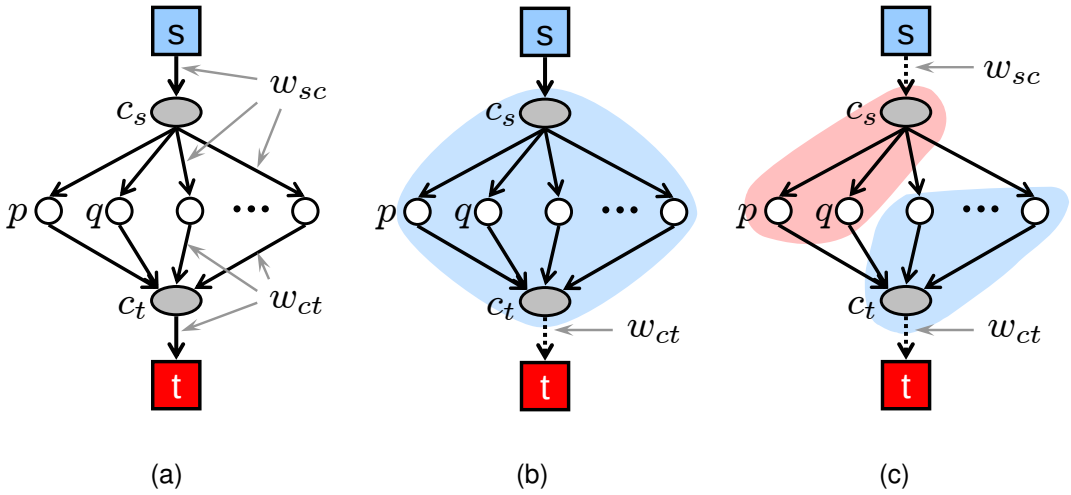


Figure 2.3. (a) Graph construction for optimizing the $\mathcal{P}^n$ Potts clique potential. The edge weights are indicated in the figure. (b) A cut with resulting labeling $y_p = 1, \forall p \in c$. (c) A cut with mixed labeling of the nodes. The neighborhood and terminal edges for the pixel nodes are not shown.

is in the cut. If instead edge (s, c_s) is in the cut, then the clique assignment would be $y_p = 0, \forall p \in c$. It is straightforward to see that these cuts have costs γ_1 and γ_0, respectively. Figure 2.3(c) shows the situation when the clique nodes have mixed labelings. Recall that the clique graph and the second order graph in figure 2.2(a) can be additively combined so that it is entirely possible for the mixed labeling case to occur. Although not shown, the combination of neighborhood and terminal edges can cause the clique to have mixed labels. Both edges (s, c_s) and (c_t, t) are in the cut in figure 2.3(c), and a penalty $w_{sc} + w_{ct}$ is incurred.

2.4.2 Finding the Mincut by Maximum Flow

Ford and Fulkerson [35] showed that finding the st-mincut is equivalent to computing the maximum flow (maxflow) from the source s to the sink t through a capacitated network. Formally let $G(\mathcal{V}, \mathcal{E})$ be a capacitated network with nonnegative capacity or weight w_{uv}. The goal is to compute the maximum flow f from the source s to the sink t subject to the edge capacity and flow balance constraints

$$0 \leq f_{uv} \leq w_{uv} \quad \forall (u, v) \in \mathcal{E}, \text{ and} \tag{2.23}$$

$$\sum_{v \in \mathcal{N}_u} (f_{vu} - f_{uv}) = 0 \quad \forall v \in \mathcal{V} \setminus \{s, t\}, \tag{2.24}$$

where f_{uv} is the flow from node u to node v, and $\mathcal{N}_v$ is the set of connected neighbors of v. Conceptually, the maxflow algorithm begins by pushing flow through the network from source to sink. As the flow increases, network edges will saturate. The maximum flow is reached when there is no more increase in flow, *i.e.* there is no path from source to sink containing an unsaturated edge. The sum of the capacities of saturated edges separating the source from the sink gives the maximum flow and hence the mincut cost.

Many polynomial time algorithms exist for solving the maxflow problem and are based mainly on either the Push-Relabel algorithm of Goldberg and Tarjan [41] or the Augmenting-Path algorithm of Ford and Fulkerson [35]. The worse-case runtime complexity for these algorithm is on the order of $\mathcal{O}(|\mathcal{E}| \cdot |\mathcal{V}|^2)$. For the experiments in this work, we use the maxflow algorithm of Boykov and Kolmogorov [9], which

has a worse-case runtime complexity of $\mathcal{O}(|\mathcal{E}| \cdot |\mathcal{V}|^2 \cdot C)$ where C is the cost of the minimum cut. However, since this algorithm is designed specifically for vision graphs, it outperforms traditional algorithms designed to operate on general graphs.

Chapter 3

Segmentation of Serial Electron Micrographs

For many image analysis applications, the segmentation algorithms are primarily designed to operate only on a single image at a time. However, there are situations where the image dataset contains multiple images of the same or similar objects at varying viewpoints. Instead of segmenting the objects in each image independently, the information redundancy contained within the other images from the set should be exploited to improve the segmentation accuracy. There are several examples from the biomedical domain where multiple images of an object are used for segmentation and 3D reconstruction. In cryo-electron micrograph (cryo-EM) experiments, up to several thousand nearly identical, but randomly arranged, macromolecules are imaged to obtain a collec-

tion of all possible viewpoints. These viewpoints are then used to reconstruct the 3D structure [31]. For larger objects such as cells and tissue sections, the 3D reconstruction is performed on serial image stacks acquired either through optically or physically sectioning the specimen. The cross sectional views of the object are combined to form the 3D model.

In this chapter, we present a set of segmentation algorithms to facilitate the 3D reconstruction of neuronal structures from serial electron microscope (EM) image stacks. We first focus on the 2D segmentation problem, where the goal is to extract the cross-sectional contours of the object from a single image plane. The framework uses graph cuts to minimize an energy defined using the image intensity and the flux of the intensity gradient field. Then we develop a 2.5D segmentation framework that takes advantage of the information redundancy from adjacent slices in the serial image stack to inform the segmentation of the current slice. The algorithm uses the segmentation of one image slice as the geometric prior to constrain the segmentation of the adjacent image, and the new label is propagated through the serial stack. Using graph cuts, our algorithms are efficient and allows the user to quickly and interactively select the object of interest with a minimal amount of input markings. A preliminary version of the 2D segmentation algorithm appears in [115].

3.1 Segmentation of Neuronal Structures from Transmission EM

In many neurophysiological studies, understanding the neuronal circuitry of the brain requires detailed 3D information of the neuroanatomy. Currently, transmission electron microscopy (TEM) is the preferred modality used to capture high magnification views of neuronal membranes, synapses, and subcellular organelles. For small tissue volumes, Electron Tomography techniques are often used to image and reconstruct the 3D information from a single block [1, 31]. However to obtain the 3D data for a large volume, a tissue block must be shaved into ultrathin (40-60 nm) serial sections and each section or slice is imaged individually with TEM. Afterwards, 2D cross-sections of various structures are manually traced through the serial image stack and assembled to reconstruct the 3D model. Although there are software tools to perform the reconstruction (http://synapse-web.org/tools/index.stm), the tracing task is mostly done manually and remains tedious and time consuming. The need for more automated segmentation tools becomes especially pronounced in large EM studies and is one of the main bottlenecks in the creation of large anatomical databases [32].

To facilitate more efficient information extraction, we propose a framework to segment neuronal structures from serial EM images. We develop and test our algorithm using several serial EM image stacks from the Synapse Web [104]. An example of two

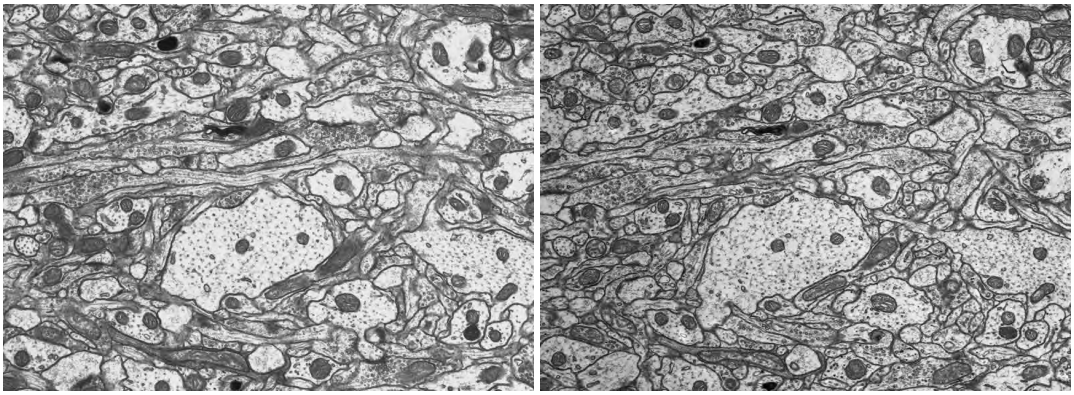
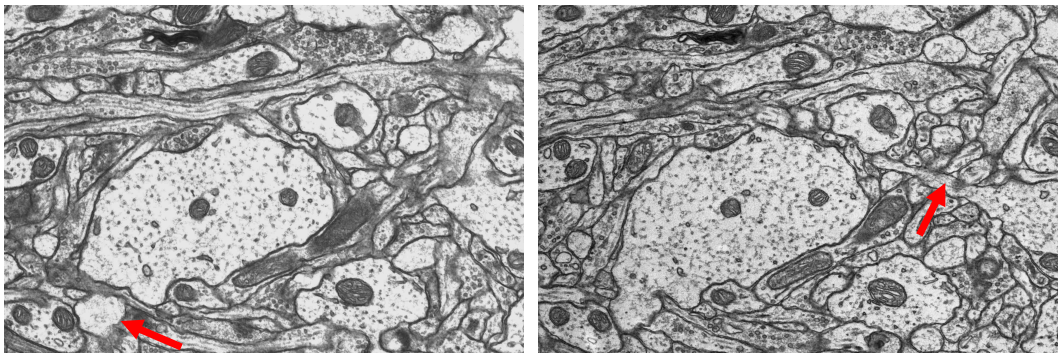


Figure 3.1. Two adjacent images from a serial EM stack. Note the illumination, level of contrast, and marked structural changes between the two images.

images from adjacent slices are shown in figure 3.1. Although the dataset is of comparatively good quality, the TEM images in this dataset present several challenges for segmentation. The images exhibit high variations in illumination, boundary contrast, and sharpness due to the imaging process. Since each serial slice is image separately, changes in the microscope parameters can cause marked photometric differences even in images from adjacent slices. Secondly, the serial reconstruction requires the slices to be ultrathin, which causes physical defects such as warping, anisotropic shrinkage, and nonuniform thickness throughout the slice. These physical artifacts manifest themselves in the final image and reduce the image quality.

Ideally if it were possible to section the tissue block into infinitely thin slices, then adjacent slices will contain a lot of information redundancy. Unfortunately, the thickness of the slice is determined by the physical slicing process, and thus each section is usually much thicker. As a result, the resolution in the z-axis is usually much lower



(a) Region merging. (b) region disappearance.

Figure 3.2. Two consecutive EM images from a serial stack. Note the intensity contrast and marked structural changes between the two images.

than that in the image plane. The resolution disparity between the z-axis and x,y-axes in physical sections is also much more pronounced than that seen in data acquired by optical slicing or by medical imaging modalities such as computed tomograph (CT) or magnetic resonance image (MRI). Because of the large gaps between the slices, large physical changes in the biological tissue can occur between two adjacent slices and much information can be lost. For example, contour merging, splitting, appearance, and disappearance are several events that often occur from one image to the next in a serial stack. The examples in figure 3.2 illustrate this point. From the left to right image, one large physical change is the merging of the object indicated by the red arrow in the left image. The red arrow in the right image indicates the appearance of a region that is missing from the left image. These events are typical for all the serial stacks in

our dataset. Due to these large changes, manual segmentation often requires the user to scroll through several adjacent images in order to detect the spatial differences.

The neuronal tissue structure also present a challenge for segmentation. In many segmentation tasks, a combination of an object's boundary and regional features are used to discriminate it from its surroundings or background. However, the major characteristic that separates an object from its background in the neuronal TEM image is the dark, albeit very thin membrane surrounding the object, and very little regional information is available. Because the image is a cross-section of a tissue volume densely packed with many similar neurons, there is often no visual distinction between the regional appearance of one neuron and the next. As figure 3.1 shows, the objects share a similar intensity and have no distinctive texture features. Consequently, the segmentation has to rely on the thin membrane as the major visual feature.

We would like to point out that in this work, we do not address the problem of registering the images in the serial stack, which in itself is a difficult challenge and remains an active area of research [126]. Secondly, we only provide the 2D planar profiles of the 3D object and do not deal with 3D visualization. However, there are available algorithms that can readily render the 3D structure given a set of 2D profiles [110]. Lastly, we do not perform any final anatomical analysis of the segmented neuronal structures, but acknowledge that the development of methods for anatomical measurements and

subsequent statistical analysis would certainly be beneficial for building a complete understanding of neurophysiological functions.

3.2 Segmentation of 2D EM Images

We begin by describing our 2D EM segmentation framework. The segmentation is posed as a minimization of an energy involving the image intensity and the flux of the intensity gradient field, and graph cuts is used to compute the globally optimal solution. The user needs to label only a small set of pixels indicating the object and background regions for each segmentation. We also make qualitative performance comparisons of our algorithm with the Random Walker algorithm [43].

3.2.1 Related Works on 2D EM Segmentation

There are several notable previous works on segmenting objects in TEM images. Carlbom *et al*. [15] developed a framework using parametric snakes [55] to segment neurons. Their method requires the user to provide a good estimate of the final contour and allows the user to interact with the snake to correct for errors during the segmentation. Fok *et al*. [34] also use snakes to segment nerve fibers in EM images. Although no user input is required, certain assumptions are made regarding the shape, size, and membrane thickness of the fibers that aid in the snake initialization. However, their

dataset is visually very different from ours, and the assumptions made in their method are not applicable. Both of the above methods use gradient descent for energy minimization and as a result, are sensitive to initializations and are prone to converging onto suboptimal solutions.

The method of Frangakis and Hegerl [37] is based on Normalized Cuts [101], where a generalized eigenvalue method is used to approximately solve a segmentation problem. The input images for their algorithm often contain only a small number of objects or regions, and each object is segmented in descending order of their saliency. However for our dataset, the object of interest is not always the most salient and the large image size can cause computational problems for the Normalized Cuts. Moreover, it is not apparent that their technique can allow for user interaction, which would help identify the objects of interest more readily. Recently, Chang *et al*. [19] use graph cuts to segment textured regions in TEM images. Their method assumes the object has discriminative texture features that help to differentiate it from the background, but as we have described above, this assumption is not applicable to the images in our dataset.

3.2.2 Interactive 2D Segmenation

We use the Conditional Random Field (CRF) model to solve the 2D segmentation problem. Given an image $\mathbf{x} = \{x_p : p \in \mathcal{P}\}$, we seek the minimum labeling $\mathbf{y} = \{y_p :$

$p \in \mathcal{P}, y_p \in \mathcal{L}\}$ of the second order CRF energy

$$E(\mathbf{y}) = \sum_{p \in \mathcal{P}} V_p(y_p) + \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} V_{pq}(y_p, y_q). \tag{3.1}$$

Here $\mathcal{P}$ is the set of image pixels, the label set $\mathcal{L} = \{0, 1\}$, and $\mathcal{N}_p$ is the set of neighbors of p. We use an 8-connected neighborhood system in all experiments. In equation (3.1) as well as for the remainder of this chapter, for convenience our notation do not explicitly indicate the dependency of the CRF energy and the clique potentials on $\mathbf{x}$, but it is assumed that this dependency exists. Note that this is a two label problem and thus for a submodular function V_{pq}, the energy can be minimized exactly with graph cuts. The graph construction is described in chapter 2. As a reminder, after an st-cut, the nodes are divided into two sets $\mathcal{S}$ and $\mathcal{T}$ so that $p \in \mathcal{S}$ if edge (p, t) is in the cut and $p \in \mathcal{T}$ if edge (s, p) is in the cut. Moreover, we assign the label $y_p = 0$ if $p \in \mathcal{T}$ and $y_p = 1$ if $p \in \mathcal{S}$. Pixels in the object will have label 0. Refer to chapter 2 for more details.

For a given image, the segmentation begins with the user placing markings to roughly indicate the sets of pixels belonging to the object and background. An example is shown in figure 3.3, where the orange stroke indicates the object and the green stroke indicates the background. Denote these preliminary object and background pixel sets as $\mathcal{O}$ and $\mathcal{B}$, respectively. The input information is incorporated into the graph by the following

terminal weight (t-weight) assignments:

$$w_{sp} = K, \ \forall p \in \mathcal{B}, \tag{3.2a}$$

$$w_{pt} = K, \ \forall p \in \mathcal{O}, \tag{3.2b}$$

where K is set to some large constant (10^3) to ensure that $\mathcal{O} \subset \mathcal{T}$ and $\mathcal{B} \subset \mathcal{S}$ after the cut.

3.2.3 Pairwise penalty using image intensity

As we have mentioned, the most salient feature that helps to separate an object of interest from the background objects is the dark membrane enclosing it. To take advantage of this feature, we use the pairwise Potts potential [63]

$$V_{pq}(y_p, y_q) = |y_p - y_q| \cdot g(x_p, x_q), \tag{3.3}$$

where

$$g(x_p, x_q) = \frac{1}{|p - q|} \left[0.001 + \exp\left(-\frac{(x_p - x_q)^2}{2\sigma_x^2} \right) \right]. \tag{3.4}$$

Here $x_p \in [0, 255]$ is the intensity value at p, $|p - q|$ is the Euclidian distance between pixels p and q, and σ_x is a parameter that controls the contrast sensitivity. The potential in equation (3.3) equals zero when the labels are the same and penalizes pairwise label assignments by an amount $g(x_p, x_q)$.

Equation (3.4) is a function of the difference between the intensities at p and q. Observe that at the object-membrane boundary, the intensity differences are often large

while they are smaller at pixel pairs within the object. The smaller differences increase the value of g and consequently discourage pairwise assignments with different labels. However, the opposite is true at the boundary, where smaller values of g decrease the costs of pairwise assignments with different labels. This is a desired behavior since we would like the label change to occur at the boundary. The constant 0.001 in equation (3.4) ensures that some minimum amount is paid for assigning different labels to two neighbors. Note that equation (3.3) satisfies the submodularity condition

$$V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(0,1) + V_{pq}(1,0), \tag{3.5}$$

and thus the CRF energy can be minimized exactly with graph cuts [65].

Using the simplified graph structure described in chapter 2, equation (3.1) is minimized by setting the weight of edges (p,q) and (q,p) as

$$w_{pq} = w_{qp} = g(x_p, x_q) \quad \forall p \in \mathcal{P},\ q \in \mathcal{N}_p. \tag{3.6}$$

Accordingly for node pairs spanning the membrane boundary, w_{pq} is small, and thus the cut is more likely to occur here than at node pairs inside the object. At this point, we can set the remaining t-weights as $w_{sp} = w_{pt} = 0,\ \forall p \in \mathcal{P} \setminus \{\mathcal{O}, \mathcal{B}\}$ (those not in equation (3.2)) and proceed with graph cuts. However as we will show, the algorithm becomes sensitive to small changes in σ_x and fails to segment convoluted regions of the object.

3.2.4 Results with Intensity

We test the algorithm on the serial EM image stacks in our dataset. For performance comparison, we use the Random Walker (RW) algorithm proposed by Grady [43], which is one of the state-of-the-art interactive segmentation algorithms available. Using the same set of user markings, the RW algorithm computes the probability that a random walker starting at an unlabeled pixel will reach one of the marked pixels. The unlabeled pixel is assigned the label of the marked pixel that has the greatest probability. The RW algorithm is formulated on a discrete lattice and is solved using the same neighborhood graph structure (without terminal nodes and edges) as that in our algorithm. Like our method, the RW also guarantee a globally optimal solution, but the optimality criterion is different from the energy in equation (3.1). Further similarities in the input marking requirements and graph structure makes the RW algorithm the ideal candidate for which to make comparisons.

For each image, the object and background marks were manually placed as shown in figures 3.3 and 3.4. Then both graph cuts and RW are run with parameters $\sigma_x = \{5, 10, 20\}$. As we have mentioned, σ_x controls the contrast sensitivity and acts similar to the standard deviation of the gaussian in equation (3.4). As sigma increases, g has a slower drop-off and the pairwise cost becomes less contrast sensitive. This means that the pairwise cost remains high and only becomes small at object-membrane boundaries with strong intensity contrast. The result is that the increase in σ_x induces the graph

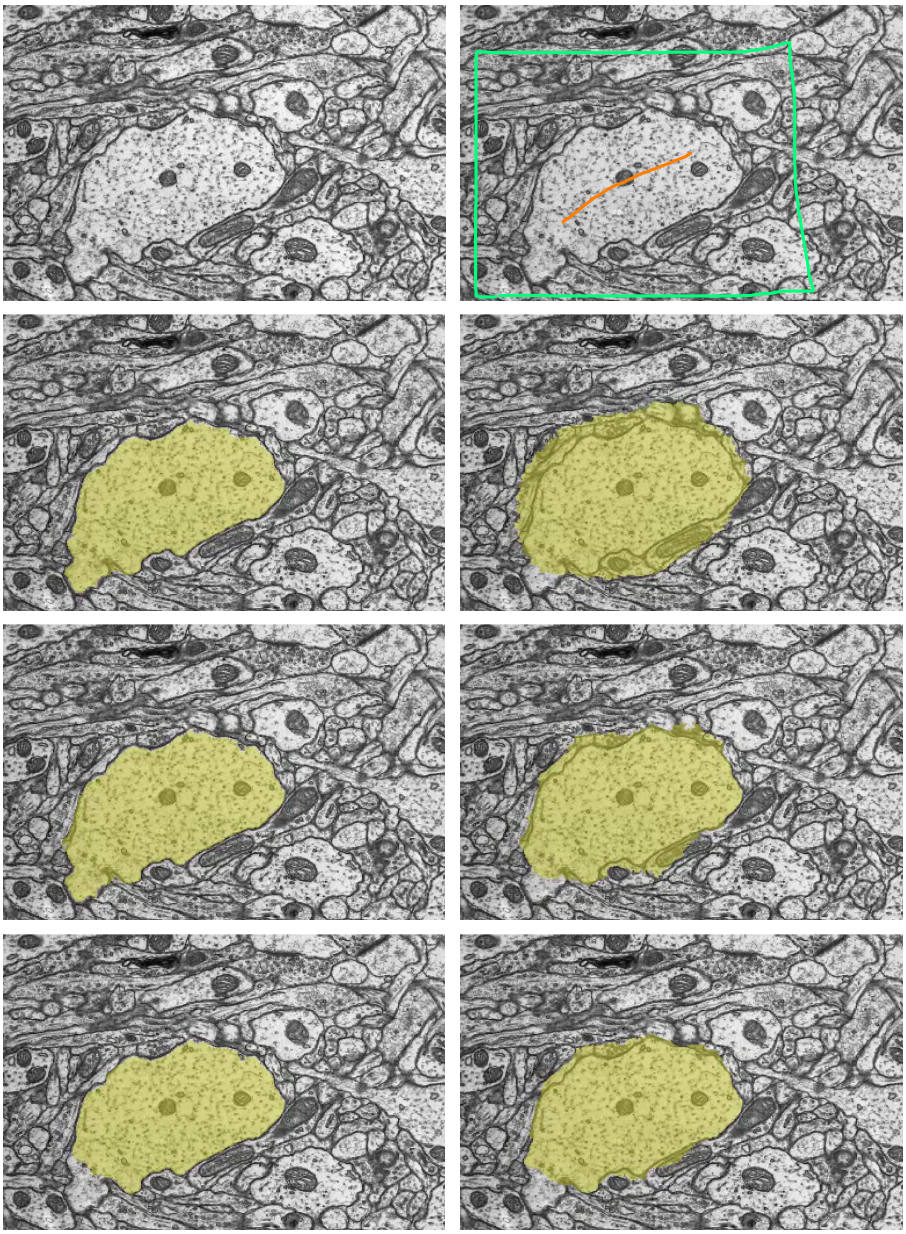


Figure 3.3. Sensitivity to of graph cuts and Random Walker to the parameter σ_x. (top) Original image 255×382 and user input. (rows 2-4) Results with increasing values of $\sigma_x = \{5, 10, 20\}$ for graph cuts (left) and Random Walker (right). As σ_x increases, the segmentation prefers smaller, more compact regions. Convoluted object parts are incorrectly labeled as background.

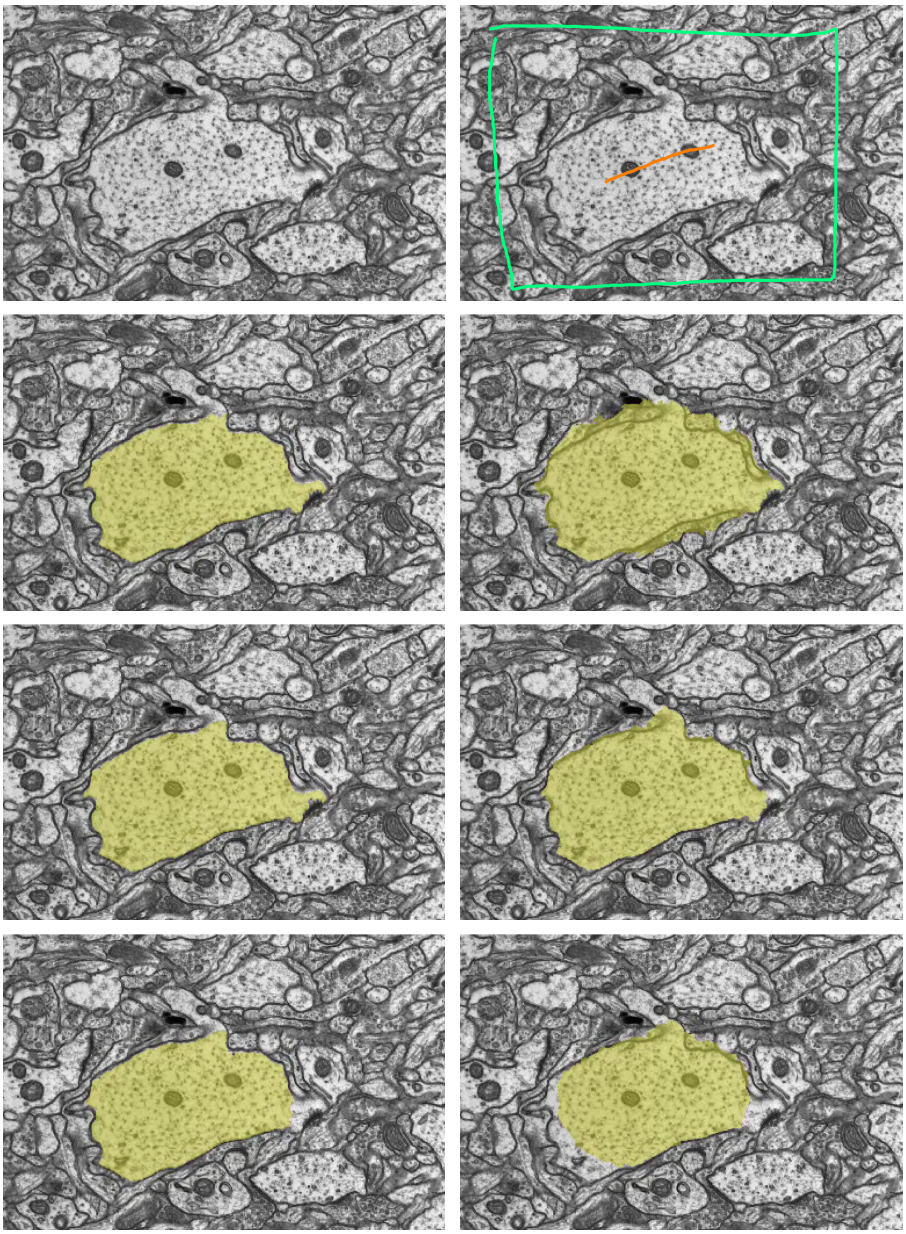


Figure 3.4. Sensitivity to of graph cuts and Random Walker to the parameter σ_x. (top) Original image 255×382 and user input. (rows 2-4) Results with increasing values of $\sigma_x = \{5, 10, 20\}$ for graph cuts (left) and Random Walker (right). As σ_x increases, the segmentation prefers smaller, more compact regions. Convoluted object parts are incorrectly labeled as background.

cuts to favor smaller, more compact cuts, since these cost less. This can be seen in the left column of rows 2-4 in figures 3.3 and 3.4.

For the RW algorithm, because of where the foreground and background markings are placed, a random walker starting at an unlabeled pixel between the two markings would have to travel across more membranes to reach the background labeled pixels compare to that in reaching the foreground labeled pixels. The random walker will have a smaller chance of crossing a membrane boundary for smaller values of σ_x because the edge weights would be small. This causes the object label to "leak" outside its boundary. As σ_x increases and consequently the edge weights become less contrast sensitive, the random walker improves its likelihood of crossing the membrane. Since there are often more background labeled pixels near the object boundary, the background label tends to "leak" into the object, especially at convoluted regions of the object. This can be seen in the right column of rows 2-4 in figures 3.3 and 3.4.

Next, we compare the two algorithm performances for different input markings but with the same $\sigma_x = 10$. Figures 3.5 and 3.6 show the results for the graph cut method in the middle column and and the RW results in the right column. The graph cut results are nearly identical for all three user inputs. This is because the contour of the minimum cut separating the object and background labels is still the object boundary. However, the RW method is more sensitive to changes in the location of the user markings because

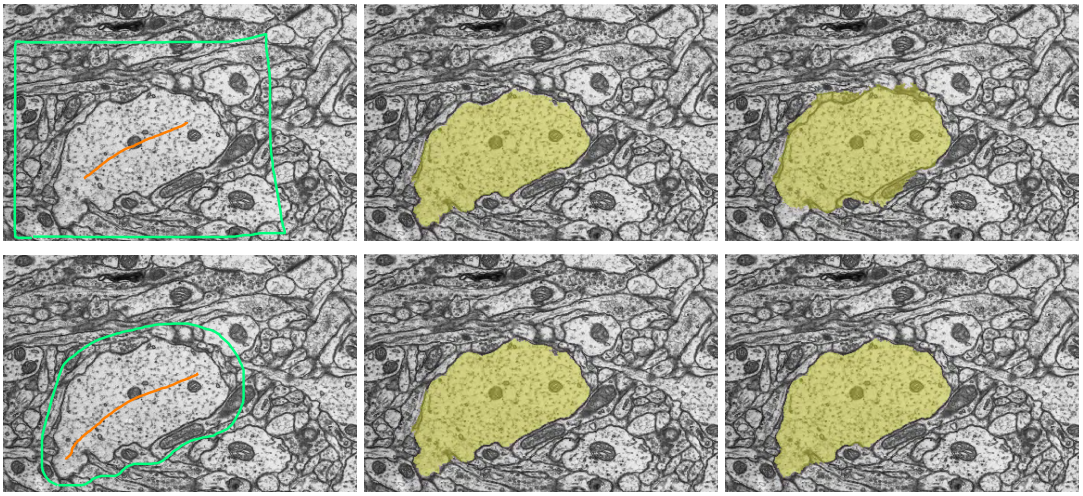


Figure 3.5. Sensitivity of graph cuts and Random Walker to input markings. (left) Input markings, (middle) graph cut results, (right) Random walker results.

the random walker probability is much more dependent on the distance (graph distance) to the labeled pixels.

The graph cut algorithm using the intensity based pairwise cost outperformed the RW method in computational efficiency as well. The running time in MATLAB for graph cuts is on average 0.3 seconds, while on average the RW takes 4 to 5 times longer. The RW computation requires solving a linear system of equations proportional to the graph size, so its runtime is expected to become much slower for larger graphs. Despite the promising qualitative results, there are several undesirable qualities that we would like to remedy. Both the graph cut and RW algorithms fails to segment smaller, convoluted parts of the object even for smaller settings of σ_x. Of course if better input markings closer to the objects are available, the segmentation would be more accurate.

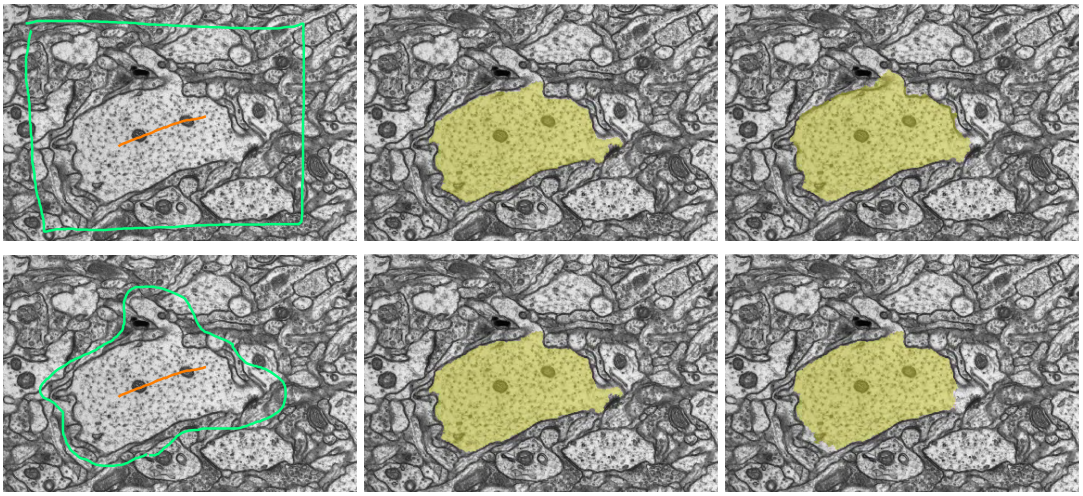


Figure 3.6. Sensitivity of graph cuts and Random Walker to input markings. (left) Input markings, (middle) graph cut results, (right) Random walker results.

However, forcing the user to provide detailed inputs would require greater user attention and is more time consuming.

3.2.5 Unary penalty using flux

The previous results are inaccurate because smaller cuts (lower costs) are favored and degrade when the object membrane is convoluted or contains gaps and noise. To improve the accuracy, we use a regional bias base on the flux of the intensity gradient field. Flux has been utilized in both level set [112] and graph cut [63] methods mainly to improve segmentation of thin structures such as blood vessels. In this work, flux is used to enhance the regional bias around the neuronal membranes. The added contrast prevents the cut from "pinching off" elongated regions of convoluted or noisy objects.

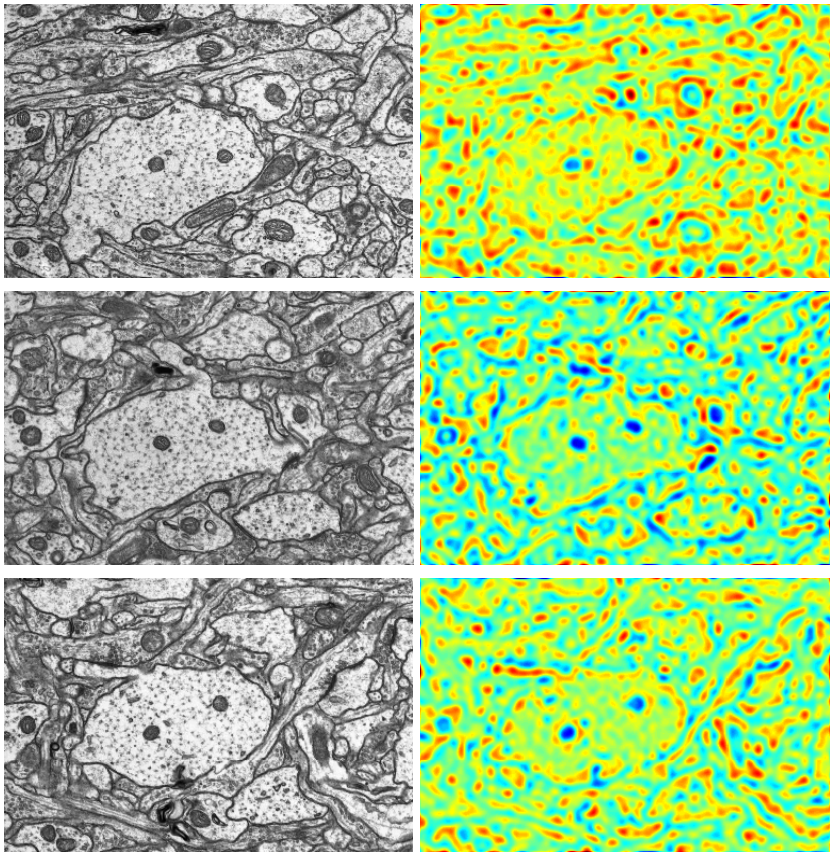


Figure 3.7. Flux of the gradient field (right) for the images on the left. Red values in the heat map indicate higher positive flux while blue values indicate negative flux.

The flux of a vector field $\mathbf{v}$ through a continuous hypersurface S is given by [63]

$$\varphi(S) = \int_S \langle \mathbf{v}, \hat{\mathbf{n}} \rangle \, dS, \tag{3.7}$$

where $\hat{\mathbf{n}}$ is the unit normal to the surface element dS and $\langle \ , \rangle$ is the Euclidian dot product. In this work, the field $\mathbf{v}$ is the normalized gradient of the gaussian smoothed image, *i.e.* $\frac{\nabla I_\sigma}{||\nabla I_\sigma||}$, and we set $\sigma = 3$. Numerically, the flux at a pixel p is computed by summing the dot products of the gradient field with the outward normals of a disk with

unit radius and centered at p. Several examples of the gradient field flux are shown in figure 3.7. The flux is more positive (redder values) in the lighter intensity regions of the object adjacent to the darker membranes, while it is more negative (bluer values) on the membrane side. This characteristic enhances the contrast between the foreground object and its surrounding membrane, especially when the membrane is blurry.

We would like to incorporate the flux into the CRF energy since it offers more visual discrimination for separating the object and the surrounding membrane. Notice from figure 3.7 that the flux provides a good regional bias and has large positive values inside convoluted portions of the objects. To include the flux feature, we can modify the unary potential in equation (3.1) to decrease the object cost at regions where the flux is positive and decrease the background cost at regions where the flux is negative. However, the flux is not well localized near the object membrane, but is defined throughout the image. Rather than using the entire flux map, we would like to confine the flux to be around the object and attenuate its magnitude further away. Accordingly, we can modulate the flux at a pixel location by its proximity to the object mark, and we use the RW probability as the proximity measure to modulate the flux.

We incorporate the flux by defining the unary potential as

$$V_p(y_p) = \Pr(RW) \cdot \theta_{\text{flux}}(x_p), \tag{3.8}$$

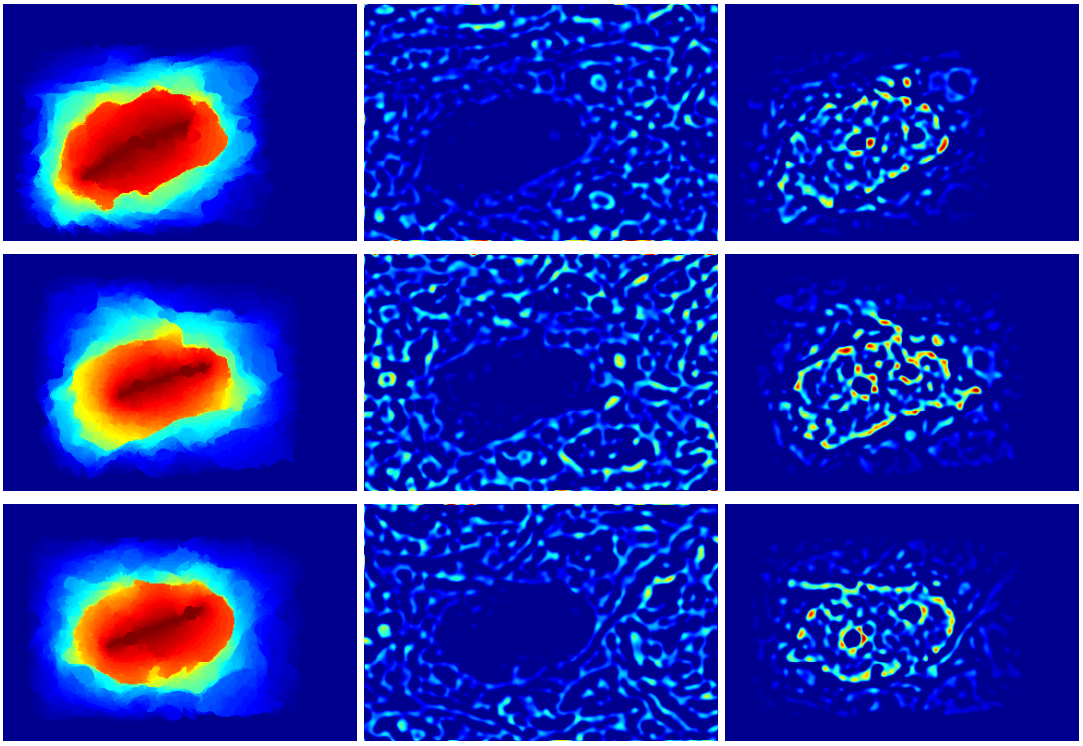


Figure 3.8. (left) Random walk probability, (middle) modulated flux cost $V_p(y_p = 0)$, (right) modulated flux cost $V_p(y_p = 1)$ for the flux maps shown in Figure 3.7.

where

$$\theta_{\text{flux}}(x_p) = \begin{cases} \max[0, -\varphi(p)] & y_p = 0 \\ \max[0, +\varphi(p)] & y_p = 1. \end{cases} \tag{3.9}$$

Here $\Pr(RW)$ is the random walk probability and is computed as before using the algorithm in [43]. Figure 3.8 shows the unary potential (middle and right column) for the flux maps in figure 3.7, which have been modulated by $\Pr(RW)$ shown in the left column. The resulting unary potential based on the modulated flux provides good

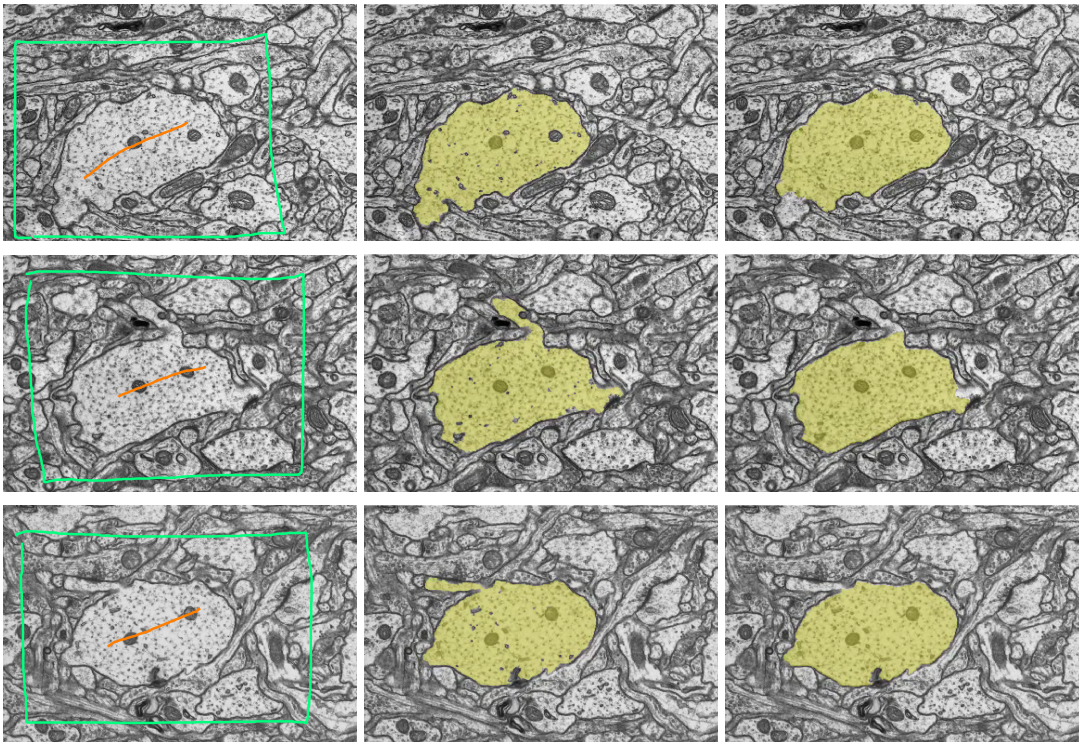


Figure 3.9. (left) input markings, (middle) results using flux unary costs, (right) results with only intensity pairwise costs.

regional separation between the object and membrane and is localized to be only around the object.

We incorporate the unary potential into the graph using the t-weight assignments:

$$w_{sp} = V_p(y_p = 0),\ \forall p \in \mathcal{P} \setminus \mathcal{B}, \tag{3.10a}$$

$$w_{pt} = V_p(y_p = 1),\ \forall p \in \mathcal{P} \setminus \mathcal{O}. \tag{3.10b}$$

This formulation favors a cut in which the object's flux is maximized, and encourages the inclusion of convoluted portions of the object. Notice that for prelabeled pixels in $\{\mathcal{O}, \mathcal{B}\}$, the t-weight assignment is still the one given in equation (3.2).

The right middle column of figure 3.9 shows the results using the new flux unary potential along and the intensity pairwise potential for the user input in the left column. As shown, the segmented objects contain convoluted regions that are missed when using only the pairwise potential (right column). Overall, we find the the addition of the flux feature also decrease the algorithm's sensitivity to the parameter σ_x.

3.3 Segmentation of Serial EM Stack

Building upon our previous work on the 2D segmentation, in this section we propose an algorithm to perform the 2.5D segmentation of the entire serial EM stack. Since the data is not truly 3D, we refer to this problem as 2.5D segmentation. We propose a graph cut segmentation method that mimics the manual 2.5D reconstruction process. Instead of performing graph cuts on the entire image stack, the segmentation is performed on a single image at a time. However unlike our 2D segmentation framework before, we use the result from the adjacent slice as a prior to inform and constrain the segmentation of the current slice. The proposed 2.5D segmentation framework is able to cope with large physical changes that occur between adjacent images. Additionally, the 2.5D method is also much more computationally efficient than the fully 3D method and requires less memory.

3.3.1 Related Works

There are segmentation methods based on both variational level sets and combinatorial graph cuts that are used for segmenting 3D data [118, 11]. Most of these methods were developed for segmenting medical datasets obtained from CT and MRI, where the data can be considered to be 3D in that the z-axis resolution is comparable to the x,y-axes resolutions. For the case of physical serial sections where the z-axis resolution is typically at least an order of magnitude less than the x,y-axes resolutions, application of these methods is not straightforward and presents several challenges. Often there is high variability in image quality across a serial image stack, i.e. uneven illumination and contrast, so that Riemannian metrics used by both level sets and graph cuts cannot be accurately computed across adjacent image slices. Furthermore, the finite difference schemes used for gradient descent in level sets must accommodate the larger z-axis spacing. To make matters worse, the z-spacing is typically not constant across image pairs or throughout the image stack. For graph cut methods, defining the graph structure and the associated edge weights connecting pixels between image pairs are also difficult due to intensity and z-spacing variations between the image pairs.

Conceptually similar to our method, Carlbom *et al*. [15] propose to propagate the segmentation result, *i.e*. the converged snake contour, from the previous image to the current image in the stack. This snake solution is used to initialize the current snake evolution. However, the large physical changes in the object often force the user to

manually adjust the snake to avoid local minima. Moreover, the topological changes in the snake contour are also handled manually. For region splitting, the user must select two points along the contour to bifurcate the snake, and for merging two snakes, the user must select a point on each snake to break the closed contours and join them.

To alleviate the manual correction and topology bookkeeping, Jiang and Tomasi propose the Level-Set Curve Particles method [53]. The contour propagation from one image to another is posed as a tracking problem. A particle filter is used to represent a distribution in the space of all planar curves, and each curve particle is a contour embedded onto a level set function. Although the authors manage to use few particles, the method remains computationally expensive. Moreover, the results presented in their work are based on image data that have more discriminative regional intensities between the object and background, and it is not clear how the performance would change for the EM images.

Instead of propagating the previous solution to the adjacent image, Riklin-Raviv *et al*. [89] propose to segment two images in an iterative manner by exploiting the mutual shape information contained in the two evolving contours. In their framework, the shape of one curve is used as a prior for the other curve and both priors are updated dynamically as the two curves evolve. Their algorithm can successfully segment objects related by a projective transformation. However, the mutual shape prior idea is not straightforwardly applicable here since the shapes between two cross sections can be

drastically different, and correctly controlling the strength of the shape prior for the entire serial stack becomes challenging.

3.3.2 3D Neighborhood Graph

Before describing our 2.5D segmentation approach, we first attempt to segment the serial stack using a full 3D neighborhood graph, similar to the approach in [11]. Instead of the 8-connected neighbors, we use a 26-connectivity system and use the pairwise potential in equation (3.3) to construct the neighborhood edge weights. Since the z-axis spacing is sparse, the distance between two adjacent images are set to be 10 to 50 times the x,y-axis spacing. The factor $1/|p-q|$ in equation (3.4) will decrease the interslice costs as the z-distance increases. We also use a range $\sigma_x = \{5, 10, 20\}$.

For a stack of approximate 11 images (the large graph size limits the number of slices we can use), we select the middle image for user input and assign the t-weights for the marked pixels as in equation (3.2). The remaining t-weights are set to be zero, which is similar to the 2D case without the flux unary cost. We did not use the modulated flux for the unary potential for two reasons: 1) our system (MATLAB, 2GB of RAM) encounters a memory shortage error while running the RW algorithm on the 3D graph even for 3 slices, and 2) the user can inadvertently place label marks that are too close or far from the object in an adjacent slice, which cause the RW to be inaccurate since it is highly dependent on the input.

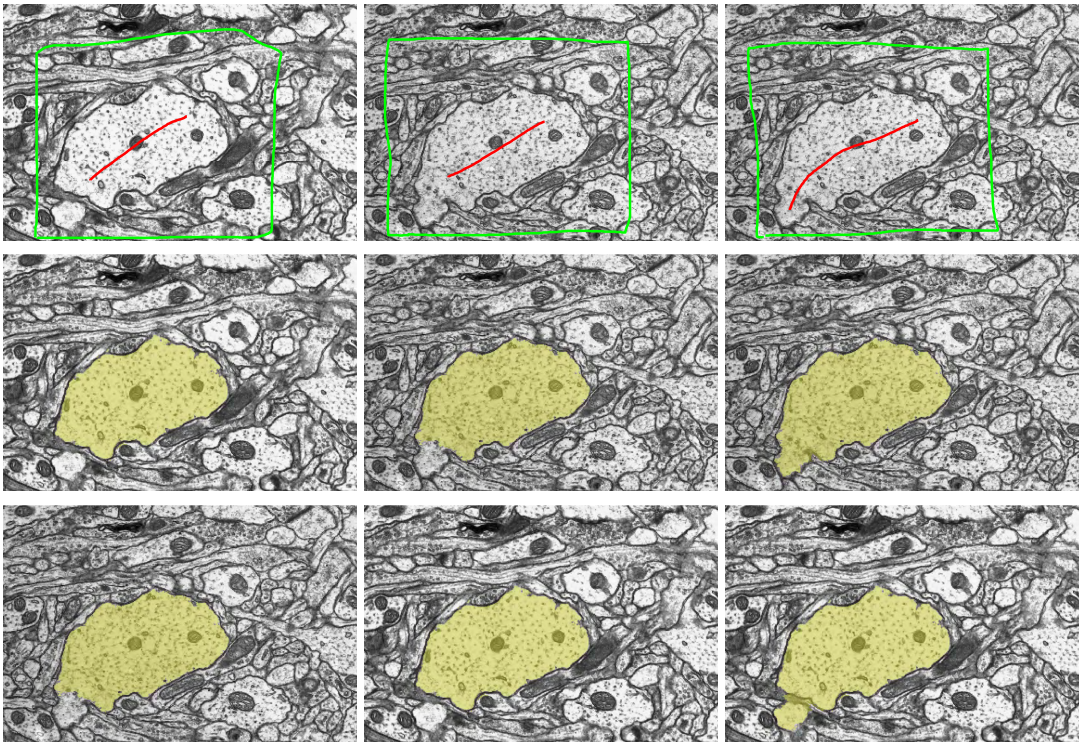


Figure 3.10. Segmentation using 3D neighborhood graph. (top) Input markings for one slice and the results for that slice (middle) and the adjacent slice (bottom) are shown. The left example shows a situation where the user failed to label a prominent branch that should be considered part of the main object. The middle and right examples show the sensitivity of the segmentation to the position of the input marks.

Some results for the 3D segmentation are shown in figure 3.10. The top row shows some examples of user inputs. The middle row shows the results for the image with the input, and the bottom row shows the results for the image that is adjacent to the one with the input. The left column illustrates a situation where the user inadvertently failed to label a prominent branch that should be considered part of the main object. Since that branch is merged to the main object in the adjacent image, it should have been segmented in the input image. However, that portion of the adjacent image is rather

noisy and has a tendency to be labeled as background (see for example figure 3.9). The combined effects of the poor input marking and the noisy adjacent image result in the segmentation error for both images. The middle and right examples show the sensitivity of the segmentation to the position of the input marks. The example in the middle column suffers from the same problems as that in the left column. The right column example shows the problem when the user becomes too specific in the markings so as to ensure the correct segmentation for the input image. Despite running the algorithm for the full parameter space, the final results alternates between the one shown in the middle or right column.

3.3.3 Label Propagation

Instead of using the 3D neighborhood graph constructed for the entire serial stack, we would like to segment one image at a time and propagate the result to the next image. In this way, the previous segmentation acts as a geometric prior that helps to inform the segmentation of the current layer. To incorporate the prior segmentation, we use the higher order clique (HOC) Potts potential proposed by Kohli *et al*. [58, 59]. The HOC potential has been shown to be very beneficial in capturing the image features of large pixel patches, where the unary and pairwise potentials proved insufficient. For a more in depth discussion of the HOC Potts potential and its graph representation, see chapter 2.

Let $\mathbf{y}^{(i-1)}$ be the segmentation or labeling computed for image $\mathbf{x}^{(i-1)}$ in the stack. Define the CRF energy of a labeling $\mathbf{y}^i = \{y_p^i : p \in \mathcal{P}\}$ conditioned on the previous segmentation as

$$E\left(\mathbf{y}^i|\mathbf{y}^{(i-1)}\right) = \sum_{p\in\mathcal{P}, q\in\mathcal{N}_p} V_{pq}(y_p^i, y_q^i) + \sum_{c\in\mathcal{C}} V_c\left(\mathbf{y}_c^i|\mathbf{y}^{(i-1)}\right). \tag{3.11}$$

The first term is defined in equation (3.3). The second term sums the potential of the HOC $c \in \mathcal{C}$, where $\mathcal{C}$ is the set of HOCs for image i. Note again that the CRF energy is dependent on the observed image $\mathbf{x}^i = \{x_p^i : p \in \mathcal{P}\}$, but for simplicity it is not explicitly expressed.

Let's assume that the set of HOCs $\mathcal{C}$ is available for the image $\mathbf{x}^i$. The HOC potential V_c should be designed to capture both the prior label $\mathbf{y}^{(i-1)}$ and the current image information $\mathbf{x}_c^i$ for that clique. We propose to use the following clique potential:

$$V_c\left(\mathbf{y}^i|\mathbf{y}^{(i-1)}\right) = \frac{1}{\sigma_c}\Big(1 + \lambda_c \cdot \theta\left(\mathbf{y}_c^i|\mathbf{y}^{(i-1)}\right)\Big), \tag{3.12}$$

where σ_c is the standard deviation of the clique intensity $\mathbf{x}_c^i$ and the parameter λ_c is set to 500 for all experiments. The fraction $1/\sigma_c$ is a simplistic way to measure the quality of a clique. According to equation (3.12), a clique that has a large σ_c indicates that it is not homogeneous and most likely contains pixels both in the object and on the membrane boundary. Such a clique should have a mixed labeling of 0's and 1's, and the cost of this labeling should remain small. For a more homogeneous clique, σ_c is small and assigning a mixed labeling to this clique is discouraged since the cost is higher.

The prior dependent term $\theta\left(\mathbf{y}_c^i|\mathbf{y}^{(i-1)}\right)$ is given by

$$\theta\left(\mathbf{y}_c^i|\mathbf{y}^{(i-1)}\right) = \begin{cases} 1 - n_c/|c| & y_p^i = 0, \forall p \in c \\ n_c/|c| & y_p^i = 1, \forall p \in c \\ 1 & \text{otherwise,} \end{cases} \tag{3.13}$$

where $|c|$ is the number of pixels in clique c and n_c is the number of pixels in c that overlaps with the prior object in $\mathbf{y}^{(i-1)}$. More specifically, $n_c = \sum_{p \in c} \delta(y_p^{(i-1)} = 0)$ with $\delta(a) = 1$ if condition a is true and $\delta(a) = 0$ otherwise. The prior dependent term $\theta\left(\mathbf{y}_c^i|\mathbf{y}^{(i-1)}\right)$ adjusts the clique cost according to the number of pixels in the clique that overlaps with the prior object. If all the clique pixels are labeled $\mathbf{y}_c^i = 0$, *i.e.* $\{y_p^i = 0 : p \in c\}$ (object), then a cost $1 - n_c/|c|$ is incurred. Instead if all the clique pixels are labeled $\mathbf{y}_c^i = 1$ (background), then a cost $n_c/|c|$ is incurred. However there are situations where the clique has mixed object/background labelings. In this case, a maximum cost of 1 is incurred. Thus when the clique completely overlaps with the prior object, the cost of labeling $\mathbf{y}_c^i = 0$ would be zero, which encourages the clique to take on this labeling. As the fraction of overlap decreases, labeling the clique as $\mathbf{y}_c^i = 0$ becomes less favorable.

The graph construction for the HOC Potts potential is described in chapter 2. For the potential in equation (3.12), we have the following weight assignments:

$$w_{sc} = V_c\left(\mathbf{y}_c^i = 0|\mathbf{y}^{(i-1)}\right),\ \forall c \in \mathcal{C} \tag{3.14a}$$

$$w_{ct} = V_c\left(\mathbf{y}_c^i = 1|\mathbf{y}^{(i-1)}\right),\ \forall c \in \mathcal{C}. \tag{3.14b}$$

The n-edge assignment is given in equation (3.6) and 8-connectivity is used. Note that the CRF energy in equation (3.11) does not contain a unary potential term and thus the t-edges (s, p) and (p, t) have zero weight.

3.3.4 Results

The HOCs are simply small patches or pixel groups in the image and can be computed using any algorithm that produces an oversegmentation of the image, such as watershed. In this work, we use the Mean Shift (MS) algorithm of Comaniciu and Meer [22] to compute the oversegmentation, as was similarly done in [59]. The MS algorithm has two parameters that we vary, the spatial range h_s and the feature space range h_r. Since the HOC graph does not limit the cliques from overlapping, we could use several oversegmented images to create the clique set. In fact, we found that using two clique maps generated by MS with $(h_r, h_s) = \{(10, 6), (10, 10)\}$ works well for our data. These settings sufficiently capture the structures of the objects at the right scale. Figure 3.11 shows an image with the two clique maps generated from MS.

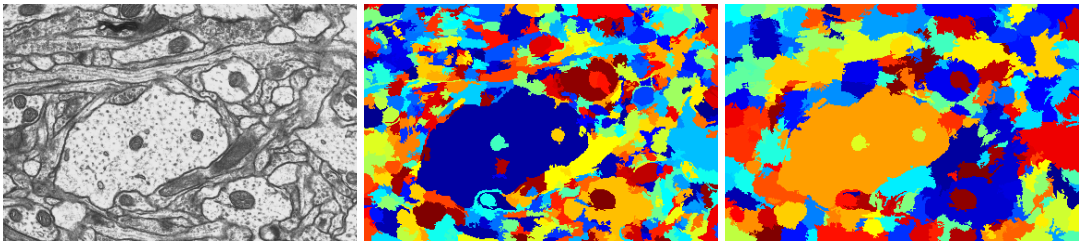


Figure 3.11. Original image and two higher order clique maps computed using Mean-Shift with settings $(h_r, h_s) = \{(10, 6), (10, 10)\}$. The clique color is randomly chosen for visualization.

The segmentation begins with the user placing input marks on either the first or last image in the stack. This image is segmented using graph cuts with flux, as described above. Then the resulting labeling is propagated to the next image via the HOC potential in equation (3.12) and a new labeling is computed. This process continues iteratively downward and then upward through the image stack until no more changes are detected. Figure 3.12 illustrates the process for two adjacent images. The label in the top left image is used as the prior for the image in the top right. The middle row shows the resulting clique potentials for $V_c\left(\mathbf{y}_c^i = 0|\mathbf{y}^{(i-1)}\right)$ (middle left) and $V_c\left(\mathbf{y}_c^i = 1|\mathbf{y}^{(i-1)}\right)$ (middle right). The bottom right image shows the clique quality measure $1/\sigma_c$ used to weigh the prior clique costs. Notice how the HOC potential manages to capture the label prior, but very strong values (red) where the previous object label overlap. The final result after label propagation is shown in the bottom right.

The 2.5D label propagation algorithm converges quickly (one or two iterations) and is very efficient for several reasons. First the clique maps for all the serial images along

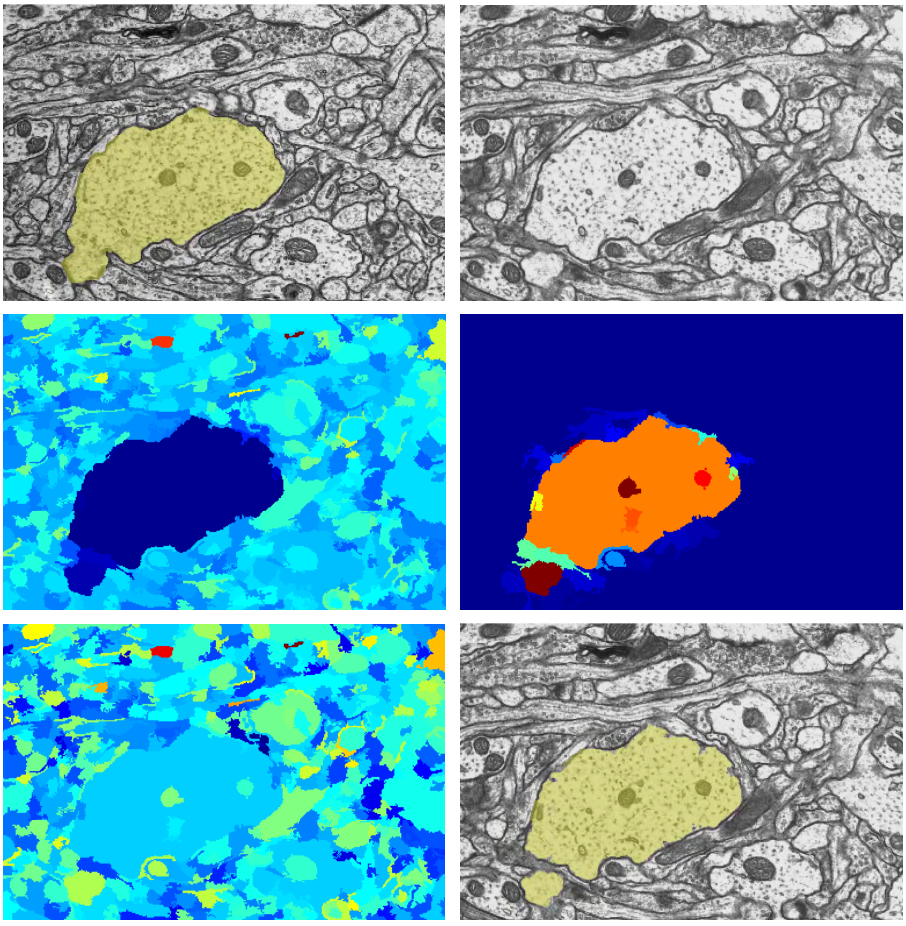


Figure 3.12. Example of the clique costs. (top left) Label prior that we wish to propagate to the adjacent image (top right). The resulting clique potentials for $V_c\left(\mathbf{y}_c^i = 0|\mathbf{y}^{(i-1)}\right)$ (middle left) and $V_c\left(\mathbf{y}_c^i = 1|\mathbf{y}^{(i-1)}\right)$ (middle right) show the dependency on the prior. (bottom left) The clique quality measure $1/\sigma_c$ and the final result for the top right image (bottom left).

with their standard deviation σ_c's are computed offline. Second, the same neighborhood graph structure is used throughout the process for all the images, and the n-edge weights and clique structure remain the same for each image. The only updates required are for the clique weights to reflect changes in the label being propagated. However, the label

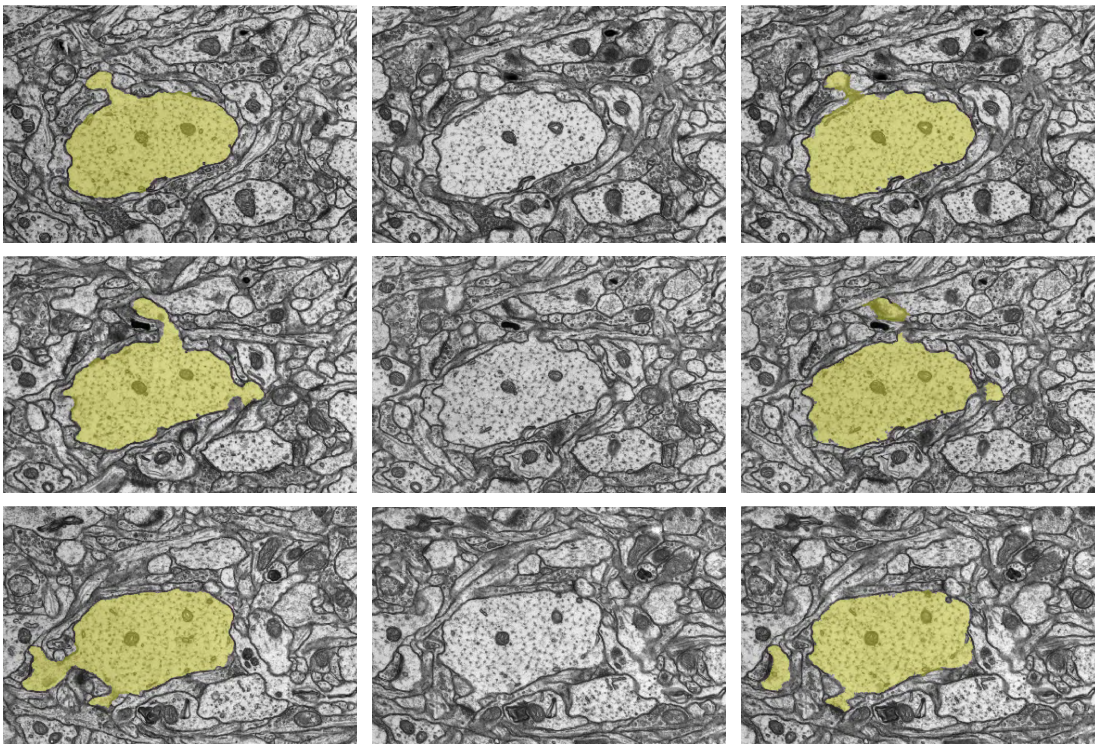
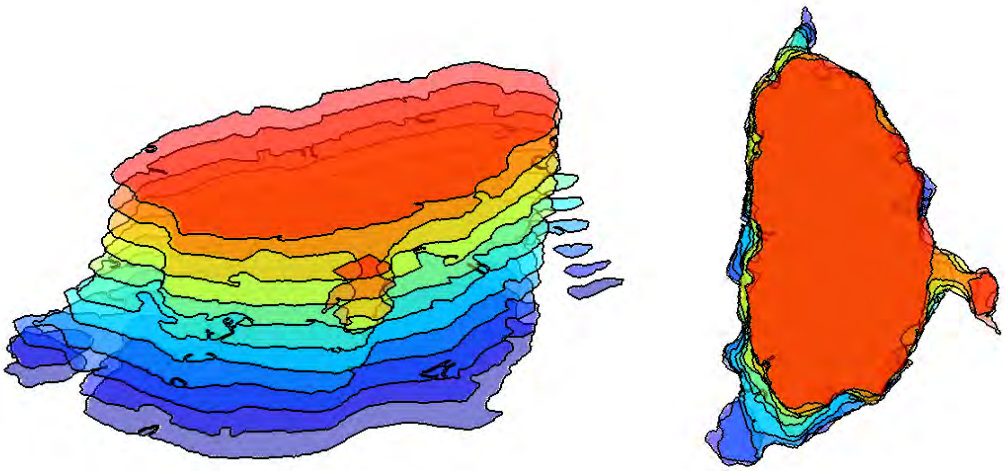


Figure 3.13. Label propagation results. (left) Label prior, (middle) adjacent image, (right) propagation result. These adjacent images are chosen to demonstrate the algorithm's ability to correctly segment branching regions in the object that are considerably separated.

prior term $\theta\left(\mathbf{y}_c^i|\mathbf{y}^{(i-1)}\right)$ requires few binary comparisons and can be computed very quickly. Unlike the method using a full 3D neighborhood graph, our 2.5D method runs graph cut on a much smaller 2D neighborhood graph for each image and consequently can accommodate larger image stacks and image sizes.

Figure 3.13 shows several more results for adjacent image pairs in the same stack. The label priors are shown in the left column, and the adjacent images for label propagation are shown in the middle column. The result of label propagation is shown in the



(a) side view (b) top view

Figure 3.14. The result of 2.5D label propagation for 11 images in a serial stack. Note the small branchings that the method is able to segment.

right column. Notice that these image pairs contain large merging and splitting events can can easily be missed if segmentation is done on each image individually. Finally the result for 11 images in the serial stack is shown in figure 3.14. As seen in the figure, the 2.5D label propagation method can correctly capture the small branching processes belonging to the object.

3.4 Conclusion

In this chapter, we proposed a set of algorithm for segmentation of serial Em images. The 2D segmentation problem is solved by minimizing a CRF energy composed of a

unary potential term based on the flux of the intensity gradient field and a pairwise potential term based on the intensity differences between two neighboring pixels. Using graph cuts, the CRF energy is minimized exactly. The combination of flux and intensity proved to be good features to provide contrast between the thin, convoluted membrane and the object. Then we proposed a 2.5D approach to segment the entire serial image stack. Starting with an initial user input for the first image in the stack, the algorithm propagates the labeling computed for that image to the adjacent image. The higher order clique potential is used to incorporate the the prior label into the current image. Our results show that the proposed 2.5D algorithm can cope with large physical changes to the object topology, such as branching, between adjacent images.

There are several steps that can be taken to improve the proposed algorithm. First, the results are assessed in a qualitative manner, and our experiments lack the ground truth data necessary for definitive comparisons with other algorithms. Second, our dataset obtained from the SynapseWeb [104] is of relatively very high quality, and it would be useful to test our algorithm on more typical serial EM datasets. However, obtaining data from other researchers remains a challenge in itself. Finally as mentioned earlier, we have not addressed is the problem of aligning the images in the stack; the serial stacks in our dataset have already been manually aligned. It would be beneficial and is of great research interest to perform the segmentation and alignment simultane-

ously, such as the work of Yezzi *et al*. [123] on joint segmentation and registration in the variational framework.

Chapter 4

Segmentation Using Shape Prior

Segmentation methods based solely on image information [16, 18] often perform poorly in the presence of noise, background clutter, and object occlusions. The addition of shape prior information has shown to significantly improve segmentation results and is popular among continuous approaches [17, 25, 71, 90, 93, 111]. Recently, there has been an increased interest in graph based segmentation algorithms [11, 7, 10], and subsequently the addition of prior shape information into their formulations. However, many continuous shape distances or dissimilarity measures can be difficult, if not impossible, to formulate as discrete energies for graph optimization. This is especially true for graph cut methods.

In this work, we present a new shape prior segmentation method using combinatorial graph cuts. First, we define the shape prior energy using a discrete version of the

shape distance proposed by Chan and Zhu [17] for the level sets framework, and incorporate this energy into the graph via terminal edge weights. Unlike those of previous graph based approaches, this shape distance is both symmetrical and obeys the triangle inequality. Second, to simultaneously segment multiple objects, we propose a multiphase graph cut approach to handle object overlap, where a pixel can have multiple object memberships (labels). The multiphase formulation differs from multiway cuts in that the former can account for object overlaps by allowing a pixel to have multiple labels. The multiway cut algorithms such as α-expansion and $\alpha\beta$-swap [10] compute solutions where a pixel is assigned only one label. We then extend our shape prior energy to incorporate multiple shape priors.

A major advantage of our algorithm is that the segmentation energy is minimized directly with graph cuts, unlike variational methods which require the energy gradient for minimization. Computation of the gradient for many energy functionals can be difficult because these energies are often non-differentiable and require approximations [25]. To make the algorithm invariant to affine transformations of the shape, we use the theory of moment invariance of binary shapes [86] for alignment, allowing direct computation of the transformation parameters without using gradient descent estimation for each parameter. Experiments demonstrate that our algorithm can cope with image noise and clutter, as well as partial occlusions and affine transformations of the shape. A preliminary version of this work was presented in [116].

In section 4.1, we review several notable graph based shape prior segmentation algorithms. Section 4.3 describes the shape prior model, and section 4.4 provides detail on using this energy in the multiphase graph cut framework for the segmentation of multiple objects. Section 4.5 extends the shape prior model to incorporate multiple prior shapes. The results of our algorithm are shown in section 4.6, followed by a brief discussion in section 4.7.

4.1 Related Works

The graph methods of Felzenszwalb [30] and Schoenemann and Cremers [98] can segment objects under elastic deformations without needing any initialization and guarantee globally optimal solutions. In [30], nonserial dynamic programming is used to find the optimal matching between a deformable template represented by triangulated polygons and the image pixels. In [98], the segmentation is found by computing the minimal ratio cycle in a product graph of the image and a shape template parameterized by arc length. Both of these methods can be slow in practice, with runtimes of up to several minutes on typical CPUs. Moreover, the triangulated polygon representations and arc length parameterizations limit the topological flexibility of the template shapes and may not easily extend to the 3D case.

There are several algorithms that employ graph cuts for shape prior segmentation. Freedman and Zhang [39] use the shape's distance transform φ to define the edge weight between neighboring pixels p and q as $\varphi((p+q)/2)$. Kolmogorov and Boykov [63] assign neighborhood edge weights to favor cuts that maximize the flux of the distance map gradient. Both of these methods largely rely on user markings to estimate the template pose. Kumar *et al*. [67] also utilize the shape's (signed) distance map, but estimate the pose using shape and appearance models constructed during training. Most closely related to our work, Malcolm *et al*. [76] impose the shape prior model on the terminal edges and perform graph cuts iteratively starting with an initial contour. Given a set of training shapes, their method constructs a statistical shape space using kernel principle component analysis (kPCA). At each iteration, the pre-image of the previous labeling in this shape space is used as the prior probability map, and the negative log of this pre-image is assigned to the terminal weights. While these methods produce promising results, their shape energies are not based on shape metrics, *e.g*. they are unsymmetrical. Furthermore, these methods do not handle affine transformations of the shapes and cannot segment multiple objects simultaneously.

4.2 Segmentation Energy

Using the Conditional Random Fields (CRF) model, the shape prior segmentation problem is formulated as an energy minimization solved using graph cuts. Let ψ^0 denote the shape prior. Then a given image $\mathbf{x}$ with pixel set $\mathcal{P}$ and a set of labels $\mathcal{L}$, the goal is to find a labeling $\mathbf{y} : \mathcal{P} \rightarrow \mathcal{L}$ that minimizes the second order energy

$$E(\mathbf{y}|\psi^0) = \sum_{p\in\mathcal{P}} V_p(y_p|\psi^0) + \sum_{p\in\mathcal{P},q\in\mathcal{N}_p} V_{pq}(y_p, y_q), \tag{4.1}$$

where $\mathcal{N}_p$ is the set of pixels in the neighborhood of p. Here $V_p(y_p|\psi^0)$ is the penalty of assigning label $y_p \in \mathcal{L}$ to p given the prior shape, and $V_{pq}(y_p, y_q)$ is the penalty of labeling the pair p and q with labels $y_p, y_q \in \mathcal{L}$, respectively. In equation (4.1) as well as for the remainder of this chapter, for convenience our notation do not explicitly indicate the dependency of the CRF energy and the clique potentials on the observation $\mathbf{x}$, but it is assumed that this dependency exists.

For the segmentation problem in this chapter, we are use binary labels $\mathcal{L} = \{0, 1\}$ and the pairwise penalty [8]

$$V_{pq}(y_p, y_q) = g(p, q) \cdot |y_p - y_q|, \tag{4.2}$$

where

$$g(p, q) = \lambda_x \exp\left(-\frac{(x_p - x_q)^2}{2\sigma_x^2}\right) \frac{1}{|p - q|}. \tag{4.3}$$

Here x_p is the intensity value at pixel p, $|p - q|$ is the Euclidian distance between pixels p and q. The parameter σ_x can be considered an estimate of camera noise, and

λ_x weights the importance of the pairwise energy. Accordingly, a penalty $g(p,q)$ is incurred only when neighboring pixels have different labels, and thus V_{pq} encourages region coherence of the labels. Note that pairwise potential V_{pq} is a submodular function, *i.e.* $V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(0,1) + V_{pq}(1,0)$ [65], and thus the minimum of $E(\mathbf{y}|\psi^0)$ can be computed efficiently using graph cuts. For convenience, the second term in equation (4.1) is denoted $E_{pq}(\mathbf{y})$.

The unary potential term is a sum of a data dependent potential and a shape prior potential and is given by

$$V_p(y_p|\psi^0) = V_D(y_p) + V_S(y_p|\psi^0). \tag{4.4}$$

The data dependent term is defined based on the image intensity, such as the log likelihood of the image model, while the shape prior term is independent of image information and only depends on the prior shape ψ^0. These terms will be described in subsequent sections.

The graph construction for the st-mincut is described in details in chapter 2, but we remind the reader of our notations. Neighboring nodes p and q are connected by n-edge (p,q) with n-weight w_{pq}. Furthermore, p is connected to terminals s and t via t-edges (s,p) and (p,t) with corresponding t-weights w_{sp} and w_{pt}, respectively. In our graph, if $(p,q) \in \mathcal{E}$ then $(q,p) \in \mathcal{E}$, the n-weights $w_{pq} = w_{qp}$, and all pixels $p \in \mathcal{P}$ are connected to both s and t. Finally, the desired graph with cut cost equaling $E(\mathbf{y}|\psi^0)$ is constructed

using the following edge weight assignments [7]:

$$w_{pq} = w_{qp} = g(p, q), \tag{4.5a}$$

$$w_{sp} = V_p(y_p = 0), \tag{4.5b}$$

$$w_{pt} = V_p(y_p = 1). \tag{4.5c}$$

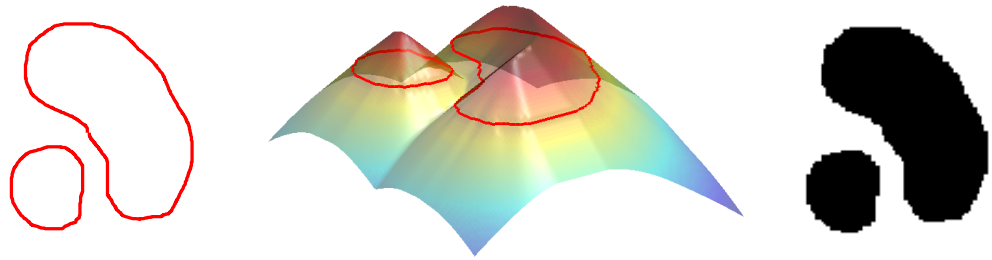
In our notation, a pixel p is assigned label $y_p = 0$ (object) if $p \in \mathcal{T}$ and $y_p = 1$ (background) if $p \in \mathcal{S}$, where $\mathcal{S}$ and $\mathcal{T}$ are the two disjoint node sets separated by the cut (see chapter 2 for more details).

4.3 Shape prior model

In this section, we describe the shape prior model and show how to define the shape penalty $V_S(y_p)$ such that a cut on the graph, with V_{pq} and V_D defined, has an added cost corresponding to the shape prior energy.

4.3.1 Shape distance

In the variational level set framework, a shape is usually represented using an embedding of the shape onto the zero level contour of the level set function. Figure 4.1 shows an example of the embedding. Given two shapes embedded onto the zero level of level set functions ϕ^a and ϕ^b on the image plane $\Omega \subset \mathbb{R}^2$, Zhu and Chan define their



(a) shape (b) level set embedding ϕ (c) binary ψ

Figure 4.1. Shape representations. The shape in (a) can be represented using level set embedding (b) or a binary representation (c). These nonparametric representations allow the shape to have arbitrary topology.

distance as [17]

$$d^2(\phi^a, \phi^b) = \int_\Omega \left(H(\phi^a(x)) - H(\phi^b(x))\right)^2 dx, \tag{4.6}$$

where $H(\cdot)$ is the Heaviside function. Many level set segmentation methods [25, 90] use this distance as the shape prior energy due to its many attractive properties: it is positive, symmetric, obeys the triangle inequality, and does not depend on the size of the domain Ω.

Since $H(\phi^i)$ effectively binarizes the shape embedding function ϕ^i, for notation simplicity we will replace $H(\phi^i)$ with ψ^i. On the discrete pixel domain, equation (4.6) can be expressed in terms of ψ^a and ψ^b as

$$\begin{aligned} d^2(\psi^a, \psi^b) &= \sum_{p \in \mathcal{P}} (\psi_p^a - \psi_p^b)^2 \\ &= \sum_{p \in \mathcal{P}} \left(\psi_p^a \bar{\psi}_p^b + \bar{\psi}_p^a \psi_p^b\right), \end{aligned} \tag{4.7}$$

where ψ_p^i is the binary value of ψ^i at pixel p and $\bar{\psi}_p^i = 1 - \psi_p^i$. The expansion in equation (4.7) is possible because both ψ^a and ψ^b are binary functions. Note that the binary representation does not restrict the shape to be a single closed contour but allows it to have arbitrary topology (holes and multiple unconnected parts). See figure 4.5 for examples of shape templates used in this work.

4.3.2 Shape penalty

In order to use the shape distance in equation (4.7) for graph cuts, we must define the shape penalty $V_S(y_p|\psi^0)$ such that, for a given prior shape template ψ^0, a cut with binary labeling $\mathbf{y}$ has an added cost equal to $d^2(\mathbf{y}, \psi^0)$. Using equation (4.7), we define the energy of a binary labeling $\mathbf{y}$ given a prior template ψ^0 as

$$E_S(\mathbf{y}|\psi^0) = d^2(\mathbf{y}, \psi^0). \tag{4.8}$$

Then the shape prior penalty is

$$V_S(y_p|\psi^0) = y_p\bar{\psi}_p^0 + \bar{y}_p\psi_p^0. \tag{4.9}$$

It follows that if p is assigned label $y_p = 0$ but $\psi_p^0 = 1$, then the t-edge (s, p) is in the cut and a penalty of 1 is added to the cost. The same is true when $y_p = 1$ but $\psi_p^0 = 0$. However, when $y_p = \psi_p^0$, no penalty is incurred. Thus, the cut which results in a labeling $\mathbf{y}$ that minimizes $d^2(\mathbf{y}, \psi^0)$ gives the minimum shape prior energy.

4.3.3 Affine invariant shape alignment

To make the shape distance in equation (4.7) invariant to geometric transformations, ψ^a and ψ^b must be properly aligned. Since ψ^a and ψ^b are effectively binary images, we use the image normalization work of Pei and Lin [86] to align these shapes. The normalization process transforms a shape to an affine invariant shape space using transformations computed intrinsically from the shape's moments (up to 3rd order). In our notation, the affine transformed shape $\hat{\psi}$ is related to the normalized shape ψ via a transformation $\hat{\psi} = T(\psi)$. We refer the reader to [86] for more details on the computation of the transformation. We would like to note that affine invariant shape alignment using Legendre moments has also been successfully applied in the level set domain[36]. However the alignment is performed by gradient descent on a shape distance composed of up to 40 higher order moments, which can computationally inefficient.

For the segmentation, assume that the prior template ψ^0 has been normalized. Given an estimate $\hat{\mathbf{y}}$ for the target labeling indicating the object (described later in section 4.4.2), $\hat{\mathbf{y}}$ is normalized by computing the transformation $T(\hat{\mathbf{y}})$. Then the template ψ^0 is aligned to $\hat{\mathbf{y}}$ by reversing the normalization procedure on ψ^0 using the transformation computed for $\hat{\mathbf{y}}$, *i.e.* $\hat{\psi^0} = T^{-1}(\psi^0)$. Finally to make the distance scale invariant, $d^2(\hat{\mathbf{y}}, \hat{\psi^0})$ is divided by $\sqrt{\lambda_1 \lambda_2}$, where λ_1 and λ_2 are the eigenvalues of the covariance matrix of $\hat{\mathbf{y}}$. For the remainder of this paper, the notation $d^2(\mathbf{y}, \psi^0)$ is assumed to be the invariant shape distance between $\mathbf{y}$ and ψ^0.

In general, alignment by intrinsic normalization does not necessarily result in the minimum distance, especially when $\hat{\mathbf{y}}$ contains many spurious or noisy parts. However, the iterative segmentation procedure described in section 4.4.2 allows us to use this alignment scheme even when the initial estimate the object, *i.e.* $\hat{\mathbf{y}}$ is very different from the template. More robust object pose estimation schemes, such as that in [67], can also be utilized but were not necessary in our experiments.

4.4 Shape prior segmentation

In this section, we present a multiphase graph cut method capable of segmenting multiple, possibly overlapping objects. For each object, the segmentation is posed as a binary labeling problem, with the shape prior penalty imposed independently for that labeling. However the data penalty is dependent on both the image intensity and the other labelings. We will denote the jth object labeling by $\mathbf{y}^j$ and its value at pixel p by y_p^j. Note that y_p^j is binary and should not be confused as y_p raised to the power j. The total segmentation energy can be expressed as the sum of energies over all the labelings, *i.e.*

$$E(\mathbf{y}|\psi^0) = E_D(\mathbf{y}) + \sum_{j=1}^{M} \left(E_{pq}(\mathbf{y}^j) + \lambda_s E_S(\mathbf{y}^j|\psi^0) \right), \tag{4.10}$$

where $\mathbf{y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$ is the set of M object labelings, and $E_D(\mathbf{y})$ is the sum of the unary data penalties of all labelings. The parameter λ_s controls the strength of the

shape penalty. We will show that equation (4.10) can be minimized iteratively by performing M mincuts on a single graph at each iteration, with only t-weight updates. The proposed multiphase graph cuts can be considered a discrete version of the multiphase level set framework of Vese and Chan [113].

4.4.1 Multiphase graph cuts

As mentioned, we will use M labelings to segment M objects in the image. The labelings can partition the image into a maximum of 2^M regions or phases. The indicator function for region k is denoted as χ^k. Assume that there is a data model to describe each image region, *e.g.* $Pr(x_p|\chi^k)$, and that each model has an associated cost θ^k, which could be considered as the log likelihood. Then the data cost for the entire image is the sum of the individual region costs, *i.e.*

$$E_D(\mathbf{y}) = \sum_{p\in\mathcal{P}} \sum_{1\le k\le 2^M} \theta_p^k \chi_p^k, \tag{4.11}$$

where we denote $\theta_p^k = \theta^k(p)$ and $\chi_p^k = \chi^k(p)$. We proceed with an intuitive discussion of how to minimize equation (4.11) with graph cuts by describing the procedure for cases where $M = 1$ and $M = 2$. The cases for $M > 2$ can be solved using a similar reasoning, but will not be described in detail.

For a single object ($M = 1$), the image is divided into object ($y_p^1 = 0$) and background ($y_p^1 = 1$) regions, and the model for the two regions will compete to best

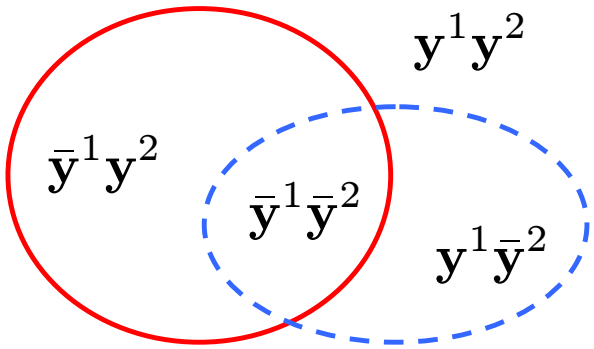


Figure 4.2. Regions for two labelings $\mathbf{y}^1$ and $\mathbf{y}^2$.

estimate the image. Given the object and background costs θ^0 and θ^1, respectively, equation (4.11) becomes

$$E_D(\mathbf{y}^1) = \sum_{p\in\mathcal{P}} \left(\theta_p^0 \bar{y}_p^1 + \theta_p^1 y_p^1\right). \tag{4.12}$$

The data energy can be incorporated into the graph construction via the following t-weight assignments:

$$w_{sp} = V_D(y_p^1 = 0) = \theta_p^0, \tag{4.13a}$$

$$w_{pt} = V_D(y_p^1 = 1) = \theta_p^1. \tag{4.13b}$$

Now for $M = 2$, consider the two labelings $\mathbf{y}^1$ and $\mathbf{y}^2$ shown in figure 4.2. With the overlap, the image is partitioned into four regions $\bar{\mathbf{y}}^1\bar{\mathbf{y}}^2$, $\mathbf{y}^1\bar{\mathbf{y}}^2$, $\bar{\mathbf{y}}^1\mathbf{y}^2$, and $\mathbf{y}^1\mathbf{y}^2$. The data energy for labeling $\mathbf{y} = \{\mathbf{y}^1, \mathbf{y}^2\}$ is

$$E_D(\mathbf{y}) = \sum_{p\in\mathcal{P}} (\theta_p^{00}\bar{y}_p^1\bar{y}_p^2 + \theta_p^{01}\bar{y}_p^1 y_p^2 + \theta_p^{10} y_p^1\bar{y}_p^2 + \theta_p^{11} y_p^1 y_p^2), \tag{4.14}$$

where θ^{00}, θ^{10}, θ^{01}, and θ^{11} are the data costs for regions $\bar{\mathbf{y}}^1\bar{\mathbf{y}}^2$, $\mathbf{y}^1\bar{\mathbf{y}}^2$, $\bar{\mathbf{y}}^1\mathbf{y}^2$, and $\mathbf{y}^1\mathbf{y}^2$, respectively. To properly impose this cost into the graph construction, first consider the labeling $\mathbf{y}^1$. For the regions of $\mathbf{y}^1$ where $y_p^2 = 0$, *i.e.* $\mathbf{y}^1\bar{\mathbf{y}}^2$ and $\bar{\mathbf{y}}^1\bar{\mathbf{y}}^2$, the two models with costs θ^{10} and θ^{00} compete for the best fit. Similarly, the two models with costs θ^{01} and θ^{11} compete to describe the regions of $\mathbf{y}^1$ where $y_p^2 = 1$, *i.e.* $\bar{\mathbf{y}}^1\mathbf{y}^2$ and $\mathbf{y}^1\mathbf{y}^2$. Thus from the perspective of $\mathbf{y}^1$, the labeling problem is similar to the single object case, except that now the object/background costs depend on $\mathbf{y}^2$ as well as the image data. A similar reasoning can be used for $\mathbf{y}^2$.

Equation (4.14) can be minimized iteratively by performing mincuts on a single graph, with each labeling computed alternatingly by updating the t-weights with data penalty assignments

$$V_D(y_p^1 = 0) = \theta_p^{00}\bar{y}_p^2 + \theta_p^{01}y_p^2, \tag{4.15a}$$

$$V_D(y_p^1 = 1) = \theta_p^{10}\bar{y}_p^2 + \theta_p^{11}y_p^2, \tag{4.15b}$$

$$V_D(y_p^2 = 0) = \theta_p^{00}\bar{y}_p^1 + \theta_p^{10}y_p^1, \tag{4.15c}$$

$$V_D(y_p^2 = 1) = \theta_p^{01}\bar{y}_p^1 + \theta_p^{11}y_p^1, \tag{4.15d}$$

where $V_D(y_p^i)$ is the penalty used to compute labeling $\mathbf{y}^i$. The pairwise penalty in equation (4.2) can be used to assign the n-weights.

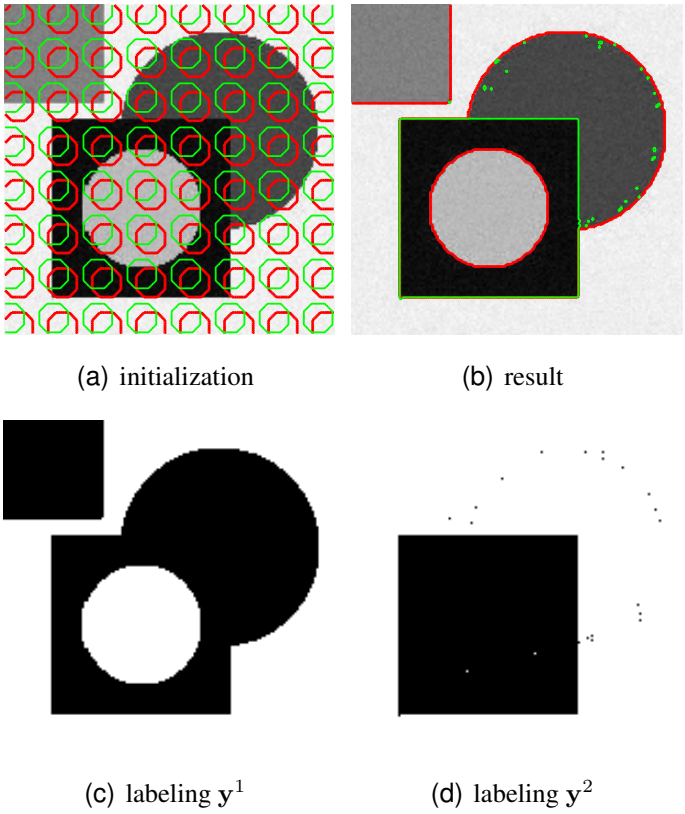
For $M > 2$, the data penalty for each labeling must account for all 2^M possible regions. These penalties are computed by considering the object/background competition

for each region. Note that the graph structure for all labelings is identical, with only the t-weights changing for each iteration. The n-weights remain the same, resulting in less memory storage and faster graph construction. Furthermore, since the mincut algorithm is inherently stable, large moves are possible during each iteration, speeding up convergence. However, like the level sets formulation, the multiphase graph cut framework does not guarantee the globally optimal solution. The multiphase graph cut procedure is summarized in algorithm 4.4.1.

Algorithm 4.4.1 Multiphase graph cuts for M labels.

Compute n-weights of graph $\mathcal{G}$ (same for all M labels).
Initialize labelings $\mathbf{y} \leftarrow \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$.
while $\mathbf{y}$ not converged **do**
  for $j = 1$ to M **do**
    1. Compute costs $\{\theta^k : \ 1 \leq k \leq 2^M\}$.
    2. $w_{sp} \leftarrow V_D(y_p^j = 0)$
    3. $w_{pt} \leftarrow V_D(y_p^j = 1)$
    4. $\mathbf{y}^j \leftarrow$ mincut $\mathcal{G}$
  end for
  $\mathbf{y} \leftarrow \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$
end while

Figure 4.3 shows an example of the multiphase graph cuts with $M = 2$. Each labeling $\mathbf{y}^i$ is initialized by an array of circles, where the red and green color indicates separate labelings. For this example, the algorithm converged after two iterations and took less than half a second. The image size is 150×150 pixels. Using the mulitphase level set method to perform the segmentation would potentially require more iterations and up to several seconds of runtime. The large moves during each iteration allows the



(a) initialization (b) result

(c) labeling $\mathbf{y}^1$ (d) labeling $\mathbf{y}^2$

Figure 4.3. Multiphase graph cuts example. Using two binary labelings $\mathbf{y}^1$ and $\mathbf{y}^2$, the algorithm can segment 4 regions. The labelings are initialized by arrays of red and green circles. The final binary labelings are also shown.

multiphase graph cuts to reach convergence much faster than the level set counterpart, which is limited by the small update step during gradient descent.

4.4.2 Iterative Segmentation with Shape Prior

The shape prior energy $E_S(\mathbf{y}^j|\psi^0)$ is dependent on the geometric transformations of the template ψ^0. However, unless the target object's pose in the image is known, the

shape penalty cannot be defined accurately. To overcome this lack of information, we perform the segmentation in an iterative manner (see algorithm 4.4.2). For the graph $\mathcal{G}$, the n-weights are computed only once and remain the same for all objects. Then given an initial labeling $\mathbf{y}^j$ for object j, the template ψ^0 is aligned to $\mathbf{y}^j$. The shape penalties are computed using equation (4.9) and the data penalties are computed as described in section 4.4.1. These unary penalties are summed and assigned to the t-weights, and the mincut solution for this graph produces a new $\mathbf{y}^j$. This process is repeated for all M objects until convergence is reached.

Algorithm 4.4.2 Segmentation of M objects given ψ^0.

Compute n-weights of graph $\mathcal{G}$ (same for all objects).
Initialize labelings $\mathbf{y} \leftarrow \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$.
while $\mathbf{y}$ not converged **do**
  for $j = 1$ to M **do**
    1. Align ψ^0 to $\mathbf{y}^j$.
    2. t-weights $\leftarrow V_D(y_p^j) + V_S(y_p^j|\psi^0)$
    3. $\mathbf{y}^j \leftarrow$ mincut $\mathcal{G}$
  end for
  $\mathbf{y} \leftarrow \{\mathbf{y}^1, \ldots, \mathbf{y}^M\}$
end while

Ideally the final labeling should be insensitive to initializations since in general, an initialization may not be a good estimate of the final segmentation. To lessen the dependency on initialization, the data penalty should dominate the cost function at the start of segmentation, while the shape penalty should remain small. As the segmentation progresses, the shape penalty then increases and forces the cut to resemble the prior template more closely. The shape penalty can be adaptively controlled by redefining

the shape energy for the ith iteration as

$$E_S(\mathbf{y}^{j,i}|\psi^0) = \alpha(\mathbf{y}^{j,i-1}, \psi^0) \cdot d^2(\mathbf{y}^{j,i}, \psi^0), \tag{4.16}$$

where the weighting function

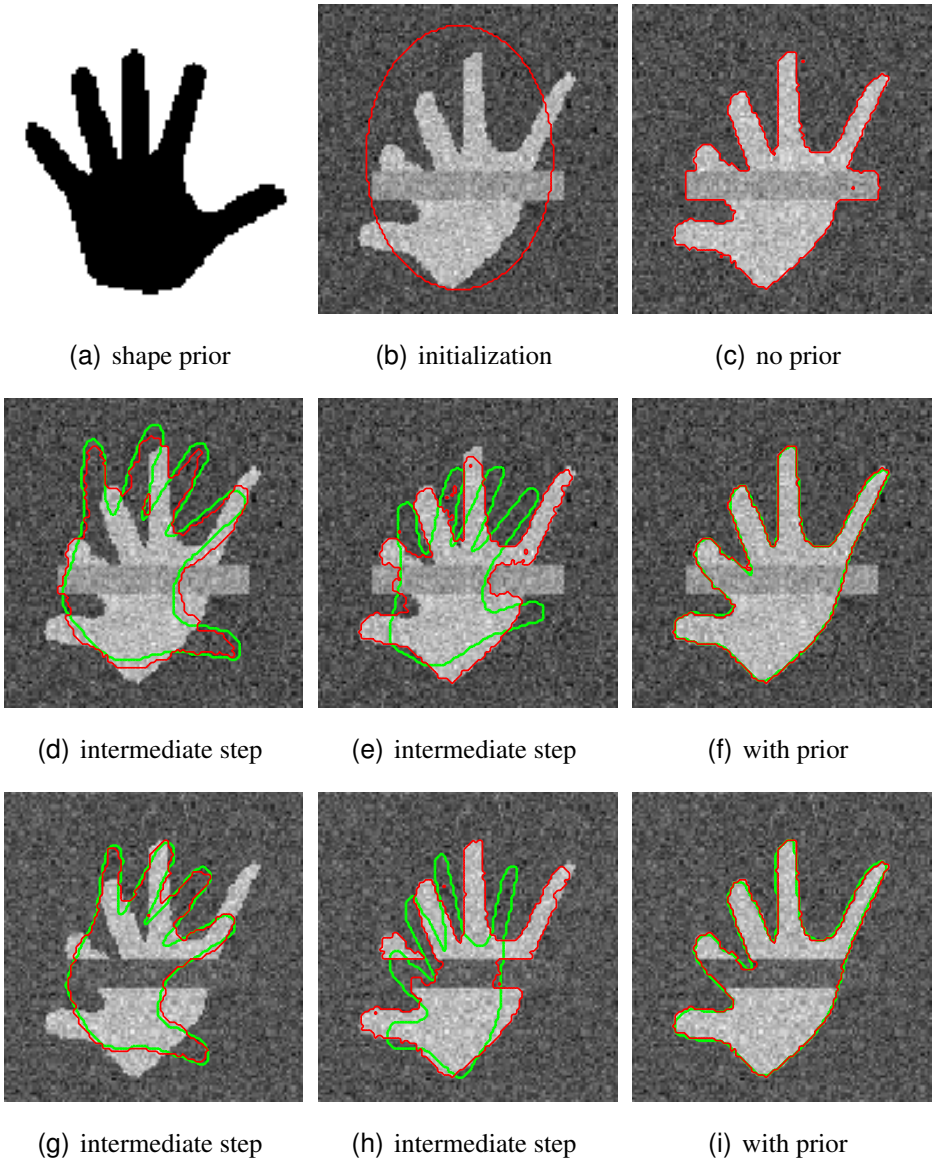
$$\alpha(\mathbf{y}^{j,i-1}, \psi^0) = \exp\left(-\frac{1}{2\sigma_s^2} d^2(\mathbf{y}^{j,i-1}, \psi^0)\right) \tag{4.17}$$

and $\mathbf{y}^{j,i}$ denotes the labeling $\mathbf{y}^j$ at iteration i. Equation (4.17) adaptively weighs the shape energy according to how similar the previous labeling $\mathbf{y}^{j,i-1}$ is to the template ψ^0. The parameter σ_s controls the rate at which the shape energy changes.

We would like to point out that the proposed method do not guarantee a globally optimal solution. However, the mincut solution $\mathbf{y}^j$ at each iteration is globally optimal for the graph constructed during that iteration. This allows for large stable "moves" in the labeling space between iterations, unlike variational methods where the updates are limited by the time step. These large moves help avoid "shallow" local minima and facilitate faster convergence.

To illustrate the iterative segmentation process, figure 4.4 shows the segmentation of a hand along with several intermediate steps. The hand in the image is an affine transformed version of the shape prior template. Starting with the initialization in figure 4.4(b), the labeling is updated according to algorithm 4.4.2. For the intermediate steps, the red curves show the current labeling and the green curves show the aligned shape prior. The middle example took 16 iterations and less than 1 second to converge.

The bottom example took 25 iterations and less than 2 seconds to converge. The result without shape prior is also shown in figure 4.4(c) for comparison.



(a) shape prior (b) initialization (c) no prior

(d) intermediate step (e) intermediate step (f) with prior

(g) intermediate step (h) intermediate step (i) with prior

Figure 4.4. Segmentation of a hand with and without shape prior. Red curves show segmentation and green curves show aligned shape prior. The middle example took 16 iterations and less than 1 second to converge. The bottom example took 25 iterations and less than 2 seconds to converge.

4.4.3 Data model

We use the piecewise-constant occlusion model of Thiruvenkadam *et al.* [107] to define the data penalty and briefly describe it here. Assume that object j has constant intensity μ_j, and the background intensity is μ_0. The occlusion model for the image is given by

$$\hat{\mathbf{x}} = \sum_{j=1}^{M} \mu_j \bar{\mathbf{y}}^j + \sum_{k=2}^{M} \sum_{\ell=1}^{\binom{M}{j}} (-1)^{k-1} \mu_{k,\ell} \chi^{k,\ell} + \mu_0 \prod_{j=1}^{M} \mathbf{y}^j, \tag{4.18}$$

where $\chi^{k,\ell}$ is the ℓ^{th} unordered intersection of k labels from $\mathbf{y}$ and $\mu_{k,\ell}$ takes one value in $\{\mu_j\}_{j=1}^{M}$. Then the cost of using $\hat{\mathbf{x}}$ to approximate a given image $\mathbf{x}$ is

$$E_D(\mathbf{y}) = \sum_{p \in \mathcal{P}} (x_p - \hat{x}_p)^2. \tag{4.19}$$

For $M = 1$, equation (4.19) becomes a discrete version of the Chan-Vese energy [18]

$$E_D(\mathbf{y}^1) = \sum_{p \in \mathcal{P}} \left((x_p - \mu_1)^2 \bar{y}_p^1 + (x_p - \mu_0)^2 y_p^1 \right). \tag{4.20}$$

For $M = 2$, the image occlusion model is

$$\hat{I} = \mu_1 \bar{\mathbf{y}}^1 + \mu_2 \bar{\mathbf{y}}^2 - \mu_{2,1} \bar{\mathbf{y}}^1 \bar{\mathbf{y}}^2 + \mu_0 \mathbf{y}^1 \mathbf{y}^2. \tag{4.21}$$

After some rearranging and using the fact that $\bar{\mathbf{y}}^1 = \bar{\mathbf{y}}^1(\mathbf{y}^2 + \bar{\mathbf{y}}^2)$ and similarly for $\bar{\mathbf{y}}^2$, equation (4.19) becomes

$$E_D(\mathbf{y}) = \sum_{p \in \mathcal{P}} (x_p^2 + a_p^{00} \bar{y}_p^1 \bar{y}_p^2 + a_p^{01} \bar{y}_p^1 y_p^2 + a_p^{10} y_p^1 \bar{y}_p^2 + a_p^{11} y_p^1 y_p^2), \tag{4.22}$$

where

$$a_p^{00} = (\mu_1 + \mu_2 - \mu_{2,1})^2 - 2(\mu_1 + \mu_2 - \mu_{2,1})x_p, \tag{4.23a}$$

$$a_p^{01} = \mu_1^2 - 2\mu_1 x_p, \tag{4.23b}$$

$$a_p^{10} = \mu_2^2 - 2\mu_2 x_p, \tag{4.23c}$$

$$a_p^{11} = \mu_0^2 - 2\mu_0 x_p. \tag{4.23d}$$

Since the first term in equation (4.22) is independent of $\mathbf{y}^1$ and $\mathbf{y}^2$, it does not factor into the minimization. The data penalties are defined by setting $\theta^{00} = a^{00}$, $\theta^{10} = a^{10}$, $\theta^{01} = a^{01}$, and $\theta^{11} = a^{11}$ in equation (4.15). Similarly for $M > 2$, equation (4.19) can be factored into a sum of costs for the regions, and the data penalties are assigned accordingly. The intensities $\{\mu_j\}_{j=1}^{M}$ are estimated by solving a linear system of equations after each iteration, and the occlusion relationship can be easily inferred from these object intensities [107].

4.5 Multiple prior shapes

For many segmentation tasks, the image can contain objects with completely different shapes (see figure 4.11) or an object that exhibits shape variability, such as the side view of a walking person (see figure 4.10). In such situations, the prior shape energy must make use of a set of prior templates or the multiple instances of a single

object. The latter case is normally addressed by formulating the shape energy based on a statistical shape space [25, 26].

In this work, the multi-template shape energy is defined as a weighted sum of the distances between the templates and the labeling $\mathbf{y}$. Given N prior templates $\Psi = \{\psi^1, \ldots, \psi^N\}$, the shape prior energy is

$$E_S(\mathbf{y}|\Psi) = \frac{\sum_{n=1}^{N} \alpha(\mathbf{y}, \psi^n) \cdot d^2(\mathbf{y}, \psi^n)}{\sum_{n=1}^{N} \alpha(\mathbf{y}, \psi^n)}, \tag{4.24}$$

with $\alpha(\mathbf{y}, \psi^n)$ given in equation (4.17). The weight $\alpha(\mathbf{y}, \psi^n)$ is a measure of the similarity between $\mathbf{y}$ and ψ^n, and hence shapes that are "closer" to the labeling $\mathbf{y}$ are given higher weights. In fact, Dambreville *et al*. [26] showed that the relationship between the distance between two shapes in a feature space constructed using kPCA, denoted $d_F^2(\psi^a, \psi^b)$, and the distance $d^2(\psi^a, \psi^b)$ is

$$d_F^2(\psi^a, \psi^b) \propto 1 - \exp\left(-\frac{d^2(\psi^a, \psi^b)}{2\sigma_s^2}\right) = 1 - \alpha(\psi^a, \psi^b). \tag{4.25}$$

It is reasonable then to assume that $\alpha(f, \psi^n)$ is a good measure of similarity between shapes in a feature space, and our experimental results reflect this fact.

4.6 Experiments

We present results on experiments on real and synthetic images. All experiments were run in MATLAB on a PC with a 2.16 GHz Intel Core Duo processor and 2GB

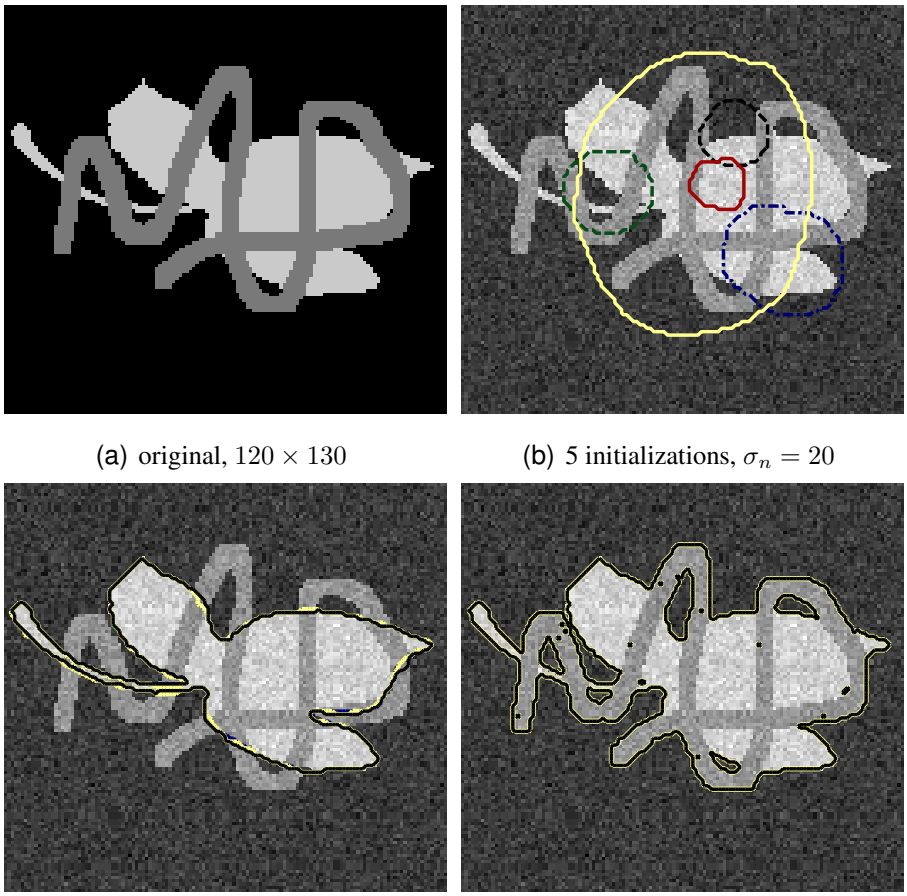
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 4.5. Shape templates. First four templates available at http://www.lems.brown.edu/~dmc/main.html.

of RAM. We use a MATLAB mex wrapper to interface with the C++ maxflow code of Boykov and Kolmogorov [9]. The run time can be improved significantly by recycling the graph and the search trees during graph cuts [61], but our current implementation does not make use of such a scheme.

The image size, number of iterations (iter), run time (sec), and parameter settings are indicated directly in the figures. For several examples, the results without shape prior are provided for comparison. Only the gray level intensity is used with $x_p \in [0, 255]$. The shape parameter σ_s, and correspondingly λ_s, is found to depend on the particular shape template, and we are currently investigating ways of determining the optimal σ_s for a given shape. Since the magnitude of the data penalty is in the range of x_p^2, the parameters λ_x and λ_s are shown with a scaling factor of 255^2. In all experiments, either the 4- or 8-connected neighborhood system is used, and convergence is reached when there is less than 1% change in the labeling(s).

The shape prior templates used in our experiments are shown in figure 4.5, and the target objects that we wish to segment are affine transformed versions of these tem-



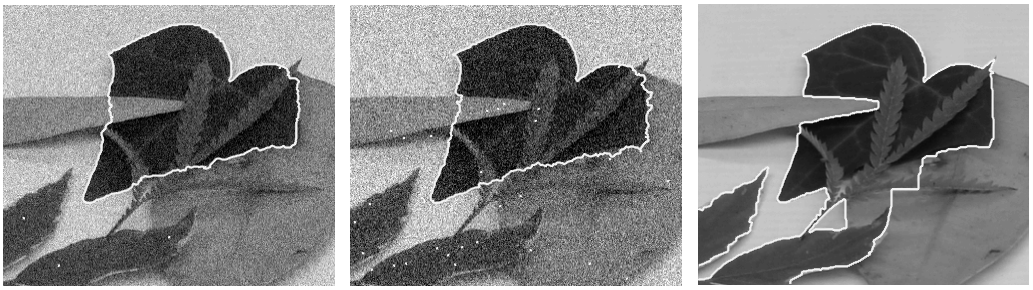
(a) original, 120×130 (b) 5 initializations, $\sigma_n = 20$

(c) shape prior (results overlap) (d) no shape prior

Figure 4.6. Five different initializations produced nearly identical results. The longest run took 11 iter, 0.921 sec. The parameters are set as $\lambda_x = \lambda_s = 0.5 \times 255^2, \sigma_s = 3$ and $\sigma_x = 15$.

plates. Figure 4.6 shows an image of a leaf produced by an affine transformation of the template in figure 4.5(a) with an added occlusion and Gaussian noise with standard deviation $\sigma_n = 20$. To demonstrate the algorithm's robustness to initializations, five different initial contours are used, as shown in figure 4.6(b). The results in figure 4.6(c) are nearly identical and difficult to separate because they overlap. Figure 4.6(d) pro-

(a) initialization, 279×330 (b) 5 iter, 1.016 sec (c) 9 iter, 1.554 sec



(d) 13 iter, 2.340 sec (e) 6 iter, 1.198 sec (f) no shape prior

Figure 4.7. (b-e) Shape prior segmentation results with increasing noise levels $\sigma_n = 0,\ 10,\ 20,\ 30$, respectively. The parameters are $\lambda_x = 0.5 \times 255^2,\ \lambda_s = 0.3 \times 255^2$, and $\sigma_s = 2$. To accommodate the noise levels, σ_x is adjusted to 10, 20, and 30 for images (c-e), respectively.

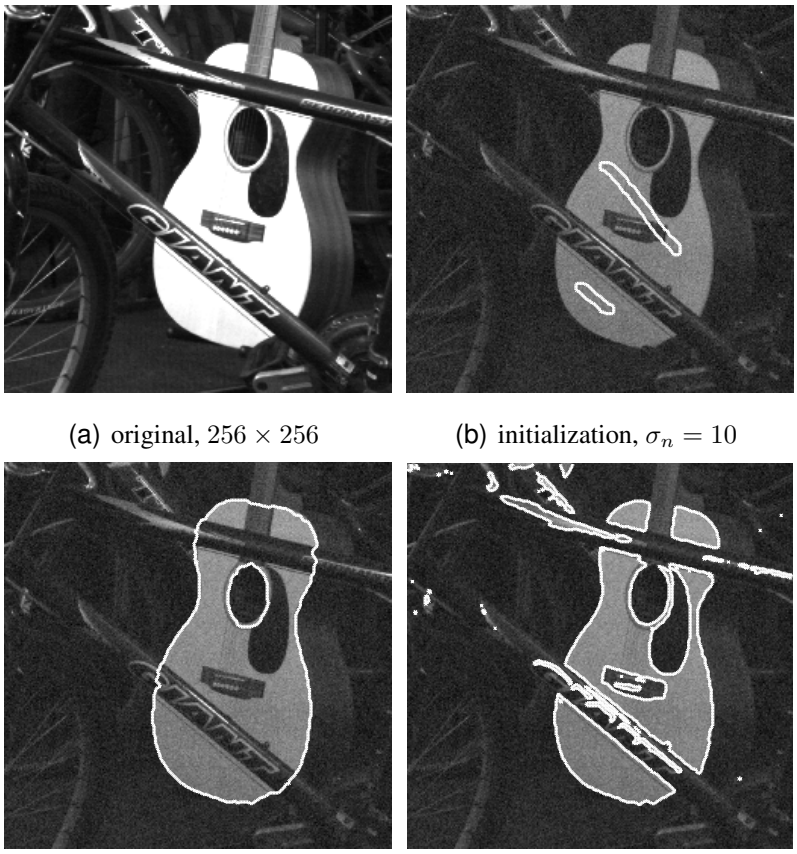
vides a comparison when no shape prior is used. Notice that the occluding scribble is identified as the object since its intensity is more similar to that of the leaf's than the background. For all five initializations, the algorithm converged quickly within a second, requiring 11 iterations or less.

Figure 4.7 shows the segmentation of a leaf with large occlusions and surrounding clutter. The shape prior template used is shown in figure 4.5(g). Gaussian noise was added with increasing levels $\sigma_n = 0,\ 10,\ 20,\ 30$, and correspondingly the parameter

σ_x is adjusted to $\{10, 10, 20, 30\}$ to accommodate the noise. Three small regions on the leaf are used for initialization, but do not provide a very good estimate of the final object's pose. The longest run took only 13 iterations and 2.3 seconds to converge. Note that the segmentations are very similar despite the noise increase. The segmentation without shape prior is shown in figure 4.7(f), where the error is mainly cased by the intensity similarity of the target leaf and the leaf in the lower left corner. The occluding leaf with intensity value similar to the background is another source of error.

Figure 4.8 shows the result of segmenting a guitar body using the template in figure 4.5(j). Their is significant occlusion by the bicycle in front, and the background is highly cluttered. Gaussian noise with $\sigma_n = 10$ is also added. Two small regions on the guitar are used for initialization, and again this selection does not provide a good estimate of the target object. The shape prior result in figure 4.8(c) shows that the guitar has been accurately segmented. Both the guitar's pickguard and bridge can be considered as occlusions, but because of the shape template, they are correctly labeled. However, the result without shape prior in figure 4.8(d) shows that these parts have been incorrectly labeled as background.

To demonstrate the algorithm's ability to incorporate multiple shape priors into the segmentation, we performed segmentation on a video sequence of a human figure walking across the screen. The video is available online from [99]. To obtain the shape prior templates, we extracted several evenly spaced frames during a half cycle of the walking



(a) original, 256×256 (b) initialization, $\sigma_n = 10$

(c) shape prior (d) no shape prior

Figure 4.8. Segmentation of a guitar body occlude by a bicycle frame. The algorithm took 18 iterations and 3.036 seconds. The parameters are $\lambda_x = 1.25 \times 255^2$, $\lambda_s = 0.625 \times 255^2, \sigma_s = 2.2$ and $\sigma_x = 10$.

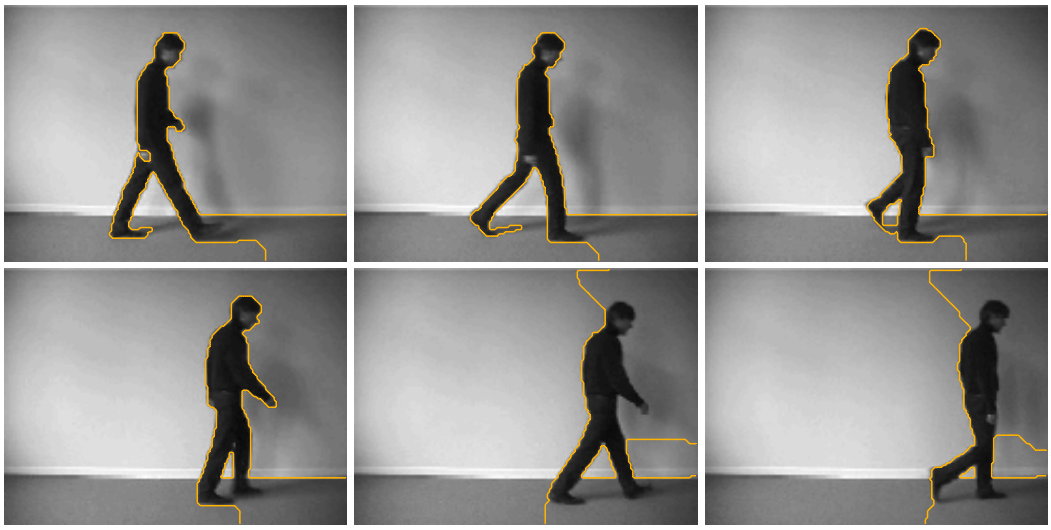
motion and manually segmented the figure from these frames. A total of ten silhouettes are used as the priors, five of which are shown in figure 4.9. Figure 4.10(b) shows the segmentation results for several frames without using shape prior, and figure 4.10(d) shows the results when the prior templates are used. For both cases, all parameters were kept the same. In most frames, the algorithm converged after 3 iterations, with

Figure 4.9. Silhouette templates used for multishape prior segmentation.

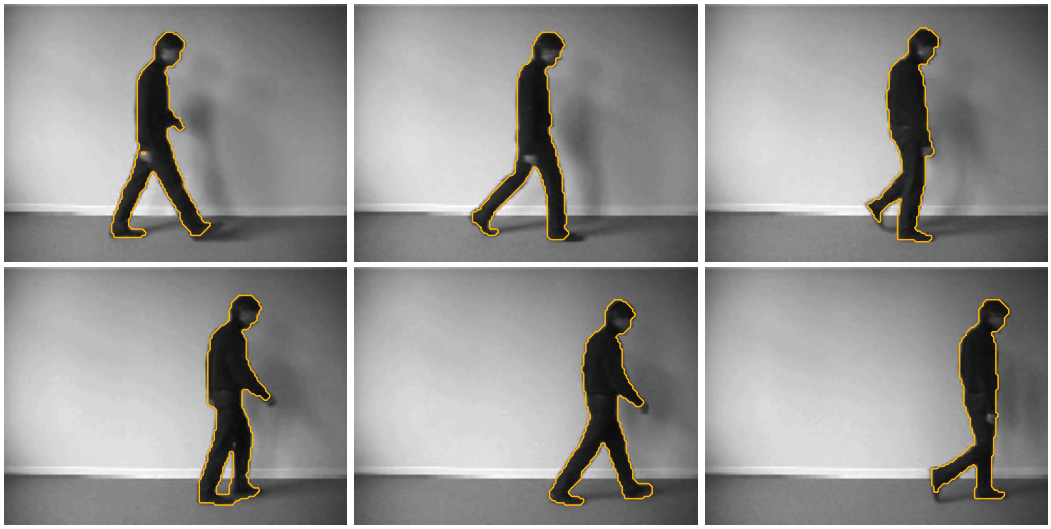
one frame requiring 10 iterations. The results clearly indicates the importance of using prior shape knowledge.

The result of segmentation of two leaves is shown in figure 4.11. The prior templates in figures 4.5(g) and 4.5(i) are used with the shape prior energy in equation (4.24). Not that the initializations for both labelings do not indicate a strong preference for either of the template. Both template are used for each labeling, but the one that more closely resembles the current labeling during the iterative process will have a stronger weight. The estimated images $\hat{\mathbf{x}}$'s using equation (4.18) for both the shape prior and no shape prior cases are almost identical, but using shape prior information encourages the two labelings to take on the correct shapes. Figure 4.12 shows another example of the two object segmentation case.

Figure 4.13 shows the results of segmenting three objects. The result in figure 4.13(b) is obtained using three labelings and the prior template in figure 4.5(e). Note that the three objects have very different poses and have large overlapping parts. The result in figure 4.13(d) also uses three labelings, but the three different templates in fig-

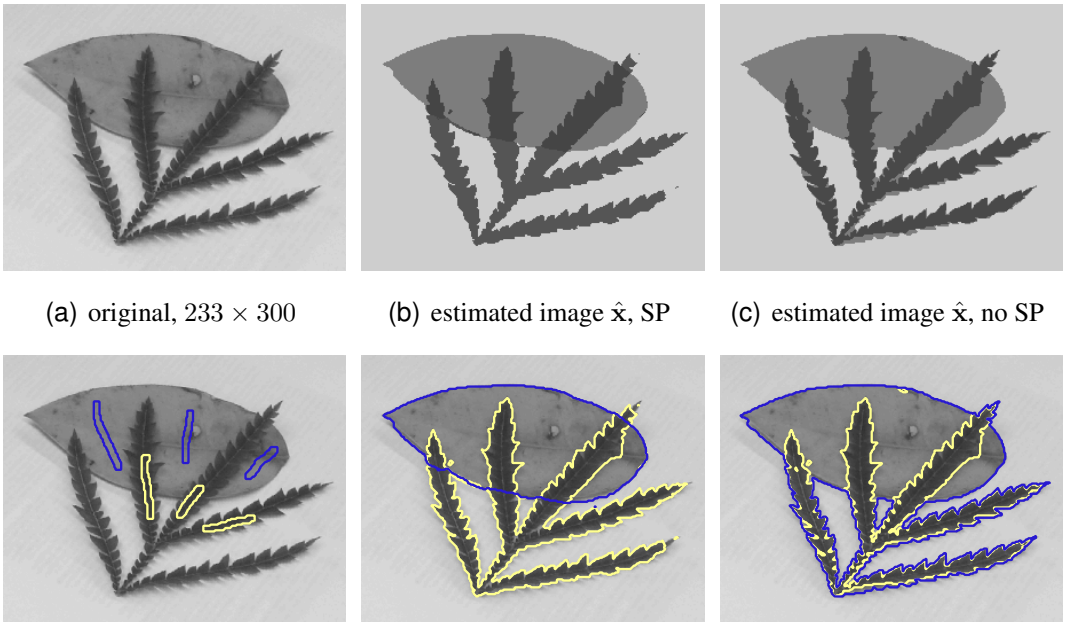


(b) without shape prior



(d) with shape prior

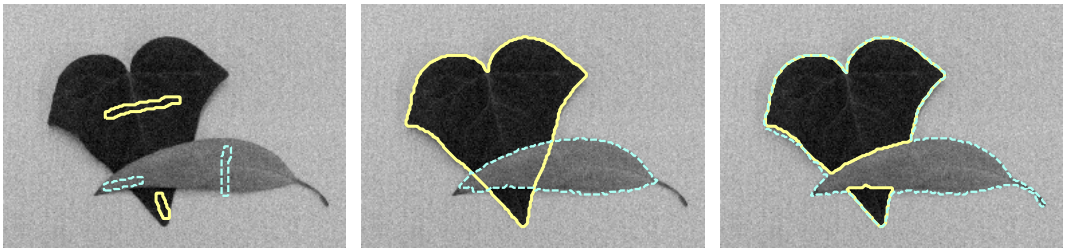
Figure 4.10. Segmentation of a walking sequence without (a) and with (b) shape prior information. Five of the ten shape templates are shown in figure 4.9. The templates were acquired from another segment of the video not included in the segmentation shown here.



(a) original, 233×300 (b) estimated image $\hat{\mathbf{x}}$, SP (c) estimated image $\hat{\mathbf{x}}$, no SP

(d) initialization (e) shape prior (f) no shape prior

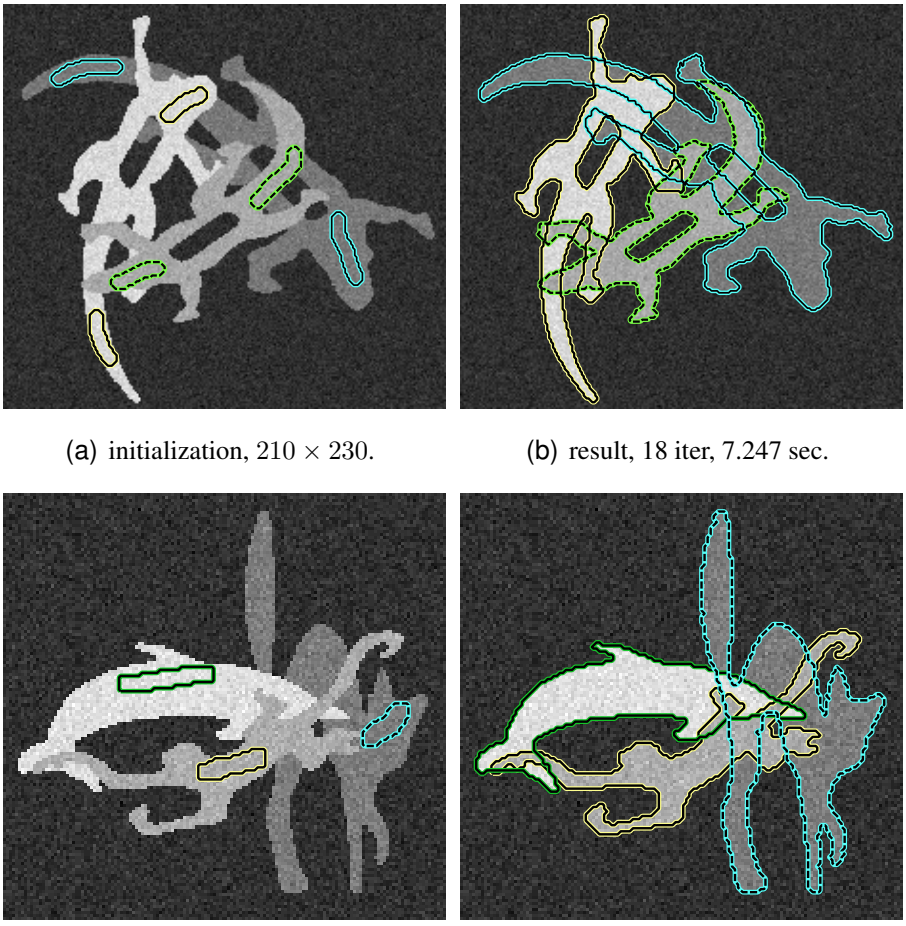
Figure 4.11. Segmentation of two overlapping objects. The process took 20 iteration and 13.864 seconds. $\lambda_x = 0.17 \times 255^2$, $\lambda_s = 0.12 \times 255^2$, $\sigma_s = 2$ and $\sigma_x = 10$.



(a) initialization, 182×256 (b) shape prior (c) no shape prior

Figure 4.12. Another example of two object segmentation, which took 14 iteration totalling 4.807 seconds. $\lambda_x = 0.15 \times 255^2$, $\lambda_s = 0.17 \times 255^2$, $\sigma_s = 3$, $\sigma_x = 20$, and $\sigma_n = 10$.

ures 4.5(b), 4.5(c), and 4.5(d) are used for each labeling. In all experiments, the objects are affine transformed versions of the templates, and the initializations do not provide good estimates of the shapes' poses nor do they more strongly favor any particular template for the cases with multiple priors.



(a) initialization, 210×230. (b) result, 18 iter, 7.247 sec.

(c) initialization, 140×145. (d) result, 12 iter, 5.320 sec.

Figure 4.13. Three objects: (a,b) Single template used with parameters $\lambda_x = \lambda_s = 0.25 \times 255^2, \sigma_s = 1.5$ and $\sigma_x = 15$. (c,d) Three templates used with parameters $\lambda_x = 0.35 \times 255^2$, $\lambda_s = 0.23 \times 255^2, \sigma_s = 2$ and $\sigma_x = 15$. Noise level $\sigma_n = 15$ for both images.

4.7 Conclusion

We presented a new method capable of segmenting multiple objects with possible overlaps. Our framework combines several ideas. First, the shape prior information is incorporated into the graph via the t-weights. Unlike those of many previous graph based approaches, the shape distance is both symmetric and obeys the triangle inequality. Second, we introduced the multiphase graph cuts, whereby the simultaneous segmentation of multiple objects is simplified to a binary labeling problem for each object. Furthermore, we extend the shape prior energy to incorporate multiple shape priors, which is necessary when the object exhibits variability or when several different objects are present in the image. A major advantage of our framework over variational methods is that it explicitly minimizes the segmentation energy and thereby avoids the computation of the energy gradient, which can be difficult and often requires approximations. The results show that the algorithm is insensitive to initializations and noise and is efficient in practice.

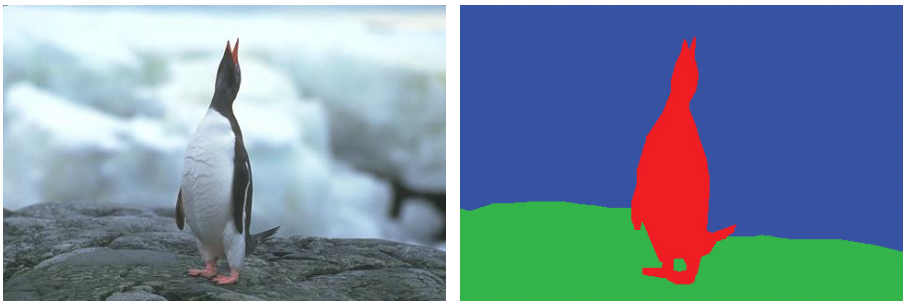
There are several potential directions for future work and improvement. First, the algorithm is sensitive to the shape parameter σ_s, which can be difficult to fine tune, and it appears that the value of σ_s is dependent on the shape prior template. One can consider this parameter as controlling the "cooling temperature" for the iterative segmentation process. A small σ_x tend to enforce the shape prior penalty too soon, neglecting the

data dependent term. For large values of σ_x, the shape prior is not enforced until the evolving labeling almost exactly resembles the template. However due to noise, occlusion, and clutter, the data term will prevent the evolving contour from ever getting close to the object. The proposed algorithm would greatly improve if an automatic method to set σ_x was available. Second, the occlusion model in equation (4.18) assumes a constant intensity for each object and limits the applicability of the algorithm on a wide range of data that contains texture or color. Developing an appearance model based on other visual features of the object besides intensity would of great value. Finally, the multiphase graph cut method, like its level set counterpart but to a lesser extent, can become computationally expensive for segmenting images containing a large number of objects. It would be worthwhile to incorporate graph recycling methods [61] to improve the algorithm's efficiency.

Chapter 5

Globally Optimal Nested Layer Segmentation

The segmentation of a general image, such as one depicting an office or outdoor scene, tends to produce junction points where three or more regions with different labels meet. This result is expected and unavoidable since there are no restrictions on the layout of the different objects or regions in the scene. For example, the segmentation of the outdoor scene in figure 5.1(a) depicting a penguin standing on a rock with icebergs in the background contains triple junctions where the penguin, rock, and sky meet (see figure 5.1(b)). However in many common biological and medical images, the spatial relationships among the image regions inherently reflect those of the anatomical structures being imaged. For a special subclass of these images, the regions exhibit

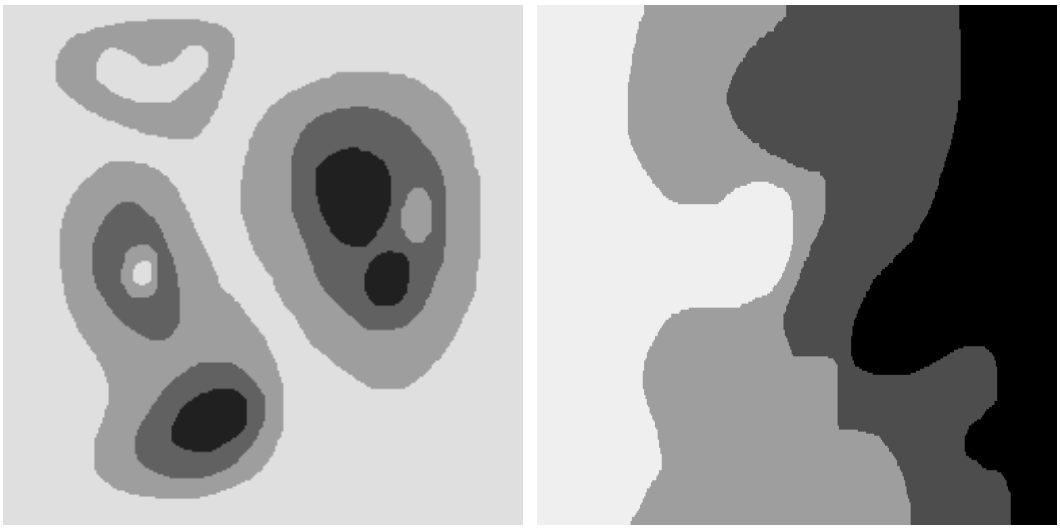


(a) (b)

Figure 5.1. Example of an outdoor scene [79]. The segmentation contains triple junctions where the red, green, and blue regions meet.

a nested relationship such that it is possible to partition the image where no junctions exist.

In this chapter, we address the problem of nested layer segmentation and show that the globally optimal solution can be found efficiently with graph cuts. Using a multi-label Conditional Random Field (CRF) model, the segmentation is posed as a pixel labeling problem with an additional label adjacency constraint to prevent junctions . The original multi-label CRF energy is transformed into an equivalent function of boolean variables through a boolean encoding scheme. We show that the resulting boolean energy is submodular, graph representable, and can be minimized exactly and efficiently with graph cuts. Our experimental results on both synthetic and real images demonstrate the utility of our algorithm.



(a) (b)

Figure 5.2. Images with nested layer topology. Each intensity value indicates a distinct label (4 labels total).

5.1 Nested Region Topology

An image having a nested region topology is one where, given an ordered set of labels, it is possible to assign labels to the image regions such that adjacent regions have consecutive labels from the set. For such a segmentation, it is not possible to have junctions where three or more regions meet. Images with nested regions are uncommon in natural or uncontrived scenes [79], but occur frequently in biological and medical datasets. The synthetic images in figure 5.2 depict some nested regional relationships often seen in the biomedical image domain. The regions of the image in figure 5.2(a) exhibits an "inside-outside" nesting relationship where one label can be said to encapsulate another label. For example, this type of nestedness are observed in

MRI images of the brain where the white matter is encapsulated by the gray matter, and in turn the gray matter is encapsulated inside the cerebral spinal fluid (CSF). However, the white matter and the CSF do not touch and no triple junctions are present. Figure 5.2(b) shows an image with the regions having a "before-after" nested relationship. This type of nestedness is frequently observed in images of tissue cross-sections, such as the epidermis layers or the mammalian retina layers. Images with nested layers are not restricted to biomedical domain but can also be found in other application areas, such as the geosciences.

Since segmentation is an ill-posed problem in general, we would like to utilize the nested layer relationship, if one exists, to improve the segmentation results. Previous works have shown that using additional priors, such as an object's shape, can dramatically enhance an algorithm's ability to correctly segment the image [71, 93, 111]. As our results will demonstrate, incorporating knowledge of the nested layer relationship will indeed improve the segmentation. An additional benefit of using this prior information is that we are able to constraint the segmentation to return only those results that are anatomically feasible. For example, we can prevent a pixel labeled white matter from being adjacent one labeled CSF.

5.1.1 Label Adjacency Constraint

More formally, for an image $\mathbf{x}$ with pixel set $\mathcal{P} = \{1, 2, \ldots, N\}$ and an ordered label set $\mathcal{L}$, we seek a labeling $\mathbf{y} \in \mathcal{Y}'$ that minimizes the CRF energy

$$E(\mathbf{y}) = \sum_{p \in \mathcal{P}} V_p(y_p) + \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} V_{pq}(y_p, y_q) \tag{5.1}$$

subject to the label adjacency constraint (LAC)

$$|y_p - y_q| \leq 1, \quad \forall y_p, y_q \in \mathcal{L}. \tag{5.2}$$

In equation (5.1) as well as for the remainder of this chapter, our notation do not explicitly indicate the dependency of the CRF energy and the clique potentials on $\mathbf{x}$, but it is assumed that this dependency exists. Condition (5.2) restricts neighboring site pairs from having labels that differs by more than one. This condition is the same as forcing the regions in the final segmentation to have a nested relationship. Without loss of generality and for convenience, we specify the label set to be $\mathcal{L} = \{1, 2, \ldots, K\}$. As a result of enforcing condition (5.2), the number of feasible labelings $\mathbf{y} \in \mathcal{Y}'$ can be significantly smaller than $|\mathcal{L}^N|$. Here, the constrained labeling space $\mathcal{Y}' \subset \mathcal{Y} = \mathcal{L}^N$.

For the remainder of this chapter, we provide a review of related works on layer segmentation. Then we describe in details how to minimize equation (5.1) exactly, subject to the constraint in equation (5.2), and provide the graph construction used for optimization with graph cuts. Note that without the LAC, the optimization problem is NP-hard and the globally optimal solution is not guaranteed. Finally the experimental

results on both synthetic and real images are provided to demonstrate the utility of our algorithm.

5.2 Related Works

We are aware of only one other segmentation method that directly addresses the segmentation of images with nested layers. In [20], Chung and Vese propose a multilayer level set approach motivated by island dynamics in epitaxial growth. In traditional level set methods, the zero level set is used to embed the evolving front or curve, dividing the image into two regions, inside and outside of the curve. The multilayer level set method uses the multiple level lines of the level set function to represent the fronts or boundaries separating multiple (> 2) regions. This is analogous to using the contour lines of a topographic map to represent the levels of elevation. Effectively, the different regions are implicitly nested, and this relationship is maintained throughout the level set evolution. An advantage of using the multi-level formulation is that only one level set function is used to segment multiple regions in the image. This is in contrast to the multiphase approach [124, 113], which requires more than one level set function and is thus less efficient.

However, the multilayer level set approach has several limitations. As pointed out by the authors, the algorithm does not guarantee a globally optimal solution and is

sensitive to the initialization. Since this method relies on performing gradient descent on an energy functional, these limitations are to be expected. Secondly, the selection of the level lines to be used as the region boundaries is manually specified by the user. While this task may not pose a problem for images with few regions, it may potentially become difficult to choose the right set of values when the number of regions becomes large.

In the graph cut formulation, Ishikawa presented a method to find the exact minimizer of equation (5.1) given that the label set is ordered and the pairwise clique potential is a convex function of the label difference [51]. To enforce the LAC in equation (5.2), we can use the convex function

$$V_{pq}(y_p, y_q) = c \cdot f(y_p - y_q) = c \cdot |y_p - y_q|^{\lambda}, \tag{5.3}$$

for $\lambda \rightarrow \infty$ and c is a quantity that can depend on the image data. Notice that when $|y_p - y_q| > 1$, the pairwise cost becomes prohibitively high, and consequently this type of pairwise labeling is prevented.

In theory, the method of Ishikawa [51] can solve the nested layer segmentation problem without much difficulty. Yet in the implementation, there are two obvious limitations that affect the efficiency and accuracy of the segmentation. The most noticeable issue is the scalability of the graph required for optimization. When $\lambda \geq 2$ in equation (5.3), the number of edges in the graph is on the order of $\mathcal{O}(N \cdot K^2)$, where N is the number of pixels and K is the number of labels. Even for modest image sizes

(256^2), the graph size can grow quickly as the number of labels increases. As we will show, the graph construction for our method reduces the number of edges to be on the order of $\mathcal{O}(N \cdot K)$. The second limitation deals with the numerical overflow of the energy calculation during graph cut optimization. In the graph structure of [51], the edge weights between neighboring nodes are proportional to the second difference of $f(y_p - y_q)$, *i.e.* $f(y_p - y_q + 1) - 2f(y_p - y_q) + f(y_p - y_q - 1)$, and as a result, the majority of these edge weights have very large values. Most feasible minimum cost cut will inevitably sever a large number of these edges, and consequently the energy computation suffers from numerical overflow and returns suboptimal results.

Finally we would like to mention two notable works that use some form of constraint on the region labels, though the methods presented therein are not directly applicable to the task of nested layer segmentation. The level set algorithm of Yang *et al*. [122] uses information about the spatial configuration of objects to model the interobject constraint. The shape and position relationships among the objects are learned from a set of training images and are used in conjunction with the image intensity to simultaneously segment the objects. Liu *et al*. [74] propose a pair of order-preserving moves for the purpose of geometric class scene labeling using graph cuts. A series of horizontal and vertical moves are made iteratively to segment the image into five regions: center, top, bottom, left, and right. At each step the method of Ishikawa [51] is used to solve for the optimal move, but overall the globally optimal solution is not guaranteed.

5.3 Graph Representation of $\mathcal{M}_s^2$ Functions

We reiterate that solving for the exact minimizer of the multi-label CRF energy in equation (5.1) is NP-hard in general [65]. However there exist certain forms of the energy function that allow for an exact solution. The most common case is when $\mathcal{L} = \{0, 1\}$ and the second order or pairwise potential $V_{pq}(y_p, y_q)$ is submodular [45, 65, 38]. We denote these submodular second order functions of binary variables as $\mathcal{F}_s^2$. For $|\mathcal{L}| > 2$ and $\mathcal{L}$ an ordered set, Ishikawa [51] gave conditions for the pairwise cost $V_{pq}(y_p, y_q)$ to be exactly minimized. Along the same line, using pseudo-boolean optimzation [6], Schlesinger and Flach [97] also showed that multi-label CRFs with convex energy functions of order two can be minimized exactly in polynomial time. Subsequently, Ramalingam *et al.* [88] outlined a more extensive set of conditions for exactly minimizing multi-label CRF energy with higher order potentials and described a principled framework for transforming the class of submodular multi-label kth order functions, denoted $\mathcal{M}_s^k$, into an equivalent class of submodular second order boolean functions $\mathcal{F}_s^2$. In this section, we review the boolean transformation technique presented in [88] (using similar notations), which is necessary for our LAC formulation in the next section.

5.3.1 Boolean Transformation $\mathcal{M}_s^2 \rightarrow \mathcal{F}_s^2$

The key idea in transforming the multi-label function into one of binary variables is to use two or more boolean variables to encode the states of a single multi-label variable. The transformation is accomplished by defining a set of encoding functions, which replace all occurrences of the multi-label variable with that of the encoding boolean variables. For a given multi-label second order function $E(\mathbf{y})$ with $\mathbf{y} \in \mathcal{Y}$, the transformation will result in a boolean second order function $E_{\text{bin}}(\mathbf{z})$, where $\mathbf{z}$ belongs to the space of boolean labelings $\mathcal{Z}$. The transformation must satisfy two conditions [88]:

- The transformation $T : \mathcal{Y} \rightarrow \mathcal{Z}$ must be one-to-one (injective) between the feasible labelings of $\mathbf{z} \in \mathcal{Z}$ and $\mathbf{y} \in \mathcal{Y}$ and bijective between the set of optimal labelings of the boolean and multi-label variables.

- The minimum value of $E(\mathbf{y})$ over $\mathbf{y}$ must equal to the minimum value of $E_{\text{bin}}(\mathbf{z})$ over $\mathbf{z}$, but these energies do not have to be equal at labelings other than their respective minima.

We begin by summarizing the boolean encoding of a unary multi-label variable.

5.3.2 Encoding Unary Multi-label Variables

The unary potential in equation (5.1) can be rewritten as

$$V_p(y_p) = \sum_{i \in \mathcal{L}} \theta_{p;i} \delta(y_p, i), \tag{5.4}$$

where $\theta_{p;i}$ is the potential for assigning label $y_p = i$ to site p, and

$$\delta(y_p, i) = \begin{cases} 1 & \text{if } y_p = i \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

In order to make equation (5.4) a function of boolean variables, the multi-label terms $\delta(y_p, i)$ should be replaced with boolean functions $f_{y_p;i}(\mathbf{z}_p)$. Here $\mathbf{z}_p = \{z_p^1, z_p^2, \ldots, z_p^M\}$, with $z_p^i \in \{0, 1\}$, and M is the number of boolean variables used to encode the multi-label variable. It follows that the function $f_{y_p;i}(\mathbf{z}_p)$ should equal 1 for $y_p = i$ and 0 otherwise.

One possible scheme [97] is to encode a K-label variable y_p using $K - 1$ boolean variables $\{z_p^1, z_p^2, \ldots, z_p^{K-1}\}$ such that

$$\{y_p = i\} \leftrightarrow \left\{z_p^1 z_p^2 \ldots z_p^{K-1} = \{\mathbf{0}_{(i-1)} \mathbf{1}_{(K-i)}\}\right\} \tag{5.6}$$

where we use the notation

$$\mathbf{0}_{(i-1)} = \underbrace{00\ldots0}_{i-1} \text{ and } \mathbf{1}_{(K-i)} = \underbrace{11\ldots1}_{K-i}. \tag{5.7}$$

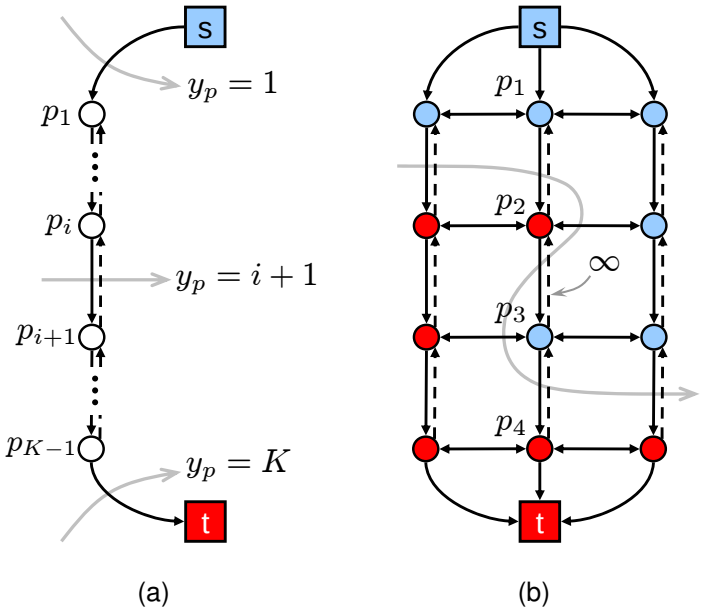


Figure 5.3. (a) Graph construction for unary variable encoding. (b) Example of an infeasible cut (gray arrow). Blue nodes belong to the source set $\mathcal{S}$ and red nodes belong to the sink set $\mathcal{T}$. Edge (p_3, p_2) is in the cut and has infinite weight, making the cut cost prohibitively high.

As an example [88], for a 4-label variable $y_p \in \mathcal{L} = \{1, 2, 3, 4\}$, the encoding using three binary variables is given by

$$\begin{aligned} \{y_p = 1\} &\leftrightarrow \{z_p^1 z_p^2 z_p^3 = \{111\}\} \\ \{y_p = 2\} &\leftrightarrow \{z_p^1 z_p^2 z_p^3 = \{011\}\} \\ \{y_p = 3\} &\leftrightarrow \{z_p^1 z_p^2 z_p^3 = \{001\}\} \\ \{y_p = 4\} &\leftrightarrow \{z_p^1 z_p^2 z_p^3 = \{000\}\}. \end{aligned} \tag{5.8}$$

The graph construction corresponding to this encoding is shown in figure 5.3(a). Each multi-label variable y_p is encoded by $K - 1$ nodes $\{p_1, p_2, \ldots, p_{K-1}\}$. Using the convention that, after the cut, $p_i \in \mathcal{S}$ implies $z_p^i = 0$ and $p_i \in \mathcal{T}$ implies $z_p^i = 1$, the

cuts corresponding to $y_p = i$ for $i \in \mathcal{L}$ result in the binary labelings in equation (5.6). Furthermore, to ensure that each cut has a corresponding cost equal to the unary energy in equation (5.4), the edge weights are assigned as:

$$w_p^{s,1} = \theta_{p;1} \tag{5.9a}$$

$$w_p^{i,i+1} = \theta_{p;i+1} \tag{5.9b}$$

$$w_p^{K-1,t} = \theta_{p;K}. \tag{5.9c}$$

Here $w_p^{s,1}$, $w_p^{i,i+1}$, and $w_p^{K-1,t}$ are the weights of directed edges (s, p_1), (p_i, p_{i+1}), and (p_{K-1}, t), respectively.

Notice that for the above example, the three boolean variables can encode a maximum of $3^2 = 8$ labelings. However, the labelings $z_p^1 z_p^2 z_p^3 = \{010, 100, 101, 110\}$ are unused, and cuts resulting in these labelings must be made infeasible. This is accomplished by adding infinite capacity edges (p_{i+1}, p_i) for $i = \{1, 2, \ldots, K-2\}$, which makes the cuts corresponding to the unused labelings have prohibitively high costs [51]. These edges are shown as dashed arrows in figure 5.3(a). Figure 5.3(b) shows an example of an infeasible cut, where according to our graph cut convention, the boolean encoding for y_p is $z_p^1 z_p^2 z_p^3 z_p^4 = \{0101\}$ and edge (p_3, p_2) is in the cut. However, since this edge has infinite weight, such a cut is prevented. In general, the penalty for the infeasible cuts can be expressed as

$$P(\mathbf{z}_p) = \sum_{i=1}^{K-2} \lambda z_p^i \bar{z}_p^{i+1}, \tag{5.10}$$

with $\lambda \rightarrow \infty$ and $\overline{z} = 1 - z$.

Finally, it is fairly straightforward to deduce from the previous encoding scheme that the function $f_{y_p;i}(\mathbf{z}_p)$ satisfies

$$f_{y_p;i}(\mathbf{z}_p) = \begin{cases} z_p^1, & i = 1 \\ z_p^i - z_p^{i-1}, & 2 \leq i \leq K-1 \\ 1 - z_p^{K-1}, & i = K. \end{cases} \tag{5.11}$$

A more detailed derivation can be found in [88]. Thus, $f_{y_p;i}(\mathbf{z}_p)$ can be substituted for every instance of $\delta(y_p, i)$ in equation (5.4), thereby transforming the multi-label function $V_p(y_p)$ into one of boolean variables. With the graph construction in figure 5.3(a), the multi-label first order potential in equation (5.4) can be minimized exactly using graph cuts. We would like to note that other boolean encoding schemes are also possible [88], but our results rely on the one presented here.

5.3.3 Encoding Pairwise Multi-label Variables

Similar to the unary potential case, the pairwise potential in equation (5.1) can be expressed as

$$V_{pq}(y_p, y_q) = \sum_{i,j \in \mathcal{L}} \theta_{pq;ij} \delta(y_p, i) \delta(y_q, j), \tag{5.12}$$

where $\theta_{pq;ij}$ is the potential associated with the pairwise label assignments of $y_p = i$ and $y_q = j$ to neighboring sites p and q, respectively. By substituting equation (5.11)

into equation (5.12), the pairwise multi-label potential becomes the boolean potential

$$V_{\text{bin}}(\mathbf{z}_p, \mathbf{z}_q) = \sum_{i,j \in \mathcal{L}_{\{-1\}}} \alpha_{ij} z_p^i z_q^j + L_1, \tag{5.13}$$

with $\mathcal{L}_{\{-1\}} = \{1, 2, \ldots, K-1\}$ and the coefficients

$$\alpha_{ij} = \theta_{pq;ij} - \theta_{pq;(i+1)j} - \theta_{pq;i(j+1)} + \theta_{pq;(i+1)(j+1)}. \tag{5.14}$$

The term L_1 is the sum of the first order terms and constants. In order to minimize equation (5.13) exactly, the coefficients must satisfy $\alpha_{ij} \leq 0$ (submodular condition) [38]. An example that satisfies the submodularity condition is the potential given in [51], *i.e.*

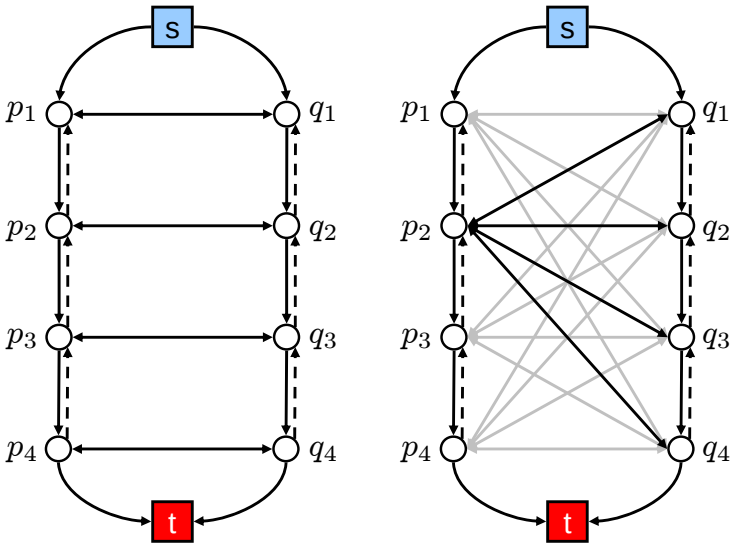
$$\theta_{pq;ij} = g(x_p, x_q) \cdot f(i-j), \tag{5.15}$$

where $f(i-j)$ is some convex function of the label difference $(i-j)$ and $g(x_p, x_p)$ is a nonnegative function of the observations x_p and x_q. The function

$$f(i-j) = |i-j|^k \tag{5.16}$$

with $k > 0$ is often used in practice.

The graph construction for a pair of variables $\{y_p, y_q\}$ with $f(i-j) = |i-j|^k$ and $K = 5$ is shown in figure 5.4 [51, 97]. The left and right columns of nodes encode the variables y_p and y_q, respectively. The nodes p_i and q_j are connected via two directed edges (p_i, q_j) and (q_j, p_i), and for simplicity these edges are represented by a single



(a) Graph for $k = 1$ in (5.16). (b) Graph for $k > 1$ in (5.16).

Figure 5.4. Graph constructions for the pairwise variable encoding according to [51, 97]. For simplicity, a bidirectional edge connecting p_i and q_j is used to represent the two directed edges (p_i, q_j) and (q_j, p_i). The number of edges grows according to $\mathcal{O}(N \cdot K^2)$ for $k > 1$.

bidirectional edge in the figure. The weight of edge (p_i, q_j) is

$$w_{p_i,q_j} = g(x_p, x_q) \cdot \frac{f(i-j+1) - 2f(i-j) + f(i-j-1)}{2}. \tag{5.17}$$

Using this weighting assignment, a feasible cut on the graph has a cost equal to the pairwise potential in equation (5.12), and the mincut corresponds to the labeling with lowest energy.

As mentioned in section 5.2, the nested layer segmentation problem can be solved if we choose $k \to \infty$ in equation (5.16). However, as the graph structure in figure 5.4(b) shows, the the number of edges in the graph can become quite large. More specifically,

let M_n and M_t be the total number of neighborhood and terminal edges, respectively, in the two-label problem. Then for a K label problem, the number of edges in the graph is $(K-1)^2 \cdot M_n + (K-1) \cdot M_t$, *i.e.* on the order of $\mathcal{O}(N \cdot K^2)$, where we have assumed that $M_n \sim \mathcal{O}(N)$. As a result, solving the nested layer segmentation using this graph structure leads to computational inefficiency and large memory requirements.

5.4 Minimizing $\mathcal{M}_s^2$ with Label Constraint

In this section, we transform the LAC in equation (5.2) for neighboring pairs of multi-label variables y_p and y_q into an equivalent constraint for the corresponding pairwise boolean variables $\mathbf{z}_p$ and $\mathbf{z}_q$. Then we show that the CRF energy $E(\mathbf{y})$, subject to condition (5.2), belongs to the class $\mathcal{M}_s^2$, and can be minimized exactly when

$$\theta_{pq;ii} - \theta_{pq;(i+1)i} - \theta_{pq;i(i+1)} + \theta_{pq;(i+1)(i+1)} \leq 0, \ \forall i \in \mathcal{L}. \tag{5.18}$$

Additionally, we provide a graph construction, which enforces the label constraint condition and for which the mincut cost gives the minimum energy.

5.4.1 Boolean Encoding with Adjacency Constraint

Recall that the set of labels is the ordered set $\mathcal{L} = \{1, 2, \ldots, K\}$. The constraint $|y_p - y_q| \leq 1$, with $y_p, y_q \in \mathcal{L}$, forces two neighboring sites p and q to have either the same label or consecutive labels from $\mathcal{L}$. Without loss of generality, assume $y_p \leq y_q$.

As an example for $K = 4$, the set of labelings $\{y_p, y_q\} = \{(1,3),(1,4),(2,4)\}$ violates this constraint. The boolean encodings using the scheme in equation (5.6) for these three cases are:

$$\begin{aligned}
\{y_p = 1, y_q = 3\} &\leftrightarrow \left\{z_p^1 z_p^2 z_p^3 = \{111\}, z_q^1 z_q^2 z_q^3 = \{001\}\right\} \\
\{y_p = 1, y_q = 4\} &\leftrightarrow \left\{z_p^1 z_p^2 z_p^3 = \{111\}, z_q^1 z_q^2 z_q^3 = \{000\}\right\} \\
\{y_p = 2, y_q = 4\} &\leftrightarrow \left\{z_p^1 z_p^2 z_p^3 = \{011\}, z_q^1 z_q^2 z_q^3 = \{000\}\right\}.
\end{aligned} \tag{5.19}$$

Observe that these illegal labelings all have in common at least one instance where a boolean variable pair $\{z_p^i, z_q^j\} = \{1, 0\}$ for $j \geq i+1$. According to the encoding scheme in equation (5.6), these assignments imply that if $y_p = i$ then $y_q > i + 1$. However it is clear that such an assignment violates the LAC.

We can state the LAC for the boolean variables more precisely. Given the boolean encodings $\mathbf{z}_p = \{z_p^1, z_p^2, \ldots, z_p^{K-1}\}$ and $\mathbf{z}_q = \{z_q^1, z_q^2, \ldots, z_q^{K-1}\}$ for the multi-label variable pair y_p and y_q, the constraint in equation (5.2) is equivalent to enforcing

$$z_p^i \overline{z}_q^j + \overline{z}_p^j z_q^i = 0 \text{ for } i \in \mathcal{L}_{\{-2\}},\ j > i, \tag{5.20}$$

where $\mathcal{L}_{\{-2\}} = \{1, 2, \ldots, K-2\}$. Finally, the penalty for the illegal pairwise boolean encodings is

$$P(\mathbf{z}_p, \mathbf{z}_q) = \sum_{i \in \mathcal{L}_{\{-2\}}, j > i} \lambda(z_p^i \overline{z}_q^j + \overline{z}_p^j z_q^i), \tag{5.21}$$

and we can express the boolean transformation of the multi-label CRF energy with adjacency constraint as

$$E_{\text{bin}}(\mathbf{z}) = \sum_{p\in\mathcal{P}} V_{\text{bin}}(\mathbf{z}_p) + \sum_{p\in\mathcal{P},q\in\mathcal{N}_p} \Big(V_{\text{bin}}(\mathbf{z}_p,\mathbf{z}_q) + P(\mathbf{z}_p,\mathbf{z}_q) \Big). \tag{5.22}$$

5.4.2 Submodularity of $\mathcal{M}_s^2$ with Label Constraint

In order to globally minimize equation (5.22), we have to show that the terms in the second summation, expressed in the form of equation (5.13), are submodular. That is all the coefficients α_{ij} of the second order terms $z_p^i z_q^j$ must be less than or equal to zero, *i.e.* condition (5.14) must be met. Observe that the pairwise boolean potential in equation (5.13) can be reexpressed as

$$\begin{aligned} V_{\text{bin}}(\mathbf{z}_p,\mathbf{z}_q) &= \sum_{i\in\mathcal{L}_{\{-2\}},j>i} \left(\alpha_{ij} z_p^i z_q^j + \alpha_{ji} z_p^j z_q^i\right) + \sum_{i\in\mathcal{L}_{\{-1\}}} \alpha_{ii} z_p^i z_q^i + L_1 \\ &= \sum_{i\in\mathcal{L}_{\{-2\}},j>i} \left(\alpha'_{ij} z_p^i \overline{z}_q^j + \alpha'_{ji} \overline{z}_p^j z_q^i\right) + \sum_{i\in\mathcal{L}_{\{-1\}}} \alpha_{ii} z_p^i z_q^i + L_1', \end{aligned} \tag{5.23}$$

where $\alpha'_{ij} = -\alpha_{ij}$ and

$$L_1' = L_1 + \sum_{i\in\mathcal{L}_{\{-2\}},j>i} \left(\alpha'_{ij} z_p^i + \alpha'_{ji} z_q^j\right). \tag{5.24}$$

With the reformulation above, exact minimization of equation (5.23) requires $\alpha_{ii} \leq 0$ and the coefficients of the second order terms $z_p^i \overline{z}_q^j$ and $\overline{z}_p^j z_q^i$ to be greater than or equal to zero. Exact minimization of L_1' is guaranteed since it is composed of first order terms and constants.

The boolean pairwise potential $V_{\text{bin}}(\mathbf{z}_p, \mathbf{z}_q)$ and the boolean penalty $P(\mathbf{z}_p, \mathbf{z}_q)$ can be combined to give

$$\begin{aligned} V'_{\text{bin}}(\mathbf{z}_p,\mathbf{z}_q) &= V_{\text{bin}}(\mathbf{z}_p, \mathbf{z}_q) + P(\mathbf{z}_p, \mathbf{z}_q) \\ &= \sum_{i\in\mathcal{L}_{\{-2\}}, j>i} \Big((\alpha'_{ij} + \lambda) z_p^i \overline{z}_q^j + (\alpha'_{ji} + \lambda) \overline{z}_p^j z_q^i \Big) + \sum_{i\in\mathcal{L}_{\{-1\}}} \alpha_{ii} z_p^i z_q^i + L'_1. \end{aligned} \tag{5.25}$$

Since $\lambda \rightarrow \infty$, all the coefficients of $z_p^i \overline{z}_q^j$ and $\overline{z}_p^j z_q^i$ are guaranteed to satisfy $(\alpha'_{ij} + \lambda) \geq 0$ and $(\alpha'_{ji} + \lambda) \geq 0$, and therefore the first summation in equation (5.25) can be minimized exactly. Consequently, the only requirement for the exact minimization of equation (5.25) is for $\alpha_{ii} \leq 0,\ i \in \mathcal{L}_{\{-1\}}$. In summary, to minimize equation (5.22), we must have

$$\theta_{pq;ii} - \theta_{pq;(i+1)i} - \theta_{pq;i(i+1)} + \theta_{pq;(i+1)(i+1)} \leq 0. \tag{5.26}$$

5.4.3 Label Adjacency Constraint Graph

Minimizing equation (5.25) with st-mincut techniques requires that all occurrences of the label pairs $z_p^i z_q^j = \{10\}$ and $z_p^j z_q^i = \{01\}$, where $i \in \mathcal{L}_{\{-2\}}$ and $j > i$, be made infeasible since these labelings violate the LAC. Recall that according the our graph cut convention, $z_q^j = 0$ if node $q_j \in \mathcal{S}$ and $z_p^i = 1$ if node $p_i \in \mathcal{T}$. Then to prevent the labelings $z_p^i z_q^j = \{10\}$, a set of directed edges (q_j, p_i) for $j > i$ with infinite weights should be added to the graph. Likewise, a set of directed edges (p_j, q_i) for $j > i$ with infinite weights should be added to prevent labelings $z_p^j z_q^i = \{01\}$. However observe

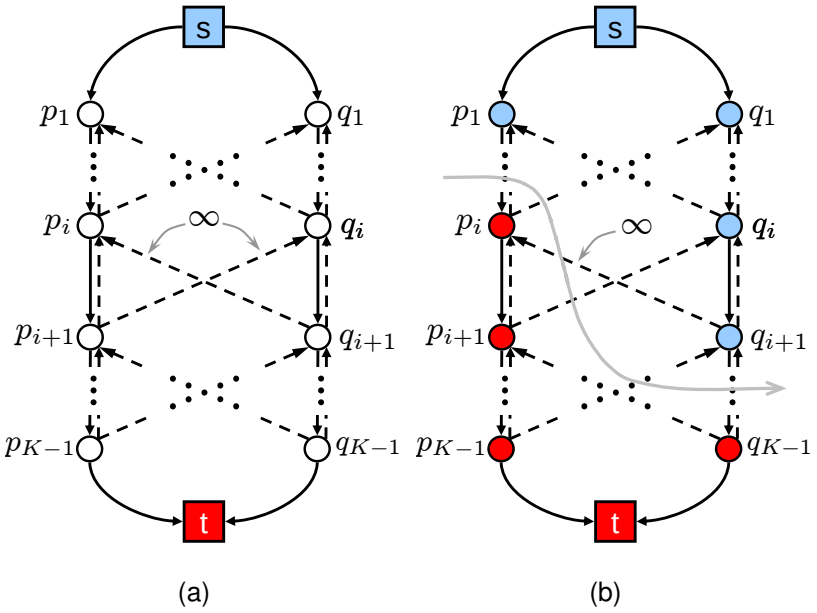


Figure 5.5. (a) The constraint edges (p_{i+1}, q_i) and (q_{i+1}, p_i) have infinity weight and enforce the label adjacency condition. (b) Example of an infeasible cut (gray arrow). The cut assigns $z_q^{i+1} = 0$ and $z_p^i = 1$, which would violate the constraint in equation (5.2). The edge (q_{i+1}, p_i) is in the cut, making the cut cost prohibitively high.

that only the edge (q_{i+1}, p_i) is needed to prevent all labelings $z_p^i z_q^j = \{10\}$ for $j > i$. This is due to the encoding scheme in equation (5.6), where $z_q^j = 0$ implies that z_q^i must also equal 0 for $i < j$. Therefore preventing the labeling $z_p^i z_q^{i+1} = \{10\}$ will also prevent $z_p^i z_q^j = \{10\}$ for $j > i + 1$. By similar reasoning, only the edge (p_{i+1}, q_i) with infinite weight is needed to prevent all labelings $z_p^j z_q^i = \{01\}$ for $j > i$.

The graph in figure 5.5(a) shows the infinity weighted edges used to enforce the LAC. Note that these are directed edges. Figure 5.5(b) shows an example of an infeasible cut, where $p_i \in \mathcal{T}$ and $q_{i+1} \in \mathcal{S}$ resulting in $z_p^i z_q^{i+1} = \{10\}$. Since edge (q_{i+1}, p_i)

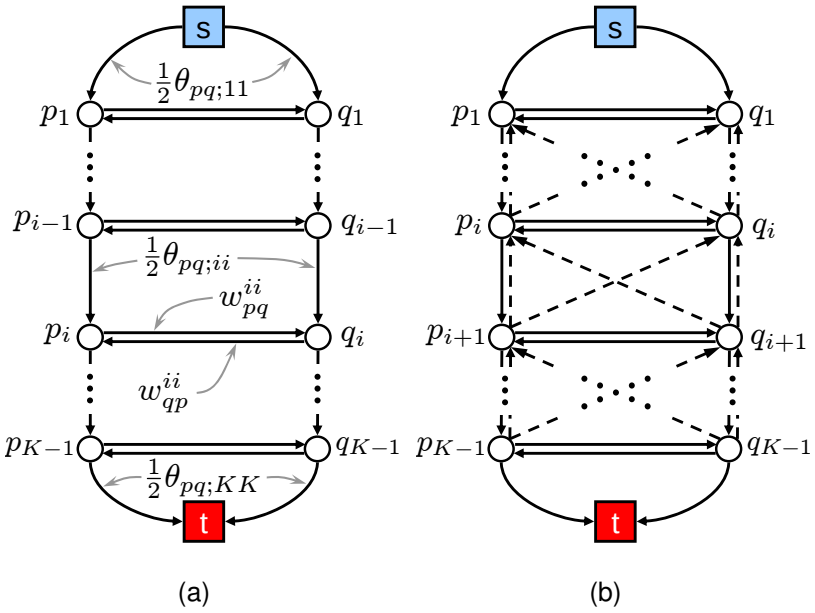


Figure 5.6. Graph construction for the minimization of a multi-label variable pair with adjacent label constraint. (a) Edge weight assignments for the pairwise potential. (b) Final graph from additively combining the graphs in (a) and figure 5.5(a).

with infinite weight is in the cut, this cut has a very high cost. Note that even if the cut assign $q_j \in \mathcal{T},\ j > i+1$ or $p_j \in \mathcal{S},\ j < i$, edge (q_{i+1}, p_i) will still be in the cut.

Up to this point, we have not addressed the portion of the graph construction that is necessary to account for the costs $\theta_{pq;ij}$. The constrained edges (q_{i+1}, p_i) and (p_{i+1}, q_i) ensure that the first summation in equation (5.25) will be zero for all feasible cuts on the graph. The only remaining task is to add the necessary edges to minimize the second summation, *i.e.* the term involving α_{ii}. Subsequently, we assume that the pairwise potential $\theta_{pq;ij}$ is submodular with respect to all adjacent label pairs $\{y_p = i, y_q = i+1\}$ so that $\alpha_{ii} \leq 0$. In figure 5.6(a), we show one possible edge weight assignment scheme,

but there are other equivalent constructions, *e.g.* see [65, 61], and their reparameterizations [64], that can also be used. The weights for edges (p_i, q_i) and (q_i, p_i) in the figure are, respectively,

$$w_{pq}^{ii} = \theta_{pq;(i+1)i} - \frac{1}{2}\Big(\theta_{pq;ii} + \theta_{pq;(i+1)(i+1)}\Big) \tag{5.27a}$$

$$w_{qp}^{ii} = \theta_{pq;i(i+1)} - \frac{1}{2}\Big(\theta_{pq;ii} + \theta_{pq;(i+1)(i+1)}\Big). \tag{5.27b}$$

Although the graph shown in figure 5.6(a) may not be the most compact construction, it provides a straightforward and intuitive representation for encoding the energy in equation (5.1).

Utilizing the additive property of graphs [65], the overall graph structure shown in figure 5.6 is produced by combining the graphs in figures 5.5(a) and 5.6(a), where the weights of directed edges linking the same nodes are added. The final graph has $(K-1)\cdot N+2$ nodes (including the terminals s and t), which is the same as the graph in figure 5.4(b). However the number of edges is $(K-1)\cdot(M_n+M_t)+(K-2)\cdot M_n$, which is on the order of $\mathcal{O}(N \cdot K)$. This is a significant reduction from the number of edges in the graph in figure 5.4(b). Note that M_n depends on the neighborhood connectivity.

Table 5.1 summarizes the weight assignments for the edges in the final graph. Note that it is possible for edges (p_i, q_i) and (q_i, p_i) to have negative weights. However, the reparameterization techniques in [61, 64] can be used to transform the graph so that these edges will have nonnegative weights. Moreover, in this work we use a pairwise

Table 5.1. Edge weight assignments for label adjacency constraint graph.

Edge	Weight
(s, p_1)	$\theta_{p;1} + \frac{1}{2}\theta_{pq;11}$
(p_i, p_{i+1})	$\theta_{p;i+1} + \frac{1}{2}\theta_{pq;(i+1)(i+1)}$
(p_{K-1}, t)	$\theta_{p;K} + \frac{1}{2}\theta_{pq;KK}$
(p_{i+1}, p_i)	∞
(p_i, q_i)	$\theta_{pq;(i+1)i} - \frac{1}{2}\Big(\theta_{pq;ii} + \theta_{pq;(i+1)(i+1)}\Big)$
(q_i, p_i)	$\theta_{pq;i(i+1)} - \frac{1}{2}\Big(\theta_{pq;ii} + \theta_{pq;(i+1)(i+1)}\Big)$
(p_{i+1}, q_i)	∞
(q_{i+1}, p_i)	∞

potential where $\theta_{pq,ii} = 0,\ \forall i \in \mathcal{L}$, and hence these edge weights are always nonnegative. The final graph construction in figure 5.6(b) allows the exact minimization of the energy in equation (5.1), subject to the LAC (5.2), in polynomial time using st-mincut techniques.

5.5 Experiments

In this section, we first describe the unary and pairwise clique potentials and the set of image features that are used for segmentation. Next we perform several experiments to validate the LAC and to demonstrate the shortcomings of using the Ishikawa algorithm [51] for nested layer segmentation. Then we present the results on 2D and 3D image data. All experiments are conducted using a PC with 2.16GHz Intel Core Duo, 2GB

RAM. We use the MATLAB mex interface to run the maxflow algorithm of Boykov and Kolmogorov [9] written in C++ and is available online (http://vision.csd.uwo.ca/code/). After user input, the density estimation takes approximately 1 to 5 seconds depending on the number of feature dimension and the training sample size. The maxflow step takes approximately 0.5 to 5 seconds depending on the image size and the number of labels used. The runtime for the Ishikawa method is approximately twice as long.

5.5.1 Clique Potentials

Let $x_p \in \mathbb{R}^d$ be a d-dimensional feature vector at location p. We use the following cost for the unary potential in equation (5.4):

$$\theta_{p,i} = -\log \Pr(x_p | y_p = i), \ \forall i \in \mathcal{L}. \tag{5.28}$$

This is simply the likelihood of observing x_p given that the label $y_p = i$. This cost favors the class label that best explains the observation x_p. The unary cost is often referred to as the data association potential [69].

We use the kernel density estimate to calculate the likelihood of an observation given a sample set of training data. Let the set of training observations be denoted as $\mathbf{x}_i = \{x_{1,i}, x_{2,i}, \ldots, x_{n_i,i}\}$, where the observations in $\mathbf{x}_i$ have associated label $i \in \mathcal{L}$. The kernel density estimate is

$$\Pr(x_p | y_p = i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|x_p - x_{j,i}\|^2}{2\sigma^2} \right). \tag{5.29}$$

In all experiments, the bandwidth parameter σ is set to $\sqrt{2d}/10$ for $x_p \in [0,1]^d$ [75], and we use the Fast Gauss Transform [121, 28] to efficiently compute equation (5.29).

For the pairwise potential, we use the function

$$V_{pq}(y_p, y_q) = g(x_p, x_q) \cdot |y_p - y_q|^k \tag{5.30}$$

for $k \to \infty$ so that when the label change $|y_p - y_q| > 1$, a very large penalty (infinity) is incurred. The data dependent function $g(x_p, x_q)$ is [8]

$$g(x_p, x_q) = \frac{1}{|p-q|}\left(\lambda_{\text{min}} + \lambda_x \cdot \exp\left(-\frac{\|x_p - x_q\|^2}{2\sigma_x^2}\right)\right) \tag{5.31}$$

and acts to penalize pairwise label changes by an amount dependent on the difference between the features x_p and x_q (plus a constant λ_{min}). Here, $|p-q|$ is the Euclidian distance between pixel p and q, which is not a constant when the neighborhood connectivity is 8 or greater. The parameter λ_{min} ensures that there is some minimum penalty for a label difference. For the experiments, both the 8- and 16-neighbor connectivity (conn) are used, and except for the slightly longer runtime for the latter connectivity, there is little noticeable difference between the results.

With equation (5.30), no cost is incurred when the labels $\{y_p, y_q\}$ are the same, a cost $g(x_p, x_q)$ is incurred when the label difference is one, and an infinite cost is incurred

when the label difference is greater than one. In summary, the penalty $\theta_{pq,ij}$ is

$$\theta_{pq,ij} = \begin{cases} 0 & \text{if } y_p = y_q \\ g(x_p, x_q) & \text{if } |y_p - y_q| = 1 \\ \infty & \text{otherwise.} \end{cases} \tag{5.32}$$

In practice, we only compute the cost $\theta_{pq,ij}$ when $|y_p - y_q| = 1$ and set the weight for the constraint edges in the graph to some large value, *e.g.* 10^6. For all experiments, the parameter σ_x, which controls the contrast sensitivity, is set to be the square root of the average square norm

$$\sigma_x = \sqrt{\frac{1}{|\mathcal{E}_{RF}|} \sum_{(p,q)\in\mathcal{E}_{RF}} \|x_p - x_q\|^2}, \tag{5.33}$$

and λ_{min} is usually set as

$$\lambda_{\text{min}} = \frac{1}{|\mathcal{E}_{RF}|} \sum_{(p,q)\in\mathcal{E}_{RF}} \exp\left(-\frac{\|x_p - x_q\|^2}{2\sigma_x^2}\right). \tag{5.34}$$

Here $\mathcal{E}_{RF}$ is the set of edges in the random field model. There are several occasions when it becomes necessary to manually set a value for λ_{min}. For such instances, we indicate this value along with the accompany results.

5.5.2 Image Features

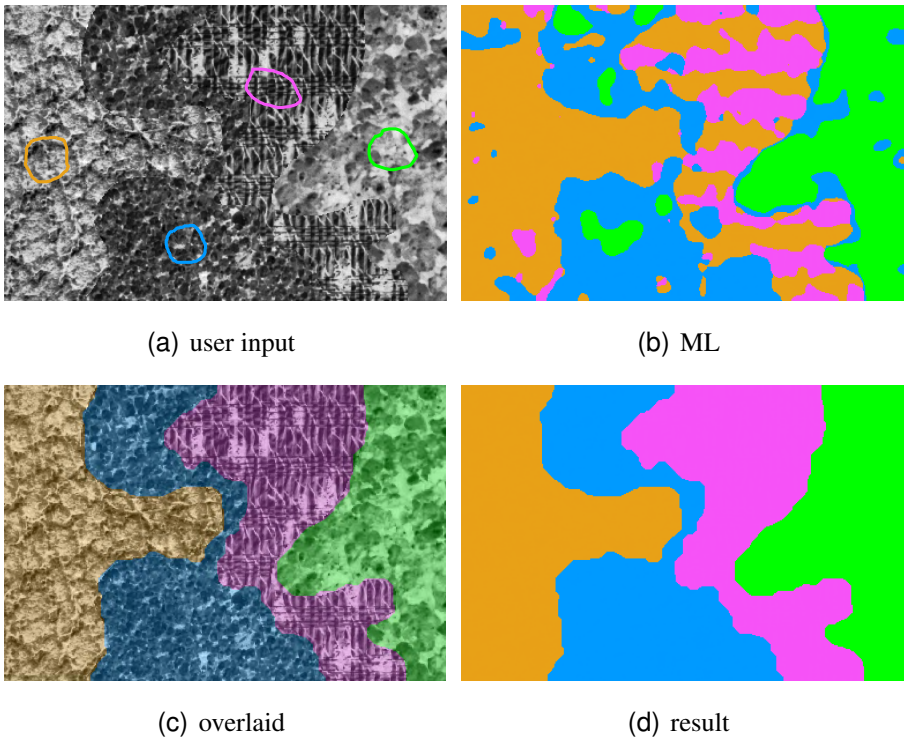
We use a combination grayscale and color values, as well as various texture descriptors, for the image features in our experiments. RGB color images are converted into

the three dimensional Luv colorspace, which we denote $[I\ I_u\ I_v]$. There are many descriptors that are popularly used in the vision community for texture discrimination, such as Gabor features [78]. However, features based on Gabor filter outputs are not the most suitable for our application because they are typically high dimensional. As we show later, the training sample sizes are typically small, and consequently high dimensional data presents a serious problem for obtaining accurate density estimation.

To capture the orientation of the texture, we use the diffusion based texture features proposed in [92]. This feature is computed from the joint nonlinear diffusion of the structure tensor components, resulting in a three dimensional vector $[I_{xx}\ I_{xy}\ I_{yy}]$. Then to characterize the scale of the texture, we use the TV flow based local scale measure I_s, a one dimensional feature, presented in [12]. The texture descriptor used in our experiments is only four dimensional, but it has been shown to perform comparably with a 12-dimensional Gabor feature [13].

5.5.3 Segmentation Workflow

For a given image, the segmentation begins with interactive input from the user. One or more exemplar regions from each layer are selectively marked, and their nesting order must be indicated by the user. Using the pixel features in the exemplar regions for training, the density estimate for the entire image is then computed. Next, the graph structure is built and the maxflow algorithm is used to find the optimal labeling.



(a) user input (b) ML

(c) overlaid (d) result

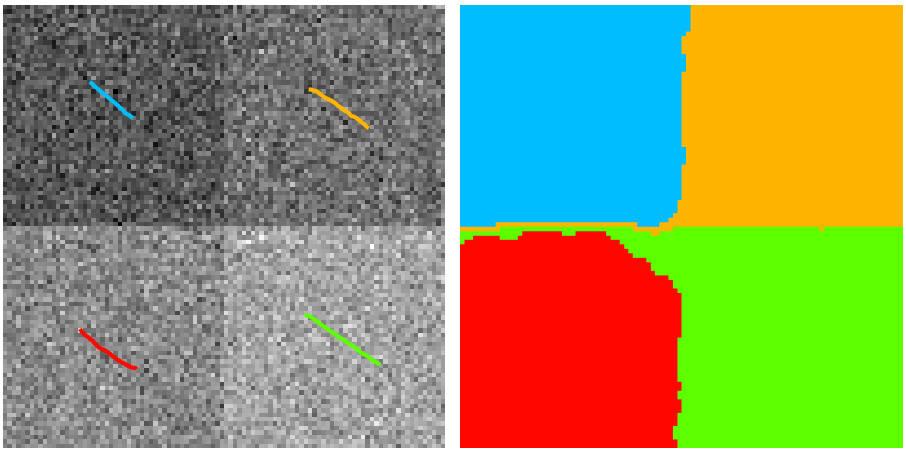
Figure 5.7. Nested layer segmentation of a texture image. Using 4 labels, the user selects exemplar region(s) for each label in order of layer nesting, either right-to-left or left-to-right for this example. ML indicates the maximum likelihood classification from the density estimation using the exemplar region features. The result (d) is overlaid or superimposed on top of the original image (c). The texture feature was used, conn = 8, and $\lambda_{pq} = 1$.

Figure 5.7 illustrates the segmentation workflow, where only the texture features are used. Figure 5.7(a) shows the exemplar regions selected by the user, with the ordering specified from either left-to-right or right-to-left. Figure 5.7(b) shows the maximum likelihood classification after the density estimate. Note that the combination of features and density estimation does not necessarily provide an accurate indication of the layer labels. Nonetheless the LAC is powerful enough to correct for this shortcoming.

Figure 5.7(c) shows the result of our algorithm overlaid on top of the original image, and figure 5.7(d) shows the result alone.

5.5.4 Label Adjacency Validation

It is entirely possible that the natural layer nesting of an image can cause any segmentation algorithm to return a solution with the same nesting relationship. To demonstrate that our method does indeed enforce the nested relationship, we test the algorithm on the synthetic image shown in figure 5.8. The regions in this image do not have a nested relationship, and there is a junction point where the four regions meet. Hence, it is impossible to correctly assign four separate labels, *e.g.* $\mathcal{L} = \{1, 2, 3, 4\}$ to these



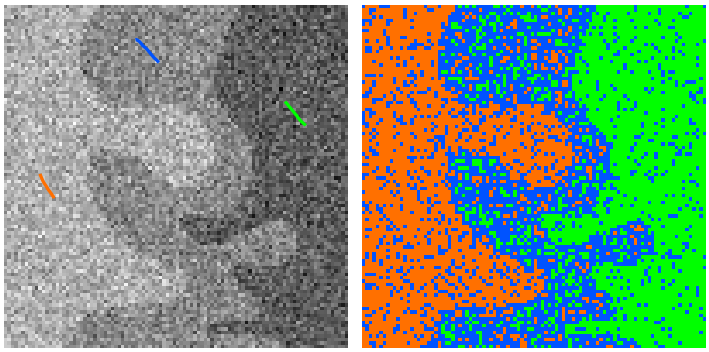
(a) user input (b) result

Figure 5.8. Label adjacency validation. Proceeding in a clockwise fashion, the four regions are assigned labels $\mathcal{L} = \{1, 2, 3, 4\}$ starting with the top left and ending with the bottom left quadrant. The two thin strips of pixels with label 2 and 3 in the result prevents label changes of greater than one.

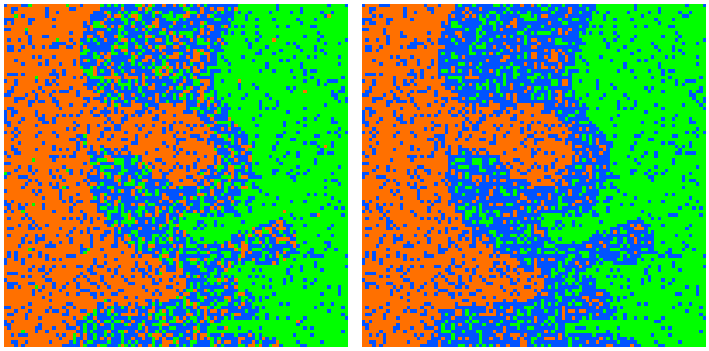
four regions without violating the LAC. However, enforcing the LAC will lead to the incorrect segmentation. We choose to assign the input labels in a clockwise direction starting with label 1 for the top left quadrant. The result using our algorithm show that there are two thin strips of pixels labeled 2 and 3 that are wedged between the regions 1 and 4 in order to maintain the LAC. As this example illustrates, our algorithm will enforce the LAC in the segmentation of any image. However, it will return undesirable results if used for the wrong application, such as segmentation of images without region nesting.

5.5.5 Comparison with Other Algorithms

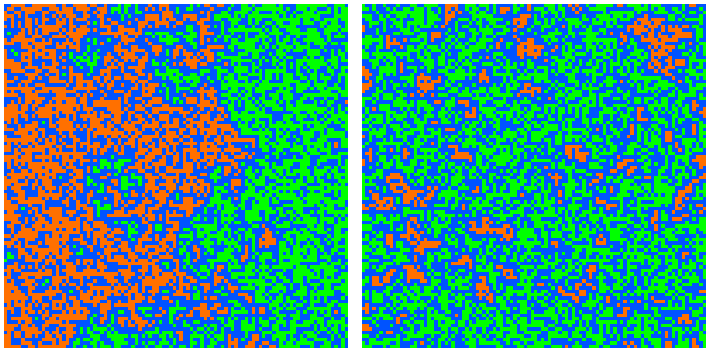
As stated in section 5.2, the graph construction of Ishikawa [51] can also be used, in theory, to solve the nested layer segmentation problem. Besides requiring a larger size graph compared to our method, in practice the Ishikawa algorithm is subject to numerical overflow problems when enforcing the LAC. We demonstrate the overflow problem on the synthetic image in figure 5.9. Using the pairwise potential in equation (5.30), we ran the Ishikawa algorithm for $k = \{1, 2, \ldots, 35\}$, $\lambda_{pq} = 0.01$ and $\lambda_{\text{min}} = 0$ and compare the results to our method using the same parameters. For each run, both the maximum flow value and the CRF energy are tabulated. However, since the energy is dependent on the value of the label adjacency penalty, we set the cost of having a pairwise violation to $10^6 \times g(x_p, x_q)$ and compute the energy based on this cost. The



(a) user input (b) our result

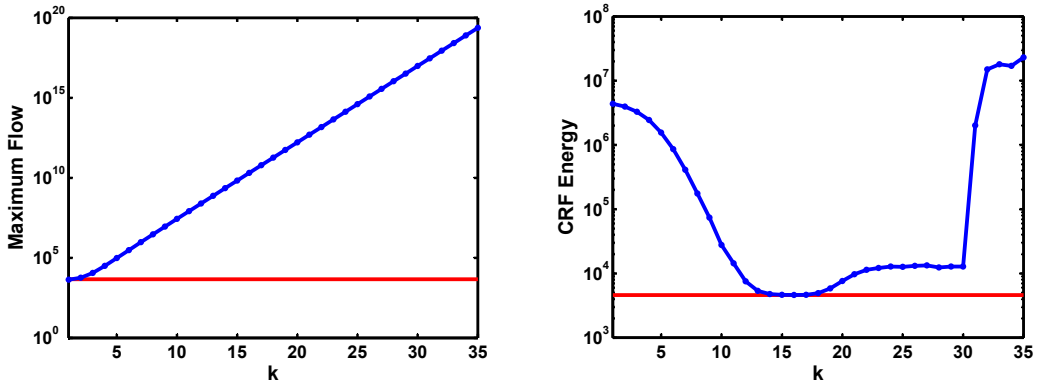


(c) $k = 5$ (d) $k = 16$



(e) $k = 20$ (f) $k = 25$

Figure 5.9. Comparison with Ishikawa method. For low values of k, the results violate the LAC. As the value of k increases, the Ishikawa method encounters numerical errors and produces increasingly less accurate results.



(a) Maximum flow. (b) CRF energy.

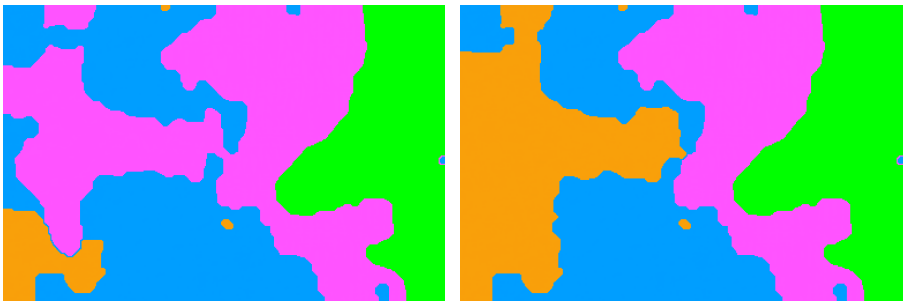
Figure 5.10. Plots of the maximum flow and energy as a function of k for the Ishikawa results (blue curves). The red curves show the maximum flow and the energy computed for our algorithm using the same parameter settings. For smaller k, the Ishikawa method violates the LAC resulting in large energies. As k increases to around 16, these violations are reduced resulting in smaller energy values, but they are still higher than the ours. For larger k values, the algorithm encounters numerical overflow errors and produces results that have very high energies.

result of our method is shown in figure 5.9(b), and the results of the Ishikawa method for four values of k are shown in the last two rows of figure 5.9.

Figure 5.10 shows the plot of the maximum flow and energy as a function of k for the Ishikawa results. The red lines in both plots show the maximum flow and the energy computed for our algorithm using the same parameter settings. Notice that for sufficiently small k, the segmentation violates the LAC resulting in a large energy. The result in figure 5.9(c) for $k = 5$ shows numerous instances of this violation. As k increases to around 16, these violations are reduced resulting in smaller energy values. Figure 5.9(d) shows the Ishikawa result for $k = 16$, where the energy is lowest. There

are no label violation and the result is similar to ours, the Ishikawa energy is still slightly higher compared to our result. Also, note that the maximum flow value continues to increase, indicating the many edges with large weights are in the cut. For sufficiently large values of k, the algorithm encounters numerical overflow problems when computing the maximum flow, and the results are no longer accurate. Figures 5.9(e) and 5.9(f) show the result for $k = 20$ and $k = 25$, respectively. At these values of k, the energy begins to increase and the segmentation becomes inaccurate.

We also compare our algorithm to the α-expansion and $\alpha\beta$-swap algorithms [10], which are state of the art algorithms used to obtain approximate solutions to the multi-label CRF problem. These algorithms iteratively makes labeling moves at each iteration to decrease the CRF energy. For more details, refer to chapter 2. As noted by Liu *et al*. [74], both the expansion and swap algorithms are more likely to get stuck in local minima when ordering constraints are used, and we observed this behavior frequently in our experiments. We test these two algorithms using the same parameter settings on the image in figure 5.7 and set the adjacency constraint penalty to 10^6. Since both algorithms compute the solution iteratively starting from a random initial estimate, the final solutions are often different from one another. The result of two separate runs for the swap algorithm is shown in figure 5.11. The results for the expansion algorithm are similar and we do not present them here. Although the results do not violate the LAC, the CRF energies for these labelings are much higher than the one shown in



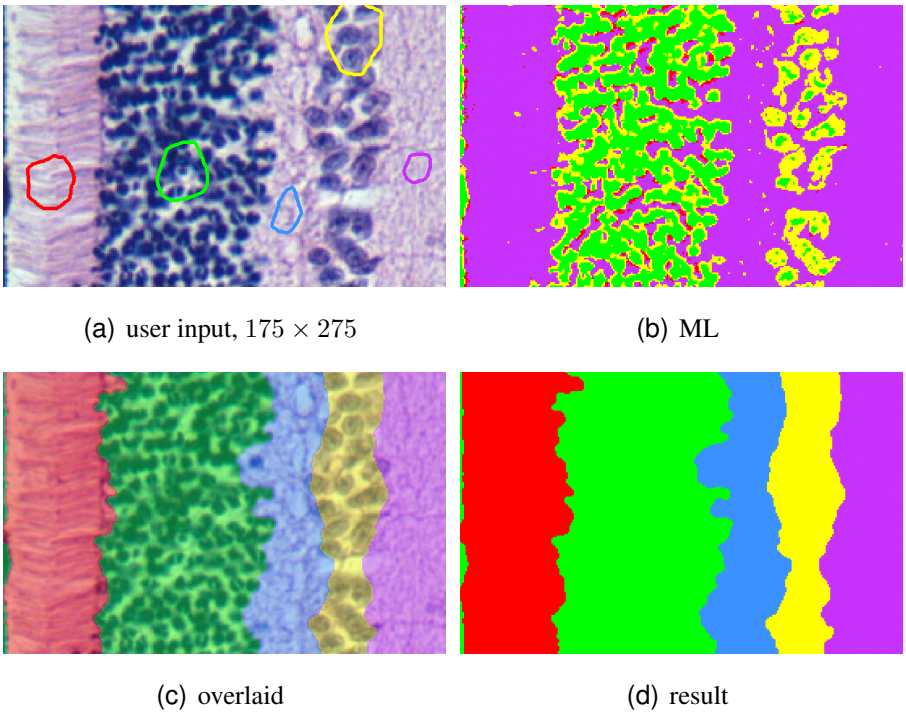
(a) Maximum flow. (b) CRF energy.

Figure 5.11. Results from two separate runs of the $\alpha\beta$-swap algorithm with LAC. Although there are no constraint violation, the CRF energies for these labelings are much higher than the one shown in figure 5.7(d) computed using our method.

figure 5.7(d) computed using our method , and these algorithms take approximately 30 times longer to run for this example.

5.5.6 Results

We show the result of our method on a set of 2D biological images. For these images, the parameter λ_{mim} is computed automatically from equation (5.34). Both the 8- and 16-neighbor connectivity (conn) are used, and except for the slightly longer runtime for the latter connectivity, there is little noticeable difference between the results. Figure 5.12 shows the segmentation of a retina cross section using Luv color feature. Notice that this color feature does not provide much discriminating power for several of the layers. This is reflected in the maximum likelihood classification (ML), where the pink label seemed better suited to characterize the red and blue regions. However, our method



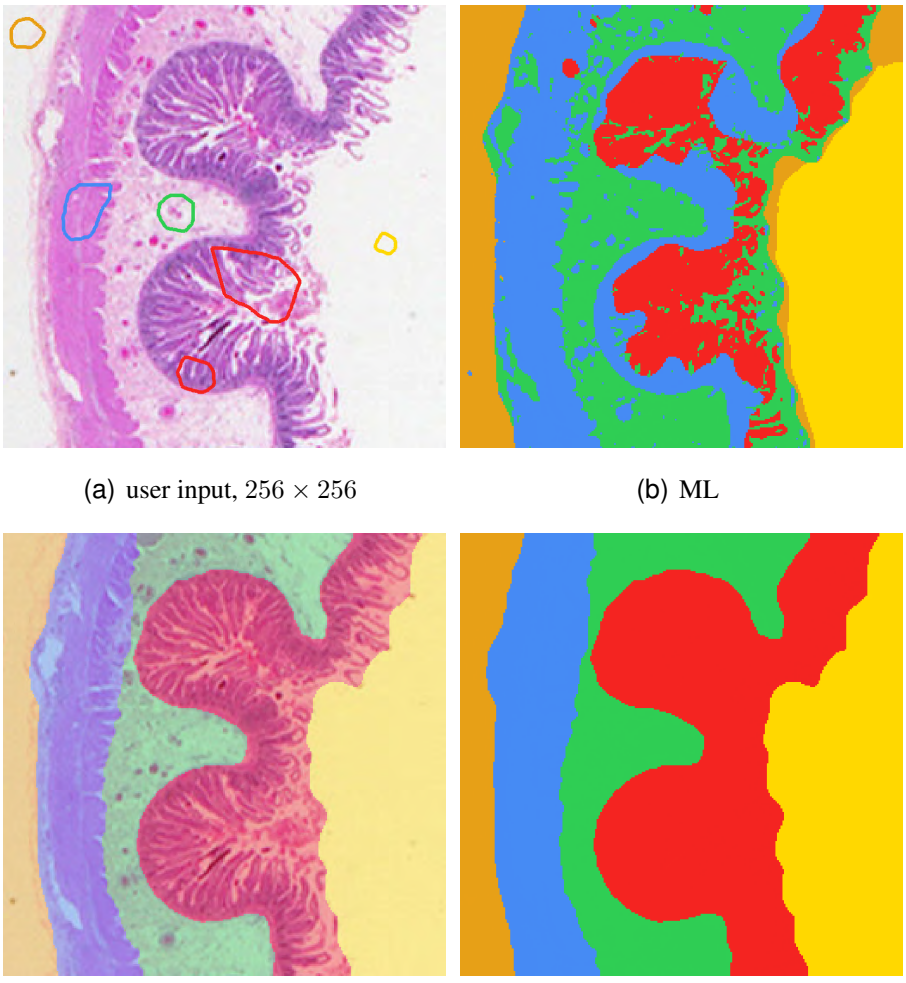
(a) user input, 175×275 (b) ML

(c) overlaid (d) result

Figure 5.12. Segmentation of a retinal cross section with 5 labels. Parameters: conn $= 16$, $\lambda_{pq} = 1$, Luv color feature.

successfully segmented all 5 regions. Texture feature could have also been used, but we wanted to demonstrate the robustness of our method in coping with poor density estimation. The image in figure 5.12 was provided by Dr. Nedim C. Buyukmihci, V.M.D., Emeritus Professor of Veterinary Medicine at University of California, Davis.

Figure 5.13 shows the segmentation of a longitudinal section of the jejunum using our texture feature and 5 labels. Note that the left and right regions (label 1 and 5) are both considered the background. However, due to the LAC, these two regions must be labeled as separate. The texture in the red labeled region exhibit strong inhomogeneity,

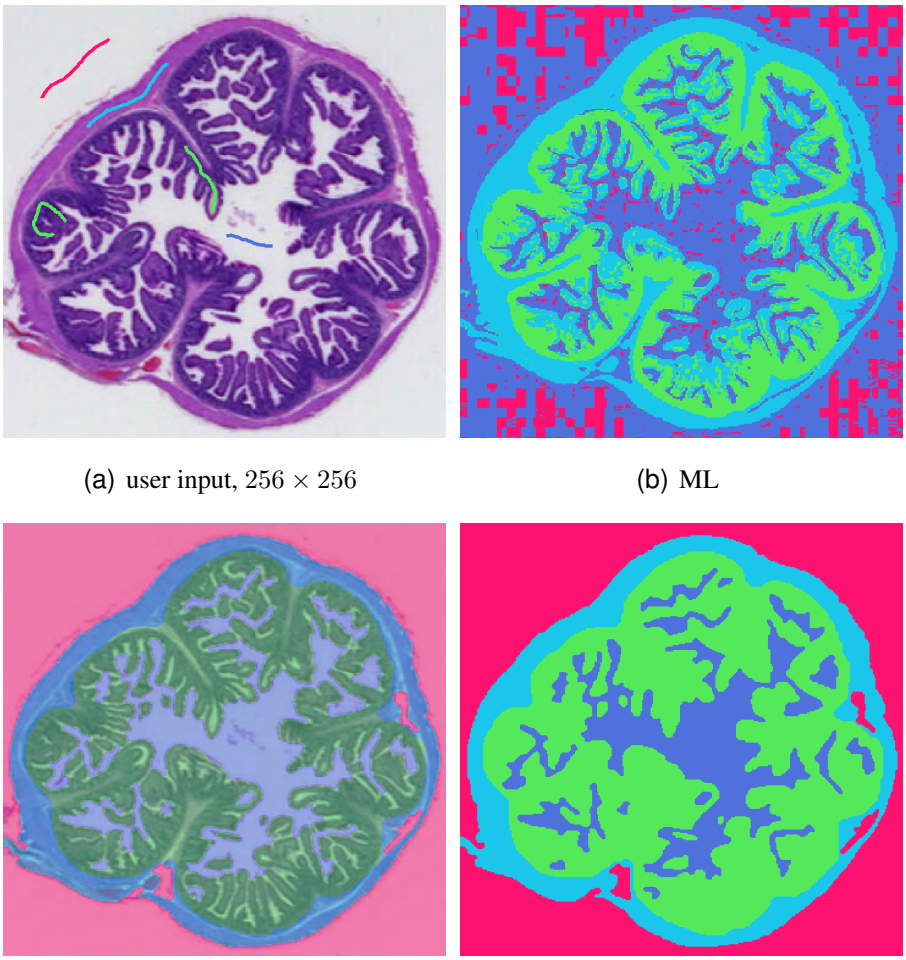


(a) user input, 256×256 (b) ML

(c) overlaid (d) result

Figure 5.13. Segmentation of a longitudinal section of the jejunum with 5 labels. Parameters: conn $= 16$, $\lambda_{pq} = 0.5$, texture feature.

and two sample regions were necessary to capture this variation. Figure 5.14 shows the segmentation of a cross section of the jejunum. Here, the Luv color features are used. Note that the input markings do not select regions, but simply select a small set of training pixels for each label. Both the images in figures 5.13 and 5.14 were provided

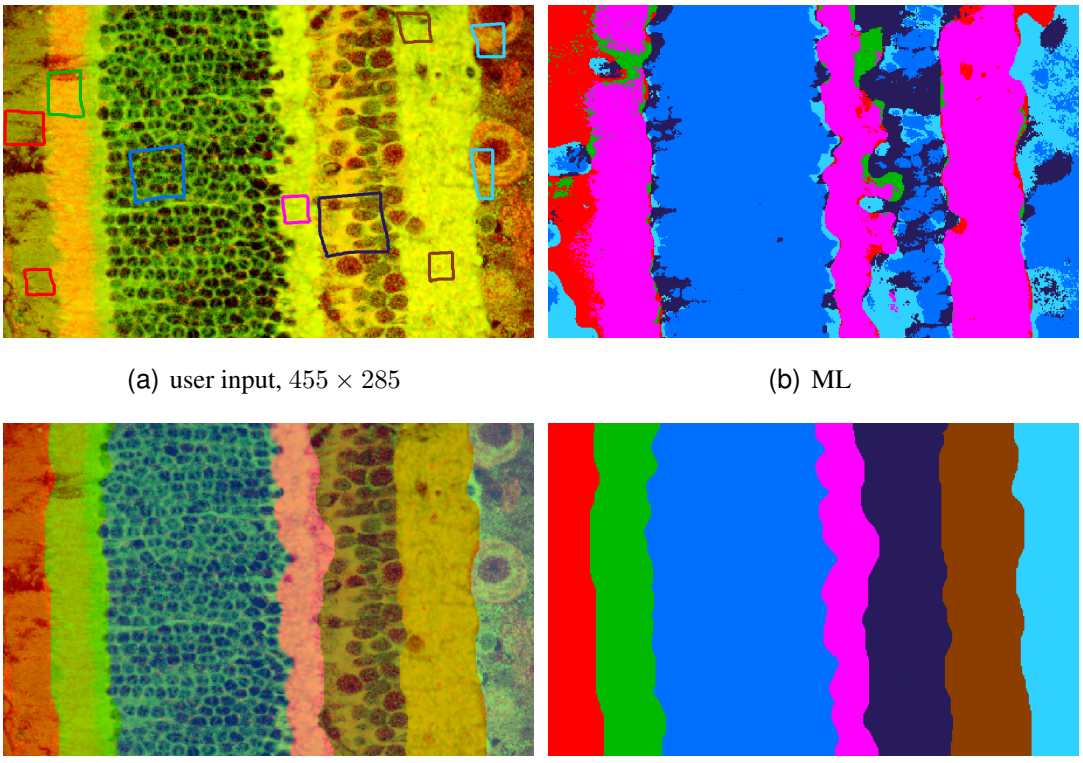


(a) user input, 256×256 (b) ML

(c) overlaid (d) result

Figure 5.14. Segmentation of a cross-section of the jejunum with 4 labels. Parameters: conn $= 16$, $\lambda_{pq} = 0.1$, Luv color feature.

by Dr. Darl Ray Swartz for the Department of Animal Science, Purdue University, West Lafayette, IN. Figure 5.15 shows the segmentation of an immunofluorescence image of a retinal cross section using 7 labels. The highly inhomogeneous textures in several of the regions require large exemplar regions for training. This image was provided by Dr.



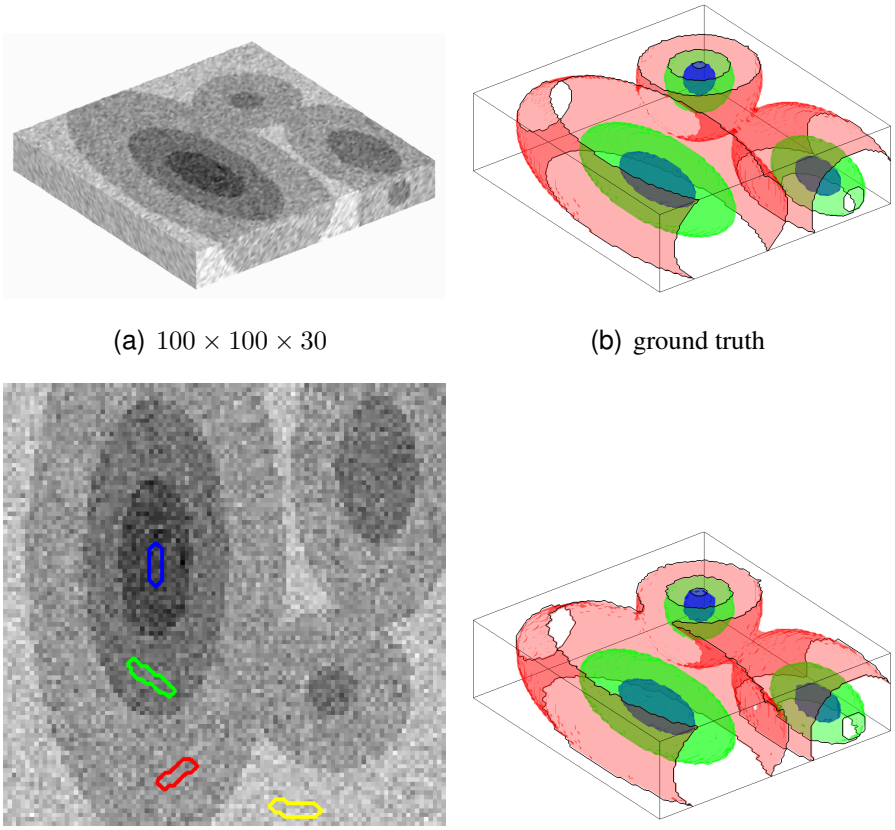
(a) user input, 455×285 (b) ML

(c) overlaid (d) result

Figure 5.15. Segmentation of immunofluorescence image of retinal cross section with 7 labels. Parameters: conn $= 16$, $\lambda_{pq} = 0.1$, texture feature.

Geoff Lewis from the Neuroscience Research Institute, University of California, Santa Barbara.

We also tested our algorithm on 3D data. Figure 5.16 shows the segmentation of a 3D phantom. A section of the 3D data is shown in figure 5.16(a) and the ground truth is shown in figure 5.16(b). The middle slice was selected for user input and is shown in figure 5.16(c). The final result isosurfaces are shown in figure 5.16(d). With some slight



(a) $100 \times 100 \times 30$ (b) ground truth

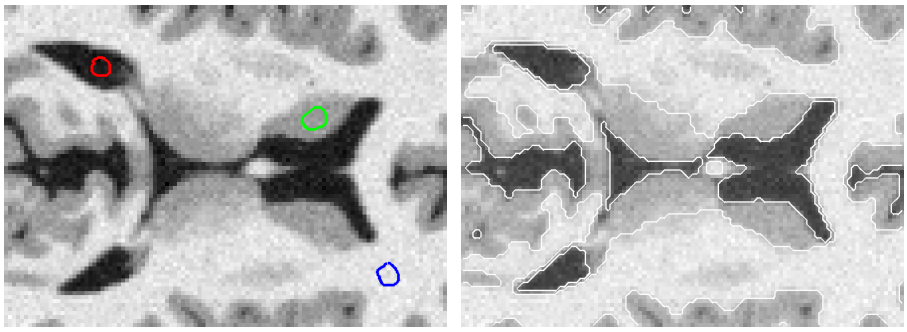
(c) user input (d) result

Figure 5.16. Segmentation of a 3D phantom. Parameters: conn $= 8$, $\lambda_{pq} = 0.1$, grayscale feature.

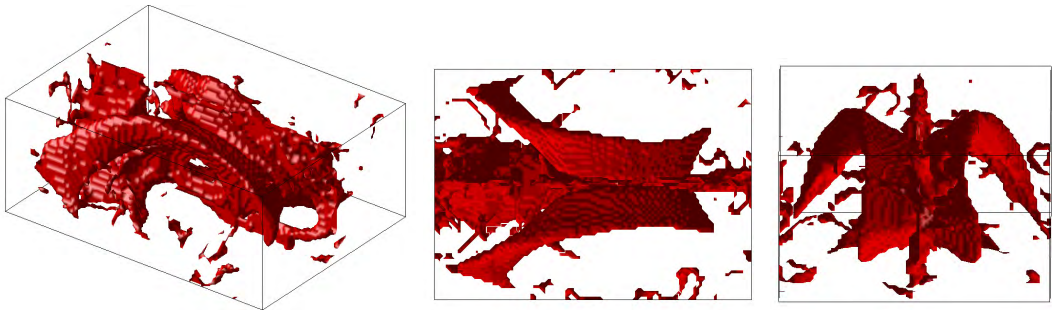
errors, the result correctly captured the structure of the phantom. The segmentation of this phantom took less than 3 seconds.

Finally, we tested our method on simulated 3D MRI T1 data generated from the brain phantom publicly available from the BrainWeb [21]. To make the data size more manageable, the volume was cropped to an $80 \times 110 \times 45$ regions surrounding the ventricles. To segment the volume, we used three labels: cerebral spinal fluid (CSF), gray

matter, and white matter. The middle slice was used for user input (see figure 5.17(a)) and only the grayscale value is used for density estimation. Figure 5.17(b) shows the result at that slice. Notice that near the dark ventricles, there is a thin layer of pixels classified as gray matter that separates the white matter from touching the CSF. The LAC helps to preserve this anatomically correct nesting relationship among the three classes. The total time for graph cuts is less than 5 seconds.



(a) input (b) result



(c) Isosurface of the ventricles.

Figure 5.17. Segmentation of 3D MRI data. Parameters: size = conn $= 8$, $\lambda_{pq} = 0.5$, feature used = grayscale value.

5.6 Conclusion

In this chapter, we presented an algorithm for nested layer segmentation and showed that the additional constraint placed on adjacent labels allows the problem to be solved exactly and efficiently. Compared to the graph cut method of Ishikawa [51], whose formulation also guarantees the globally optimal solution, our algorithm is more efficient in that it significantly reduces the graph size and does not encounter numerical overflow errors. Although our method, as far as we are aware, offers the simplest graph construction to solve the problem, it is still limited to relatively small data sizes, especially in 3D. This can be very problematic since biomedical datasets are typically very large. Fortunately, there are research efforts in developing maxflow algorithms for large vision graphs [27] and methods to run graph cuts on large graphs using GPUs [98].

There are several future directions that we can explore. We are beginning incorporate the higher order $\mathcal{P}^n$ Potts potential into the proposed method to improve its ability to capture larger spatial dependencies among pixel groups. This would improve the segmentation of regions where the texture is inhomogeneous and has large scale. These types of textures are difficult to characterize using existing texture descriptors. A second direction is to investigate the performance of our algorithm for hierarchical segmentation. For example, the regions of the image in figure 5.8 are not globally nested, but when considering a subregion of the image, *e.g.* the top or bottom half,

the regions are indeed nested. Our algorithm can be used to correctly segment the image in a hierarchical manner by dividing the image into subregions and computing the solution for each subregion. In this case, the globally optimal solution is no longer guaranteed, but with respect to the subregions, the solution is still exact. Finally, there are situations in biomedical imaging where the layers are nested, but the nesting is not absolute and certain normally nonadjacent layers can still touch with some probability. It would be worthwhile to investigate whether our method can be modified to accommodate these situations while still have strong optimality guarantees. We discuss these future research directions in more detail in the final chapter.

Chapter 6

Towards Bioimage Analysis

Image segmentation is a challenging problem and in itself generates much interest in the research community. However, the driving force that motivates much of image segmentation research is, arguably, the end applications. This application driven paradigm is nowhere more apparent than in biomedical image analysis, where segmentation plays an increasingly important role in the analysis pipeline. Its capability for efficient information extraction makes segmentation a vital intermediary between the image acquisition process and the knowledge formation process.

In the first part of this chapter, we present an interactive editing algorithm that brings the segmentation process closer to the information extraction step. This algorithm allows the user to quickly correct errors from a previous segmentation instead of having to rerun the segmentation entirely or fine-tune its parameters. Following the first sec-

tion, we present two applications that illustrate the utility of image segmentation in bioimage analysis. The first analysis relies on a segmentation result to measure the layer thickness and nuclear density of the outer nuclear layer (ONL) of the retina. The second analysis uses the retina layer segmentation to compute profiles of protein distributions across the retinal layers. Both of these analysis provide quantitative measures of anatomical changes occurring after retina detachment and reattachment, and offers quantitative insights to corroborate qualitative observations.

6.1 Interactive Editing

Whereas the majority of segmentation algorithms are designed to be broadly applicable to a wide range of vision tasks, those that are too general become less desirable for biomedical applications since these applications often demand highly specific and accurate results. Consequently, biomedical researchers tend to devote a huge amount of time and effort fine-tuning the segmentation algorithms to suit their specific needs. However up to a certain point, the tradeoff between the time expended and the accuracy gained begins to diminish, and the researcher is forced to either accept the nearly correct result or to manually perform the segmentation. To a degree, we can expect any segmentation algorithm to produce some unintended results because the algorithm parameters may be difficult to tune for an entire dataset or because the energy formu-

lation fails to sufficiently capture the domain knowledge. Allowing the user to edit the segmentation results can be very beneficial and less time consuming compared to parameter tuning and learning or reformulating the energy functional.

The editing algorithm should meet several criteria in order for it to be useful. First, the editing should be fast so that the user can make repeated edits quickly. Second, the method should produce intuitive results that take advantage of image information. Finally, the method should only act locally near editing marks so that the previous segmentation result is not drastically altered, since it is assumed to be nearly correct. There are several segmentation methods that allow for user interaction and editing during the segmentation process [55, 29, 81]. However these methods cannot be used to correct a previous segmentation result, especially one that was not computed using the same algorithm. The are also graph cut methods that allow for editing after the segmentation [7, 44]. However, these methods either modify the original graph cut solution based on editing marks and rerun graph cuts, or they do not make full use of the image information during editing.

Similar to [44], our editing algorithm utilizes the previous segmentation output (pre-segmentation), since it is assumed that that result is nearly correct. Secondly, only pixels that change their labels during editing are penalized. Contrary to the formulation in [44], our relabeling penalty is greater for label changes that are farther from the edit marks, since these pixels are less likely to be considered for relabeling in the

user's editing thought process. Furthermore, our proximity measure is a function of the geodesic distance on the image intensity instead of the Euclidian distance used in [44]. The geodesic distance is a more intuitive measure because pixels separated by more membranes or edges are less likely to belong to the same object. Finally our algorithm deals with the general multi-label case and reduces to the formulation in [44] for the binary case. A preliminary version of our editing algorithm for the two-label case is presented in [115].

6.1.1 Edit Energy

Given an image $\mathbf{x} = \{x_p : p \in \mathcal{P}\}$ with presegmentation $\mathbf{y}' = \{y'_p : p \in \mathcal{P},\ y'_p \in \mathcal{L}\}$, the user begins the editing process by marking small subsets of pixels indicating the regions where corrections are desired. Here the set of image pixels is denoted $\mathcal{P}$, the set of pixels marked with label $i \in \mathcal{L}$ is denoted $\mathcal{M}_i$, and the set of all markings is denoted $\mathcal{M} = \{\mathcal{M}_i : i \in \mathcal{L}\}$. Using the Conditional Random Field (CRF) model (see chapter 2 for more details), we define the editing energy for a new segmentation or labeling $\mathbf{y} = \{y_p : p \in \mathcal{P}\}$ given the presegmentation $\mathbf{y}'$ and the edit marks $\mathcal{M}$ as

$$E(\mathbf{y}|\mathbf{y}', \mathcal{M}) = \sum_{p \in \mathcal{P}} V_p(y_p|y'_p, \mathcal{M}) + \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} V_{pq}(y_p, y_q). \tag{6.1}$$

In the CRF model, $\mathcal{N}_p$ is the set of neighbors of pixel p. The unary potential $V_p(y_p|y'_p)$ is dependent on both the presegmentation and the editing marks, while the pairwise

potential $V_{pq}(y_p, y_q)$ is only dependent on the image data. For simplicity, we not dot explicitly show the dependency of the CRF energy on the image $\mathbf{x}$, but it is understood to exist.

To enforce the smoothness of adjacent labels, we use the pairwise potential

$$V_{pq}(y_p, y_q) = g(p, q) \cdot f(y_p, y_q), \tag{6.2}$$

where

$$g(p, q) = \exp\left(-\frac{(x_p - x_q)^2}{2\sigma_x^2}\right)\frac{1}{|p - q|} \tag{6.3}$$

and $f(y_p, y_q)$ is some function of the pairwise labels. The pairwise potential penalizes label assignments $\{y_p, y_q\}$ for neighboring pixels $\{p, q\}$ by an amount $g(p, q) \cdot f(y_p, y_q)$. The unary cost is dependent on the presegmentation and user markings and is defined as

$$V_p(y_p|y'_p, \mathcal{M}) = \sum_{i\in\mathcal{L}} \sum_{j\in\mathcal{L}, j\neq i} \theta_{p;ji}(\mathcal{M}_i)\delta(y'_p, j)\delta(y_p, i), \tag{6.4}$$

and

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{otherwise.} \end{cases} \tag{6.5}$$

The function $\theta_{p;ji}(\mathcal{M}_i)$ is the cost of relabeling $y'_p = j$ to $y_p = i$ and is dependent on the the edit mark $\mathcal{M}_i$. Intuitively, the unary cost only penalizes label changes between the presegmentation and the new result, *i.e.* when $\delta(y'_p, j)\delta(y_p, i) = 1,\ i \neq j$. Since the

presegmentation is assumed to be nearly correct, this type of cost function encourages pixels to keep their original labels unless the change is specified by the user.

The label-change cost $\theta_{p;ji}(\mathcal{M}_i)$ is designed to penalize more heavily label changes that are farther from the user edit marks than closer ones, since more distant pixels are less likely to be considered for change in the user's editing thought process. Accordingly, we set the label-change cost to

$$\theta_{p;ji}(\mathcal{M}_i) = \lambda \cdot d_j(p, \mathcal{M}_i). \tag{6.6}$$

Here λ is an algorithm parameter that weights the importance of the presegmentation compare to the image data and is set to $\lambda = 0.01$ for all experiments. The function $d_j(p, \mathcal{M}_i)$ is the shortest geodesic distance of a pixel p with prelabel $y'_p = j$ to the a pixel in the edit mark set $\mathcal{M}_i$. The subscript j indicates that the geodesic function may differ for each region with prelabel j since the regions may have a different geodesic distance metrics. This choice of relabel penalty is very different from the one proposed by Grady and Funka-Lea [44] in two ways. First, their label-change cost for pixel p is a function of the Euclidian, not geodesic, distance from p to *all* the edit marks $\mathcal{M}$. Secondly, label changes that occur closer to the edit marks are penalized more, and not less as is the case in equation (6.6). This is somewhat counterintuitive since having such a penalty would discourage label changes near areas that require editing.

6.1.2 Two-Label Case

For a presegmentation with two labels $\mathcal{L} = \{1, 2\}$, we use the pairwise potential in equation (6.2) with $f(y_p, y_q) = |y_p - y_q|$, which is the same Potts potential used in chapters 3 and 4. Only pairwise label assignments that are different are penalized. The unary potential for the two-label case becomes

$$V_p(y_p|y'_p, \mathcal{M}) = \theta_{p;21}(\mathcal{M}_1)\delta(y'_p, 2)\delta(y_p, 1) + \theta_{p;12}(\mathcal{M}_2)\delta(y'_p, 1)\delta(y_p, 2). \tag{6.7}$$

Which is similar in form to the one proposed in [44]. In this work, we assume that d_j has the same distance metric for all labels $j \in \mathcal{L}$ and the subscript j is dropped. The geodesic distance d from pixel p to $\mathcal{M}_i$ is given by the solution of

$$|\nabla d(p, \mathcal{M}_i)|F = 1, \tag{6.8}$$

which can be computed efficiently using the Fast Marching algorithm [100]. Intuitively, equation (6.8) states that the gradient magnitude of the geodesic distance is inversely proportional to the speed F of the "image terrain."

Assuming that $\mathbf{x}$ is a grayscale image, we use $F = 1/(|\nabla \mathbf{x}| + \epsilon)$ for the speed term with ϵ a small constant. This speed term ensures that pixels separated from the edit marks by more edges (larger gradient) are geodesically farther than those separated by fewer edges. This formulation meets the locality criterion by confining the editing changes to areas near edit markings. There are cases where the gradient magnitude may not be a suitable, such as when the image region is textured. For such cases, we can use

the adaptive weighted distances proposed by Protiere and Sapiro [87]. However, in our experiments, the gradient magnitude proved to be adequate.

Since the pairwise potential $V_{pq}(y_p, y_q)$ is submodular (see chapter 2), the CRF energy can be minimized exactly with graph cuts. The graph construction details are given in chapter 2, and the weight assignments are given by

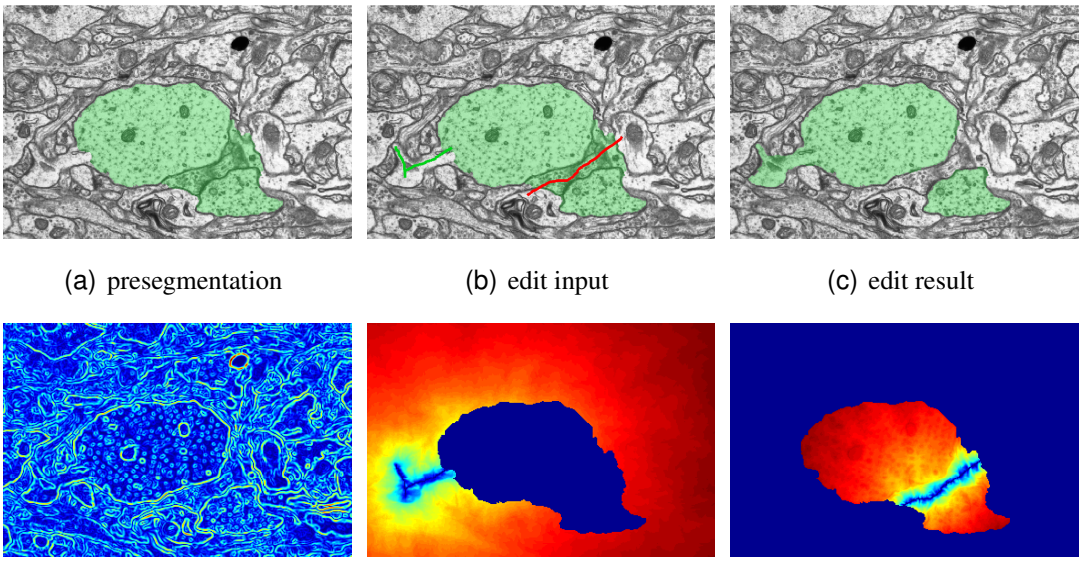
$$w_{sp} = V_p(y_p = 1|y'_p, \mathcal{M}_1) \tag{6.9a}$$

$$w_{pt} = V_p(y_p = 2|y'_p, \mathcal{M}_2) \tag{6.9b}$$

$$w_{pq} = g(p, q). \tag{6.9c}$$

With this convention, if edge (s, p) is in the cut, then $y_p = 1$, while $y_p = 2$ if edge (p, t) is in the cut.

Figure 6.1 shows the editing process for an EM image. For the presegmentation in figure 6.1(a), the green pixels have prelabel $y_p = 1$ and the non-green pixels have prelabel $y_p = 2$. The user then relabels a set of pixels as object (green) and background (red) as shown in figure 6.1(b). Figure 6.1(c) shows the result after editing. The gradient magnitude used to compute the speed F is shown in figure 6.1(d), and the resulting costs $V_p(y_p = 1|y'_p, \mathcal{M}_1)$ and $V_p(y_p = 2|y'_p, \mathcal{M}_2)$ are shown in figures 6.1(e) and 6.1(f), respectively. For better visualization, the logarithm of these costs are shown. Notice that the cost $V_p(y_p = i|y'_p, \mathcal{M}_i)$ is zero (blue region) where the presegmentation label equals i. This indicates that if there is no label change, then no cost is incurred. Notice

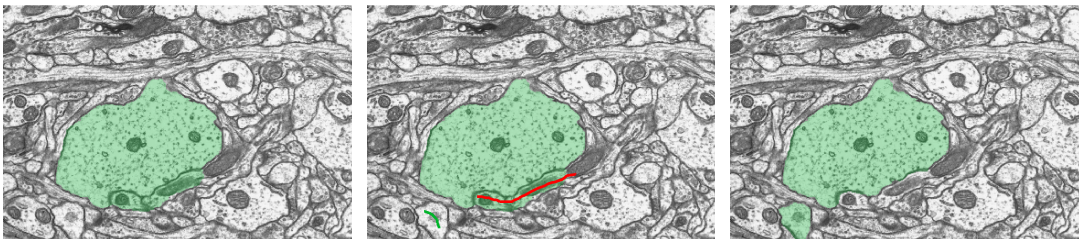


(a) presegmentation (b) edit input (c) edit result

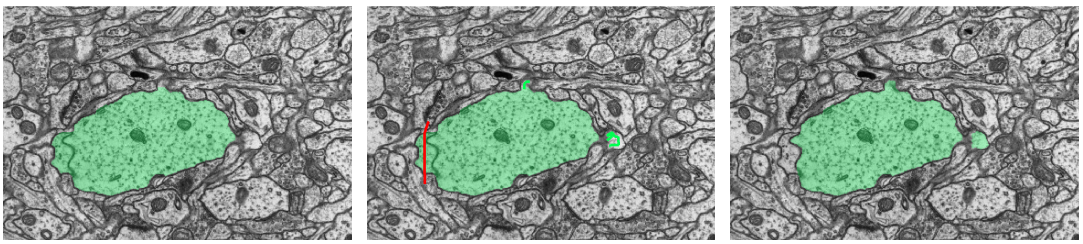
(d) $|\nabla \mathbf{x}|$ (e) $V_p(y_p = 1|y'_p, \mathcal{M}_1)$ (f) $V_p(y_p = 2|y'_p, \mathcal{M}_2)$

Figure 6.1. Example of segmentation editing for the two label case. After user input, the gradient magnitude is used as the speed metric for computing the geodesic distances $d_j(p, \mathcal{M}_i)$. The costs for labeling $y_p = 1$ and $y_p = 2$ are shown in figures 6.1(e) and 6.1(f), respectively. The logarithm of the cost is used for better visualization of details. Higher values are redder while lower values are bluer.

also that this cost is zero near the mark $\mathcal{M}_i$ and progressively gets larger (redder) for pixels farther away. The effect of the geodesic distance is especially noticeable in figure 6.1(f) at the transition from light blue to yellow. The intervening edge causes the immediate pixels on the yellow side of the edge to be much farther from the red mark than the immediate pixels on the blue side. More results of editing for the two-label case are shown in figure 6.2.



(a) presegmentation (b) edit input (c) edit result



(d) presegmentation (e) edit input (f) edit result

Figure 6.2. Examples of segmentation editing for the two-label case. After user input, the editing runtime is approximately ten times faster than recomputing the whole segmentation from scratch.

6.1.3 Multi-Label Case

For the multi-label case, *i.e.* $\mathcal{L} = \{1, 2, \ldots, K\}$, we can use the pairwise potential in equation (6.2) with

$$f(y_p, y_q) = \begin{cases} 1 & \text{if } |y_p - y_q| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.10}$$

However since this potential is not submodular, we must rely on algorithms such as the α-expansion and $\alpha\beta$-swap [10] to find approximate solutions. For cases where we know that the image regions have a nested topology (see chapter 5), we can use

$$f(y_p, y_q) = |y_p - y_q|^k, \tag{6.11}$$

for $k \rightarrow \infty$. For such a case, the CRF energy can be minimized exactly using graph cuts. Chapter 5 describes the graph construction for this problem, and we use the following edge weight assignments:

$$w_{sp_1} = V_p(y_p = 1|y'_p, \mathcal{M}_1) \tag{6.12a}$$

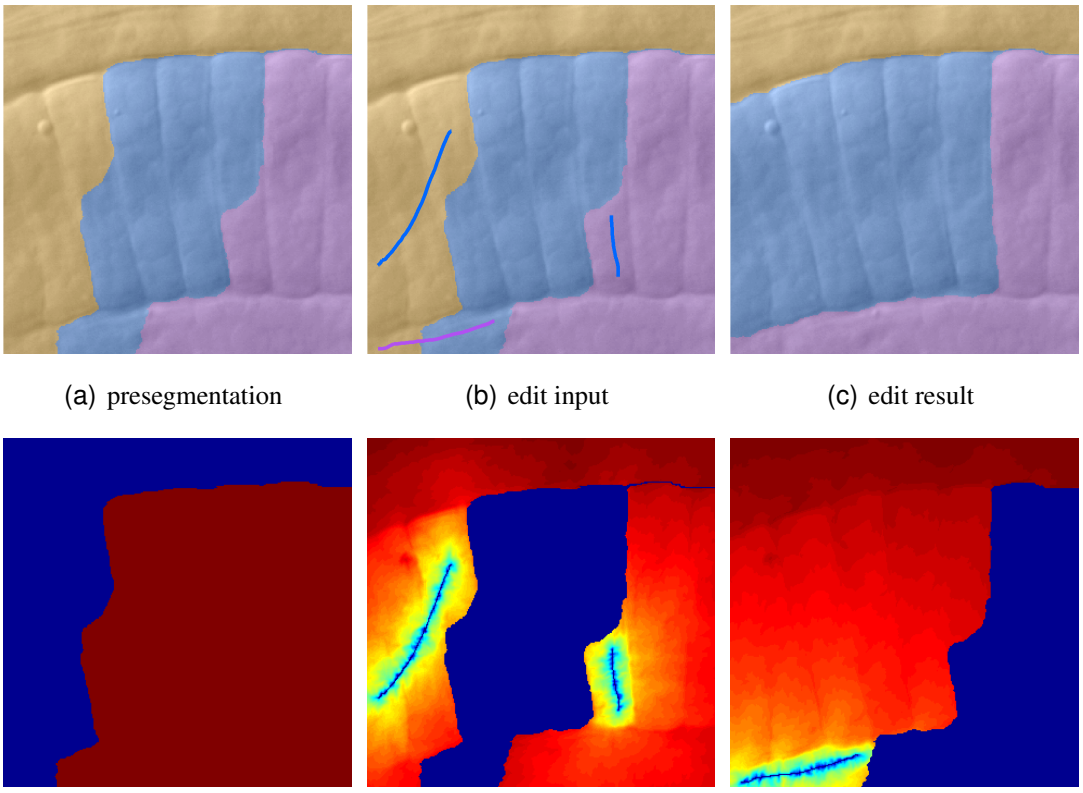
$$w_{p_i p_{i+1}} = V_p(y_p = i+1|y'_p, \mathcal{M}_{i+1}) \tag{6.12b}$$

$$w_{p_{K-1}t} = V_p(y_p = K|y'_p, \mathcal{M}_K) \tag{6.12c}$$

$$w_{p_i q_i} = w_{q_i p_i} = g(p, q). \tag{6.12d}$$

The gradient magnitude is again used in the speed term for the computation of the geodesic distances.

Figure 6.3 shows an example of the editing for the three-label case. Given the pre-segmentation in figure 6.3(a), the user provides input marks indicating the desired label changes as shown in figure 6.3(b). Here the labels are $\{\text{brown} = 1, \text{blue} = 2, \text{purple} = 3\}$. Note that for this case, only edit marks for labels 2 and 3 are given. Figures 6.3(d), 6.3(e), and 6.3(f) show the unary costs $V_p(y_p = i|y'_p, \mathcal{M}_i), \forall i \in \mathcal{L}$. since no input is provided for label 1, all label changes $y_p = 1 \neq y'_p$ is penalized equally while no penalty is incurred for $y_p = 1 = y_p$. This is shown as the solid red and blue regions in figure 6.3(d). The other two costs reflect their respectively input markings, and again for visualization purposes, the logarithm of these costs are shown in figures 6.3(e) and 6.3(f). Notice the effects of the gradient magnitude on the geodesic distance near

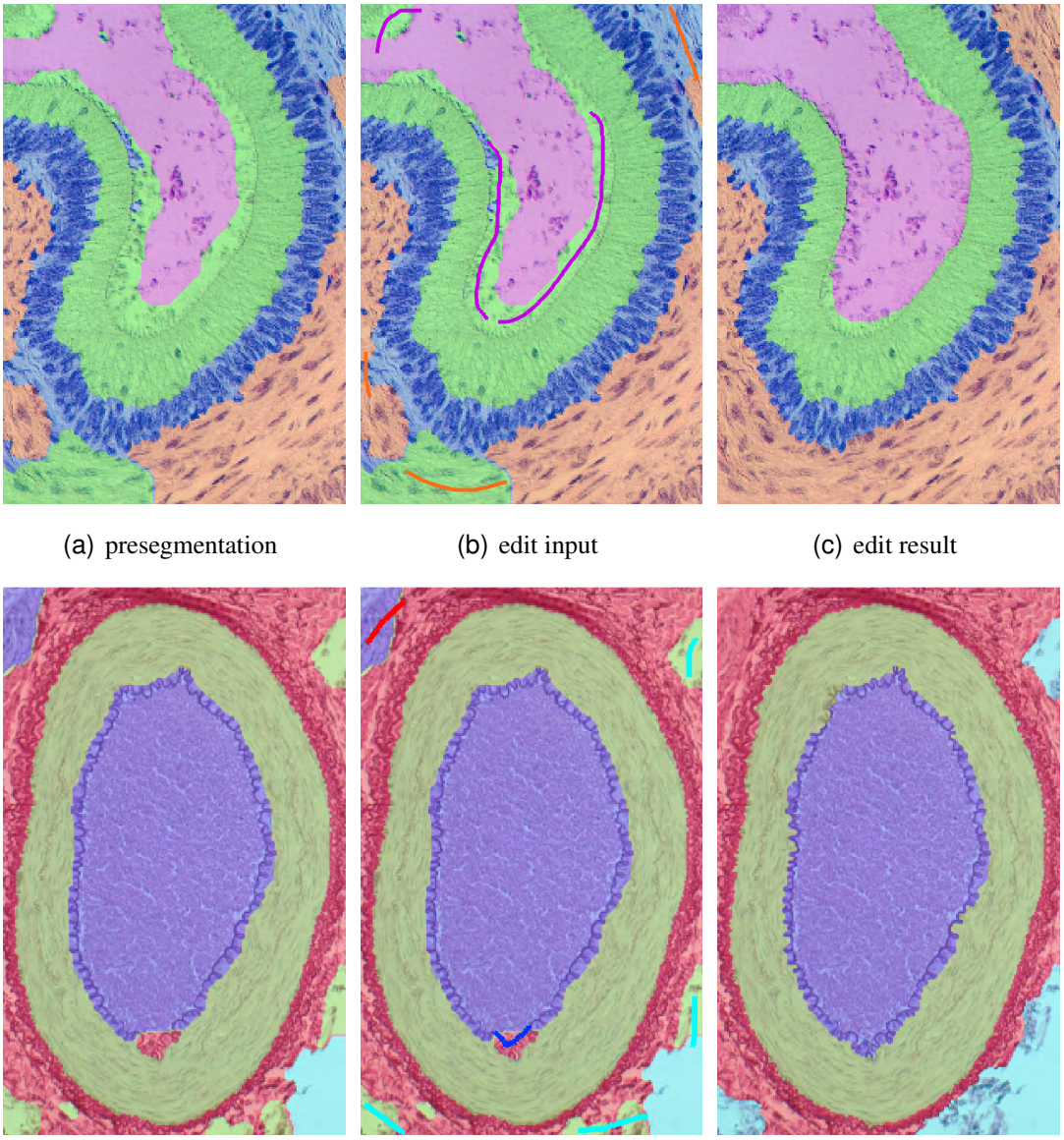


(a) presegmentation (b) edit input (c) edit result

(d) $V_p(y_p = 1|y'_p, \mathcal{M}_1)$ (e) $V_p(y_p = 2|y'_p, \mathcal{M}_2)$ (f) $V_p(y_p = 3|y'_p, \mathcal{M}_3)$

Figure 6.3. Example of segmentation editing for the three-label case. Here the labels are $\{\text{brown} = 1, \text{blue} = 2, \text{purple} = 3\}$. The gradient magnitude is used to compute the geodesic distances.

edges. The contrast between the light blue and yellow regions is very sharp. The image in figure 6.3 was provided by Dr. Michael Veeman from the Department of Molecular, Cellular and Developmental Biology at UC Santa Barbara. Figure 6.4 shows two more examples of presegmentation editing for images with nested region topologies and 4 labels. The top image was provided by Professor Nedim C. Buyukmihci from Veterinary



(a) presegmentation (b) edit input (c) edit result

(d) presegmentation (e) edit input (f) edit result

Figure 6.4. Examples of segmentation editing for the four-label case. After user input, the editing runtime is approximately ten times faster than recomputing the whole segmentation from scratch.

Medicine at UC Davis, and the bottom image was provided by Dr. Darl Ray Swartz from the Department of Animal Science at Purdue University, IN.

Finally, we conclude this section by briefly discussing the runtime. For both the two-label and multi-label case, the run time after user input is approximately three seconds or less. The geodesic distance computation using the Fast Marching algorithm can be done very quickly, usually under half a second. The maximum flow and hence minimum cost cut can be computed efficiently because a large portion of the edges have weights equal to zero, *i.e*. large regions with no label change. These edges are already saturated and so the maxflow algorithm does not need to spend much time pushing flow through them. This allows the the graph cut algorithm to run ten or more times faster compared to recomputing the entire segmentation from scratch, such as using the methods in chapter 3 to segment the EM images or the methods in chapter 5 to segment the layer images. We now turn our attention to the bioimage applications that use segmentation results for information extraction and analysis.

6.2 Analysis of the Outer Nuclear Layer

Recent advances in cytochemical antibody labeling and confocal imaging have allowed biologists to observe protein expressions in relatively small tissue sections with greater detail. In studying the mammalian retina and its response to injury, antibody

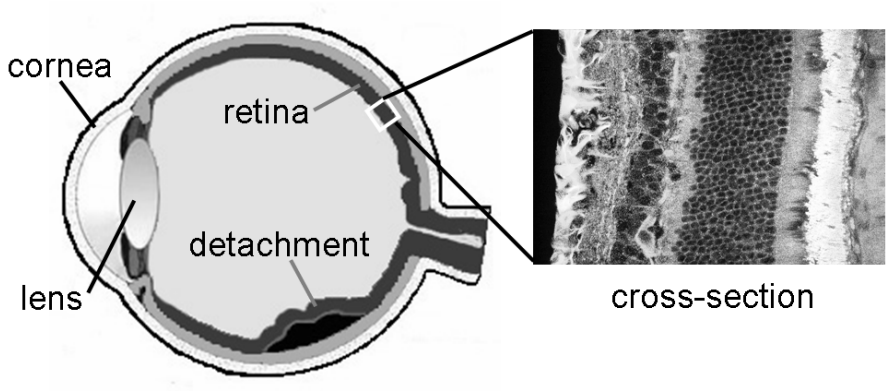


Figure 6.5. Retina detachment.

labeling is often used to target specific tissue layers or cell populations and can reveal intricate structural changes that are closely correlated with functional impairments. In this and the subsequent sections, we focus on analyzing confocal microscope images of the feline retina acquired during retinal detachment experiments. The retina images we use were provided by Dr. Geoffrey Lewis, Dr. Mark Verado, and Prof. Steven Fisher from the Neuroscience Research Institute at UC Santa Barbara.

The retina is the light-sensitive layer of tissue that lines the inside of the eye and is composed of three layers of nerve cell bodies and two layers of synapses (see figure 6.5). The main function of this complex tissue structure is to transform captured light into image-forming signals which are transmitted to the brain. When the retina detaches, it is lifted or pulled from the pigmented epithelium, which normally provides

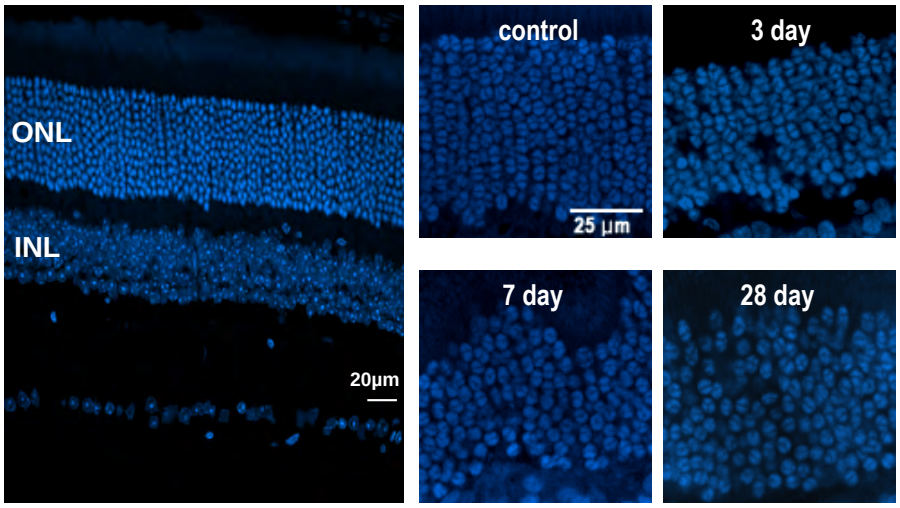


Figure 6.6. Retina images stained with TOPRO. The thickness and photoreceptor density of the outer nuclear layer (ONL) is the focus of our analysis. (b) Note the change in the photoreceptor density after detachment.

the retina cells with the necessary nutrients. If not promptly treated, retinal detachment can cause permanent vision loss.

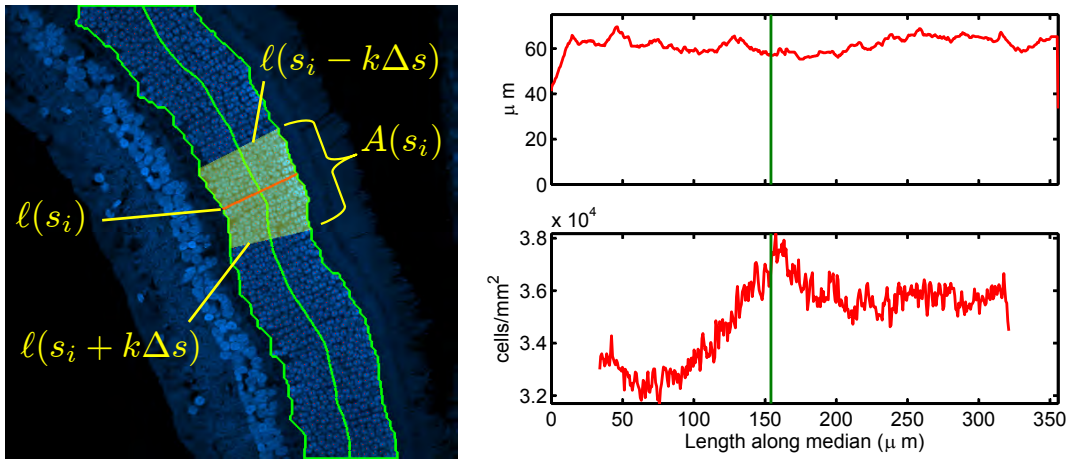
Understanding the mechanisms behind the loss and recovery of vision following retinal detachment has been the focus of many studies [33]. Photoreceptors have received the greatest attention since photoreceptor outer segment degeneration is considered the primary effect of detachment. For example, the outer nuclear layer (ONL) appears to be much more loosely packed with nuclei as a result of cell loss following detachment [33] (see figure 6.6). Degeneration of the photoreceptors has been measured in various ways [73, 80, 33]: by the number of rows of nuclei in the ONL, the area of the ONL,

the thickness of the ONL, and the number of nuclei. The change in these measurements over time is used as an index of photoreceptor degeneration.

In this study, our goal is to measure the change in thickness and photoreceptor density of the ONL during detachment. In order to detect the photoreceptors and highlight the ONL, the retinal cross-sections were stained with the molecular probe TOPRO and imaged using a laser scanning confocal microscope. Figure 6.6 shows several examples. Before computing density measurements, an accurate detection of the photoreceptors is need to ensure the density calculation is reliable. The nuclear detector described in [14], with detection accuracy comparable to that of human experts, was used to locate the photoreceptors. Afterwards simple morphological operations on the nuclei centers were carried out to segment the ONL boundary. However, more sophisticated methods, such as the one in chapter 5 could have been used instead. Nonetheless, this analysis highlights the need for accurate segmentation.

6.2.1 Local Thickness and Density Measurements

Thickness measurements require finding correspondences between points on the inner and outer boundaries of the ONL, but this is a challenging task. To simplify the computation, we extract a median curve that runs along the length of the ONL and use it for the thickness and density measurements. The median extraction process is described in an earlier work [14]. Figure 6.7(a) shows an example of the median curve.



(a) (b)

Figure 6.7. (a) Computation of the ONL thickness and photoreceptor density using the median curve. (b) The thickness and density profiles along the entire length of the ONL.

For every location s_i along the curve, a line $\ell(s_i)$ orthogonal to that median location is constructed and extended outward to the inner and outer boundaries ONL (red line in figure 6.7(a)). The thickness at location s_i is defined as the length of $\ell(s_i)$. Thickness measurements along the entire length of the median provide a continuous thickness profile as shown in the upper plot in figure 6.7(b). The vertical green line in the plot indicates the location of the point s_i in figure 6.7(a).

To compute the local density at s_i, a region $A(s_i)$ centered on a median point s_i and bounded by $\ell(s_i - k\Delta s)$ and $\ell(s_i + k\Delta)$ is constructed. Here k is an integer parameter controlling the size of $A(s_i)$ and Δs is the finite length element between adjacent median points. The highlighted portion of the ONL in figure 6.7(a) shows an

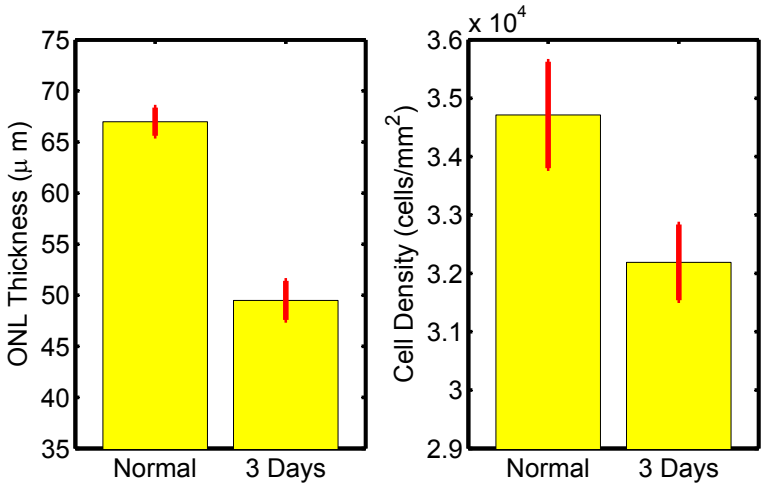


Figure 6.8. ONL thickness and photoreceptor density for images of normal and 3-day detached retinas. Both decreases in these measurements are significant ($p = 0.01$) and suggest that the number of photoreceptors declined in response to retinal detachment.

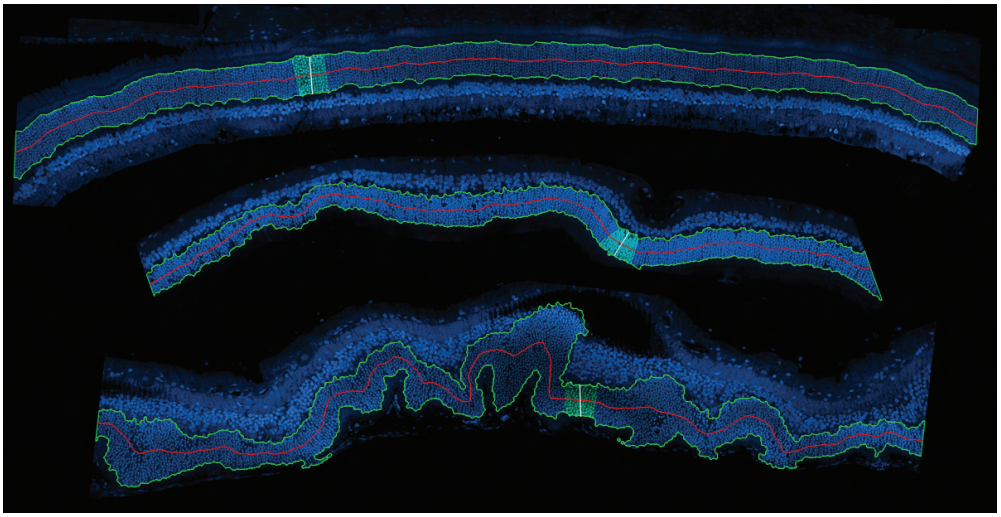
example of a this region. Then density for location s_i is simply the number of nuclei located inside $A(s_i)$ divided by the area of $A(s_i)$. By sliding $A(s_i)$ along the length (except near the two ends) of the median, the local density profile is generated. The bottom plot in figure 6.7(b) shows the density profile along the length of the ONL for this example.

6.2.2 Analysis Results

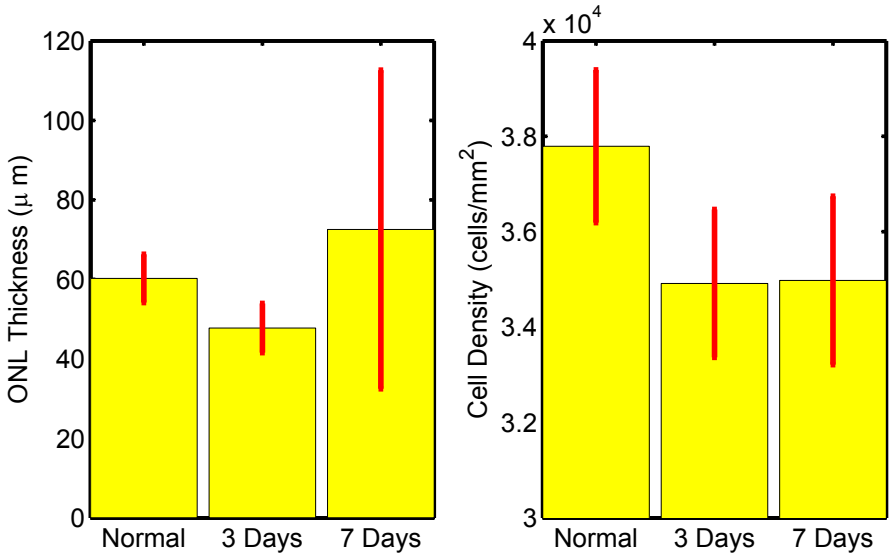
We use the measurement methods described to extract the ONL thickness and photoreceptor density profiles from a set of TOPRO stained images. The image collection consists of 21 control or normal and 20 3-day detached feline retina cross-sections. For each profile, we calculate the mean and standard deviation, and to measure the global

change in ONL, we average these means over all images under the same experimental condition. Figure 6.8 shows the result of the thickness and density calculations for the two experimental groups. Both the ONL thickness and density measurements confirm that the number of photoreceptors in the ONL decreased in response to retinal detachment. The average ONL thickness significantly decreased from 66.9874 μm to 49.4947 μm after detachment ($p = 0.01$). Likewise, the same significant trend is observed in the average density. These measurements corroborate earlier qualitative predictions made in [33].

In a second experiment, large retina cross-sections from normal, 3-day detached, and 7-day detached retinas were examined. Figure 6.9(a) shows the three cross-sections which were mosaiced together from smaller images. The normal retina is shown on top, while the middle and bottom sections corresponds to the 3-day and 7-day detached cases, respectively. Figure 6.9(b) shows the result of thickness and density measurements. The normal to 3-day decrease is similar to that of the previous experiment. However after seven days of detachment, the ONL becomes structurally convoluted and the thickness increases but exhibits large variations. Interestingly, the cell density level remains approximately the same at the 3-day detached stage. The combination of these two measurements may suggest that the majority of photoreceptor deaths occurs before and up to the first three days of detachment, but the population becomes stable



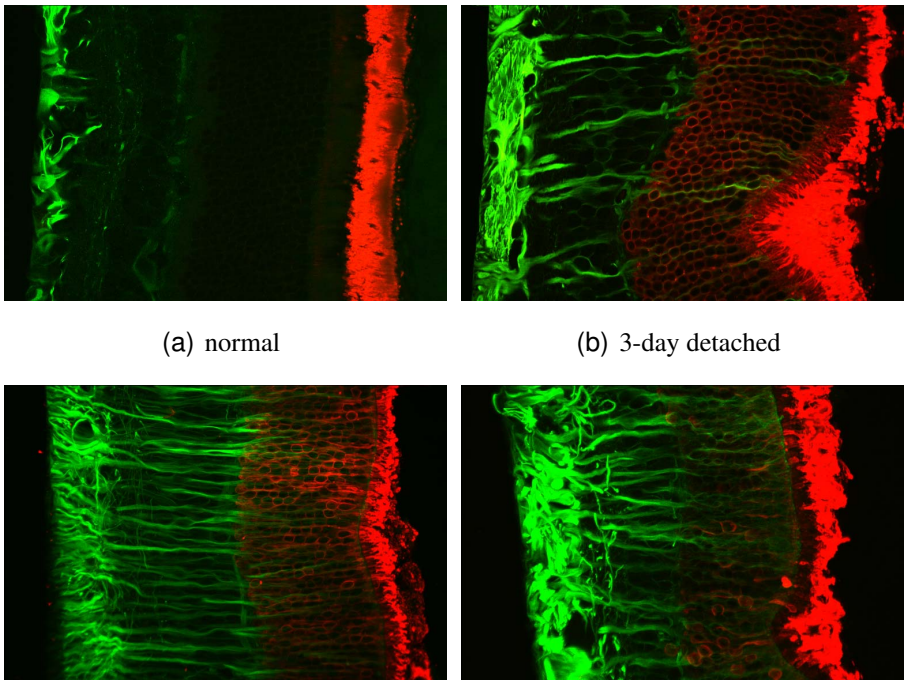
(a)



(b)

Figure 6.9. (a) From top to bottom, mosaiced images of normal, 3-day detached, and 7-day detached retinas. (b) Average ONL thickness and density for the three images.

after three days. However, more experiments are needed to confirm this trend. Next, we describe the analysis of other structural changes that occur during retinal detachment.



(a) normal (b) 3-day detached

(c) 7-day detached (d) 28-day detached

Figure 6.10. Retina cross-sections stained with rod opsin (red) and GFAP (green). The spatial redistribution of these antibody proteins provides insight into structural changes in the retina.

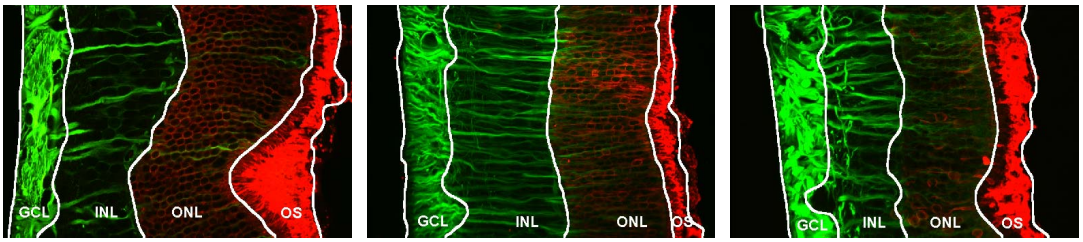
6.3 Spatial Analysis of Antibody Expression Levels

Besides using TOPRO to stain the photoreceptor nuclei, two other important antibody labels are used to target highly responsive proteins during retinal detachment experiments. The first, rod opsin, is found in rod photoreceptor outer segments under normal conditions and is a good indicator of the rod's ability to detect light stimuli. The second, glial fibrillary acidic protein (GFAP), is predominantly localized in Muller cell endfoot regions under normal conditions (see figure 6.10). These Muller cells have

been found to be highly reactive to detachment, undergoing hypertrophy and triggering a cascade of undesirable events leading to decreased neuronal stability and potentially significant vision impairment [33].

To quantify the extent of tissue restructuring during detachment, it is necessary to determine both the changes in magnitude and location of the antibody expression levels across the different detachment conditions. Measures such as the percentage of GFAP penetration into the ONL or the change in rod opsin labeling across the ONL are important biologically and need the spatial correspondence of the ONLs throughout the image set. However, the nature of imaging destroys the specimen being imaged and thus there is no exact physical correspondences that can be made between any two images. The dataset we use in this analysis is composed of images of rod opsin (red channel) and GFAP (green channel) labeled feline retina cross-sections during four stages of retinal detachment. There are 28, 36, 13, and 45 images of normal or undetached, 3-day, 7-day, and 28-day detached retinas, respectively. Because imaging requires destroying the tissue samples and the specimens are costly, only the most interesting stages are fully explored, leading to an unequal number of images in each stage. All images are 512×768 pixels in size, and an example image for each stage is shown in figure 6.10.

In this section, we present a method to compute the rod opsin and GFAP expression levels for an image, which allows for the spatial alignment and comparison of these antibody levels across different images. The computation involves: i) dividing



(a) 3-day detached (b) 7-day detached (c) 28-day detached

Figure 6.11. Segmentation results for several images shown in figure 6.10. The retina has been divided longitudinally into four layers: the ganglion cell layer (GCL), the inner nuclear layer (INL), the ONL, and the rod outer segment (OS).

the retinal layers along their lengths into smaller corresponding sub-layers using the solution of the Laplace equation with Dirichlet boundary conditions, and ii) computing the expression level of each sub-layer and comparing expression levels for retinas at different experimental stages. Before the expression levels can be measured, accurate segmentations of the retinal into distinct layers are required. In this work, we use the segmentation results presented in an earlier work [114] (see figure 6.11), but the layer segmentation algorithm in chapter 5 could have been used as well. Regardless of the segmentation method used, the analysis presented here emphasizes the reliant on accurate segmentations before information extraction can take place.

6.3.1 Expression Level Correspondence

For each image, the retina cross-section is divided longitudinally into four layers: the ganglion cell layer (GCL), the inner nuclear layer (INL), the ONL, and the rod

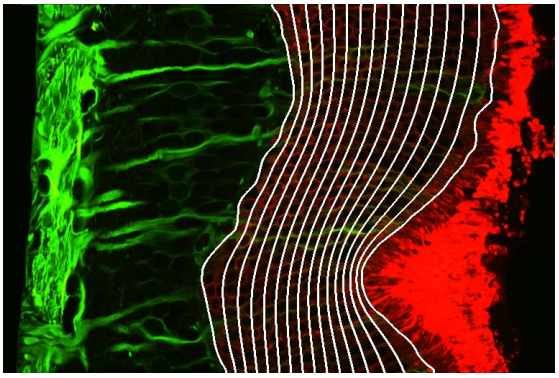


Figure 6.12. ONL sliced into 15 sublayers.

outer segment (OS). To compute the spatial distribution of antibody levels across the retina, we further divide each retinal layer into thinner sublayers (see figure 6.12). This is achieved by computing the solution of the Laplace equation with Dirichlet boundary conditions. Then the sublayer expression levels for each retina are projected onto a spatial layer template that is independent of the particular retina being analyzed.

Let a retina layer be denoted R and its inner and outer boundaries ∂R_0 and ∂R_1, respectively. Then the solution u to the Laplace equation

$$\Delta u = 0, \tag{6.13}$$

subject to the boundary conditions

$$u(\partial R_0) = 0 \text{ and } u(\partial R_1) = 1, \tag{6.14}$$

provides a set of equal potential contours between the two boundaries. The layer R is sliced longitudinally into sublayers by thresholding u at values between 0 and 1. If L

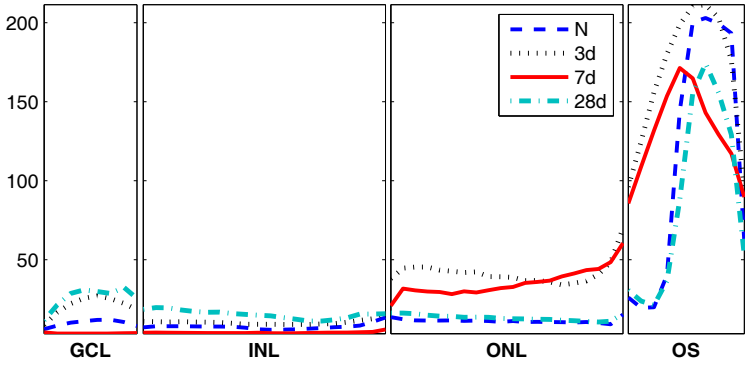
sublayers are needed, then each sublayer ℓ_n is given by

$$\ell_n = \{u : \ (n-1)\Delta u \leq u < n\Delta u\}, \tag{6.15}$$

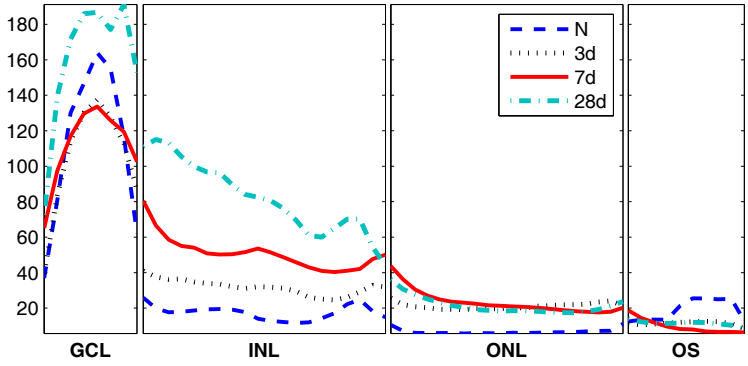
where $\Delta u = 1/L$ and $n = 1, \ 2, \ldots L$. Figure 6.12 shows the result of slicing the ONL layer into 15 sub-layers. After slicing, the protein expression statistics, such as mean and standard deviation, for each sublayer can be easily computed.

6.3.2 Preliminary Biological Analysis

The GCL, INL, ONL, and OS layers in all images are sliced into 8, 20, 20 and 10 sublayers, respectively. For each detachment stage, the average rod opsin and GFAP expression levels in each sublayers are computed and the results are plotted in figure 6.13. Note that the x-axis identifies the relative locations in the retina where the expression values were computed. There are two interesting trends in the figure. First, the rod opsin level in the ONL increased shortly after detachment, but after 28 days of detachment, this level decreased toward the normal level. This may suggest that the rod cells are recovering some of their normal functions without any intervention such as reattachment. The second trend shows the GFAP level in the INL and ONL increased throughout detachment and remain high even after 28 days. This may suggest that once the Muller cells undergo cytoskeletal changes, the effects are difficult to reverse. Stu-



(a) rod opsin expression levels



(b) GFAP expression levels

Figure 6.13. Antibody expression levels for the four stages of retinal detachment.

dent t-tests with $p = 0.05$ confirmed that the differences in expression levels of these two trends are significant.

6.4 Conclusion

For many applications, information extraction and subsequently knowledge formation are highly dependent on the quality of image segmentation results. Yet, the ill-posedness of the segmentation problem makes it extremely unlikely that completely accurate solutions can be found for every image. This is especially true for biomedical data, since imaging conditions can vary greatly even for images from the same experiment. In the first part of this chapter, we presented an algorithm for editing segmentation results. Given that we can obtain nearly correct results, editing is an attractive and viable alternative to fine-tuning existing algorithms for marginal gains in accuracy. Our algorithm is efficient, requires minimal user input, and makes use of both the pre-segmentation and the image data. In the second part of the chapter, we described two bioimage analysis applications that rely on segmentation results for information extraction. In the first study, the ONL segmentation was used to compute the layer thickness and photoreceptor density. In the second study, the layer segmentation results were used to compute and compare antibody expression profiles. These analyses provide valuable quantitative measurements of anatomical changes occurring during retinal detachment experiments.

Chapter 7

Conclusion and Future Outlook

In this thesis, we presented a set of novel image segmentation algorithms that utilize prior information available from domain knowledge to reduce the inherently ill-posedness of the segmentation problem and constrain the results to a more semantically meaningful solution space. These segmentation algorithms provide a sound framework and can serve as good starting points on which to build future extensions. Though the experiments showed that these algorithms can compute quality solutions to some difficult segmentation problems, their performances are by no means completely satisfactory in terms of accuracy, efficiency, and robustness. In this chapter, we offer several preliminary proposals for improving some of the algorithms presented and discuss potential applications that can benefit from using our algorithms.

7.1 Future Directions

Collectively, the computational efficiency of the proposed algorithms can be greatly improved by using a faster programming language, such as C++ , instead of the MATLAB environment for the implementation . To increase segmentation accuracy, a more thorough investigation of the robustness of the visual features used and the sensitivity of the parameter settings is warranted. However, the discussion in this section focuses mainly on the potential research directions of the nested layer segmentation algorithm described in chapter 5.

7.1.1 Hierarchical Layer Segmentation

We have already alluded to using the nested layer segmentation algorithm in chapter 5 to segment images that exhibit a hierarchical nested layer relationship. For example, the image of an Ascidian shown in figure 7.1 has been segmented into five different layers. As can be seen, the notochord cell layer (purple) also exhibits an ordered spatial relationship. Potentially, these cells can be individually segmented by performing another nested layer segmentation on this layer. There are similar cases where several image regions, not necessarily nested, can be grouped into a single layer. After the nested layer segmentation, this mixed layer can be further partitioned using methods such as the α-expansion algorithm [10].

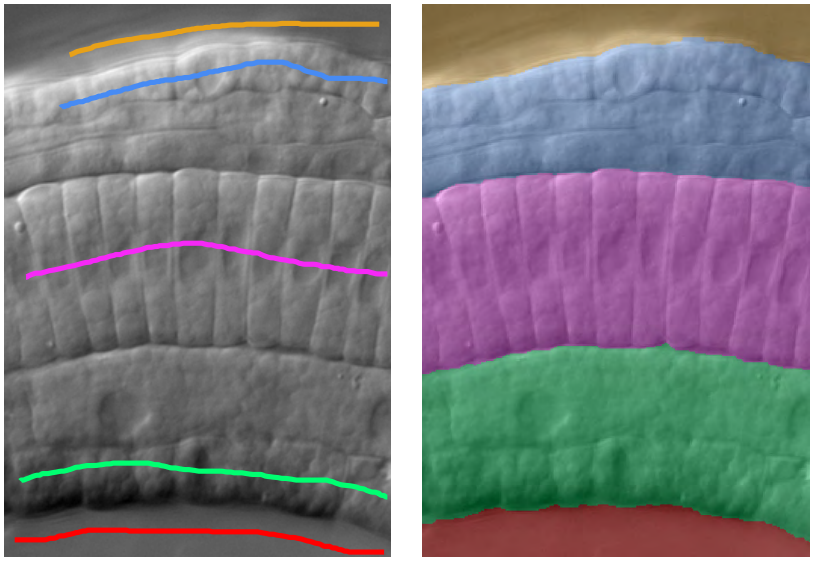


Figure 7.1. First step in the hierarchical layer segmentation of an Ascidian image. A second step of nested layer segmentation can be carried out on the purple region where the notochord cells exhibit spatial layering.

7.1.2 Higher Order $\mathcal{P}^n$ Potts Potential

The nested layer algorithm can also benefit from having a more descriptive clique potential that is able to capture higher order spatial patterns. Several of the segmentation examples in chapter 5 required relatively large exemplar regions for density estimation. Such selections are necessary because the texture features that we used are unable to adequately characterize larger, more inhomogeneous textures. Moreover, the efficiency of the kernel density estimation is also reduced given the large sample size. It has been shown that larger spatial texture patterns can be better represented using a texture dictionary of small image tiles [58, 59]. By tiling the image and representing

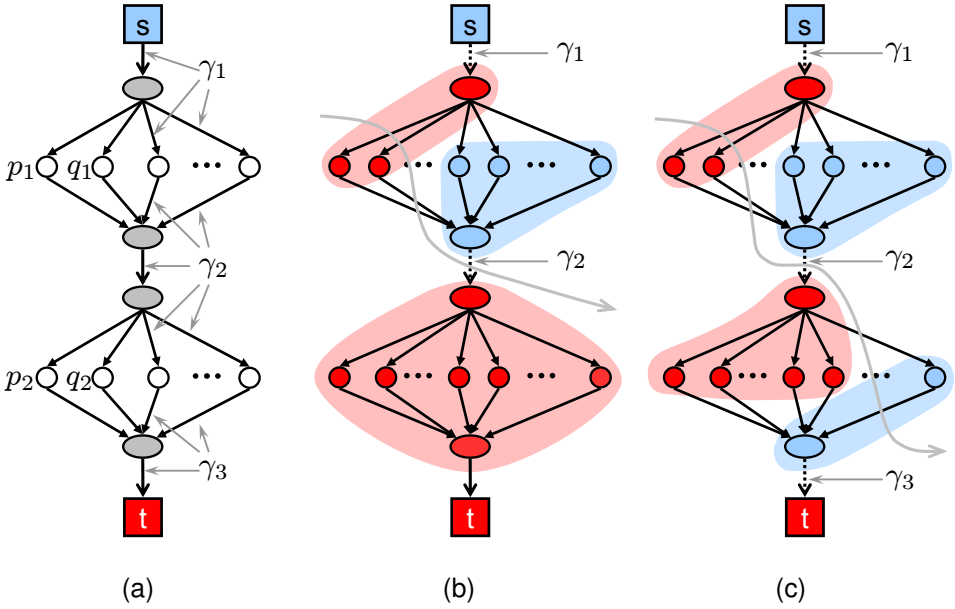


Figure 7.2. The graph representation for the $\mathcal{P}^n$ higher order Potts potential with three labels $\mathcal{L} = \{1, 2, 3\}$. (b) Cut in which the clique nodes have mixed labels 1 and 2, and the cost $\gamma_1 + \gamma_2$ is incurred. (c) Cut in which the mixed labels are 1, 2, and 3. The cut cost is $\gamma_1 + \gamma_2 + \gamma_3$ for this case.

each tile's pixels as a clique, the $\mathcal{P}^n$ Potts potential described in chapter 2 can be used to better discriminate the texture patterns.

We have recently begun to explore the potential for adding the higher order cliques into the proposed nested layer segmentation algorithm. The extension is straightforward and we propose the following Potts potential of clique label $\mathbf{y}_c$:

$$V_c(\mathbf{y}_c) = \begin{cases} \gamma_i & \text{if } y_p = i, \forall p \in c, \\ \sum_{i \in \mathbf{y}_c} \gamma_i & \text{otherwise.} \end{cases} \tag{7.1}$$

This equation simply states that if all the clique pixels are assigned the same label $i \in \mathcal{L}$, then a cost of γ_i is incurred. However, if the clique labels are mixed, then the penalty is the sum of the different label assignment costs in that clique. The graph representation for this higher order potential with three labels $\mathcal{L} = \{1, 2, 3\}$ is shown in figure 7.2 along with the edge weight assignments. Figure 7.2(b) illustrates a cut where the clique pixels have mixed labels of 1 and 2, and the edges with costs γ_1 and γ_2 are in the cut. Figure 7.2(c) shows an example where the clique pixels are assigned a mixture of all three labels. In this case, the penalty for the three cut edges is $\gamma_1 + \gamma_2 + \gamma_3$. Note that the label adjacency constraint prevents the case where the clique pixels have mixed labels that are not consecutive, such as 1 and 3.

7.1.3 Layer Segmentation with Probabilistic Nesting

The nested layer segmentation algorithm strictly enforces the label adjacency constraint, preventing neighboring pixel pairs from having label assignments that differ by more than one. However, there are situations in biology, such as in a diseased state, where the anatomical regions have a probabilistic nesting relationship. That is, a layer A may be adjacent to layer B with probability P_1 and is adjacent to layer C with probability P_2. It is desirable then to enforce the label adjacency constraint in a probabilistic manner. We have not explored this idea further, but anticipate that a strong, possibly

globally optimal, solution can be found if the nesting probabilities are monotonically decreasing with increasing label differences.

7.2 Conclusion

In this thesis, we presented several novel image segmentation algorithms that use higher-level semantic priors to help constrain the feasible solution space. These algorithms formulate the segmentation as a problem of computing the maximum *a posteriori* solutions to Conditional Random Field (CRF) models. Priors such as the object shape, the layer topology, and the adjacent cross-sectional contours, and the presegmentation are incorporated into the CRF energy and subsequently into graph structures that allow for efficient optimization. We also highlighted two examples of bioimage analysis applications that rely on having accurate segmentation results for information extraction. We do not claim that the proposed algorithms can solve the general segmentation problem. However, as the experiments demonstrate, these algorithms produce quality segmentations for specific applications when the available prior information is appropriately utilized.

Bibliography

[1] A. Bartesaghi, G. Sapiro, and S. Subramaniam, "An energy-based three-dimensional segmentation approach for the quantitative interpretation of electron tomograms," *IEEE Trans. Image Process.*, vol. 14, no. 9, pp. 1314–1323, Sep. 2005. 49

[2] L. Bertelli, M. Zuliani, and B. Manjunath, "Pairwise similarities across images for multiple view rigid/non-rigid segmentation and registration," in *Proc. IEEE Int'l Conf. Computer Vision*, 2007. 7, 8

[3] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. R. Statist. Soc. B*, vol. 36, no. 2, pp. 192–236, 1974. 25

[4] ——, "On the statistical analysis of dirty pictures," *J. R. Statist. Soc. B*, vol. 48, no. 3, pp. 259–302, 1986. 19, 21, 24, 34

[5] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *Proc. Euro. Conf. Computer Vision*, 2004. 29

[6] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 155–225, Nov. 2002. 31, 124

[7] Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & regionsegmentation of objects in n-d images," in *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, Vancouver, BC, Canada, 2001, pp. 105–112. 11, 21, 29, 83, 89, 161

[8] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proc. IEEE Int'l Conf. Computer Vision*, Oct. 2003, pp. 26–33. 87, 141

[9] ——, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004. 13, 21, 45, 105, 140

[10] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001. 12, 33, 35, 83, 84, 149, 168, 188

[11] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *Int'l J. Computer Vision*, vol. 70, no. 2, pp. 109–131, Nov. 2006. 13, 69, 71, 83

[12] T. Brox and J. Weickert, "A tv flow based local scale measure for texture discrimination," in *Proc. Euro. Conf. Computer Vision*, 2004. 143

[13] ——, "A TV flow based local scale estimate and its application to texture discrimination," *J. Vis. Commun. Image R.*, vol. 17, no. 5, pp. 1053–1073, Oct. 2006. 143

[14] J. Byun, N. Vu, B. Sumengen, and B. Manjunath, "Quantitative analysis of immunofluorescent retinal images," in *Proc. IEEE Int'l Symp. Biomedical Imaging: Macro to Nano*, Apr. 2006, pp. 1268–1271. 175

[15] I. Carlbom, D. Terzopoulos, and K. M. Harris, "Computer-assisted registration, segmentation, and 3d reconstructionfrom images of neuronal tissue sections," *IEEE Trans. Med. Imag.*, vol. 13, no. 2, pp. 351–362, Jun. 1994. 11, 53, 69

[16] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int'l J. Computer Vision*, vol. 22, no. 1, pp. 61–79, Feb. 1997. 4, 12, 20, 83

[17] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, Jun. 2005, pp. 1164–1170. 83, 84, 90

[18] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001. 4, 12, 20, 83, 102

[19] H. Chang, Q. Yang, M. Auer, and B. Parvin, "Modeling of front evolution with graph cut optimization," in *Proc. IEEE Int'l Conf. Image Process.*, vol. 1, Sep./Oct. 2007. 54

[20] G. Chung and L. A. Vese, "Energy minimization based segmentation and denoising using a multilayer level set approach," in *Proc. Int'l Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 3757, St. Augustine, FL, United States, Nov. 2005, pp. 439–455. 9, 121

[21] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Trans. Med. Imag.*, vol. 17, no. 3, pp. 463–468, Jun. 1998. 155

[22] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002. 76

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001. 5

[24] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, Jan. 1995. 6

[25] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *Int'l J. Computer Vision*, vol. 69, no. 3, pp. 335–351, Sep. 2006. 83, 84, 90, 104

[26] S. Dambreville, Y. Rathi, and A. Tannenbaum, "Shape-based approach to robust image segmentation using kernel PCA," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2006, pp. 977–984. 104

[27] A. Delong and Y. Boykov, "A scalable graph-cut algorithm for n-d grids," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8. 157

[28] A. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 11, pp. 1499–1504, Nov. 2003. 141

[29] A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. d. A. Lotufo, "User-steered image segmentation paradigms: Live wire and live lane," *Graphical Models and Image Process.*, vol. 60, no. 4, pp. 233–260, Jul. 1998. 161

[30] P. F. Felzenszwalb, "Representation and detection of deformable shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 208–220, Feb. 2005. 6, 85

[31] J. J. Fernandez, C. O. S. Sorzano, R. Marabini, and J. M. Carazo, "Image processing and 3-d reconstruction in electron microscopy," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 84–94, May 2006. 8, 48, 49

[32] J. C. Fiala, "Three-dimensional structure of synapses in the brain and on the web," in *Proc. Int'l Joint Conf. Neural Networks*, vol. 1, Honolulu, HI, USA, 2002, pp. 1–4. 49

[33] S. K. Fisher, G. P. Lewis, K. A. Linberg, and M. R. Verardo, "Cellular remodeling in mammalian retina: results from studies of experimental retinal detachment," *Progress in Retinal and Eye Research*, vol. 24, pp. 395–431, 2005. 174, 178, 181

[34] Y.-L. Fok, J. C. K. Chan, and R. T. Chin, "Automated analysis of nerve-cell images using active contour models," *IEEE Trans. Med. Imag.*, vol. 15, no. 3, pp. 353–368, Jun. 1996. 53

[35] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton University Press, 1962. 38, 45

[36] A. Foulonneau, P. Charbonnier, and F. Heitz, "Affine-invariant geometric shape priors for region-based active contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1352–1357, Aug. 2006. 92

[37] A. S. Frangakis and R. Hegerl, "Segmentation of biomedical images with eigenvectors," in *Proc. IEEE Int'l Symp. Biomedical Imaging: Nano to Macro*, 2002, pp. 90–93. 54

[38] D. Freedman and P. Drineas, "Energy minimization via graph cuts: settling what is possible," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, Jun. 2005, pp. 939–946. 124, 130

[39] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2005, pp. 755–762. 6, 13, 86

[40] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984. 19, 21, 24, 27, 34

[41] A. V. Goldberg and R. E. Tarjan, "New approach to the maximum-flow problem," *J. Association Computing Machinery*, vol. 35, no. 4, pp. 921–940, Oct. 1988. 45

[42] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall, 2002. 19

[43] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006. 22, 53, 58, 66

[44] L. Grady and G. Funka-Lea, "An energy minimization approach to the data driven editing of presegmented images/volumes," in *Proc. Int'l Conf. Medical Image Computing and Computer-Assisted Intervention*, vol. 4191, Copenhagen, Denmark, Oct. 2006, pp. 888–895. 161, 162, 164, 165

[45] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *J. R. Statist. Soc. B*, vol. 51, pp. 271–279, 1989. 12, 31, 33, 34, 124

[46] U. Grenander, Y. Chow, and D. M. Keenan, *Hands: a pattern theoretic study of biological shapes*. Springer-Verlag, New York, 1991. 6

[47] J. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," 1971, unpublished manuscript. 25

[48] X. Han, C. Xu, and J. L. Prince, "A 2d moving grid geometric deformable model," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2003. 9

[49] ——, "A topology preserving level set method for geometric deformable models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 6, pp. 755–768, Jun. 2003. 9

[50] G. H. P. Ho and P. Shi, "Domain partitioning level set surface for topology constrained multiobject segmentation," in *Proc. IEEE Int'l Symp. Biomedical Imaging: Nano to Macro*, Apr. 2004, pp. 1299–1302. 9

[51] H. Ishikawa, "Exact optimization for markov random fields with convex priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, Oct. 2003. 10, 15, 17, 33, 122, 123, 124, 128, 130, 131, 139, 146, 157

[52] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning, "A shape-space-based approach to tracking myocardial borders andquantifying regional left-ventricular function applied inechocardiography," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 226–238, Mar. 2002. 6

[53] T. Jiang and C. Tomasi, "Level-set curve particles," in *Proc. Euro. Conf. Computer Vision*, 2006. 70

[54] O. Juan, R. Keriven, and G. Postelnicu, "Stochastic motion and the level set method in computer vision: Stochastic active contours," *Int'l J. Computer Vision*, vol. 69, no. 1, pp. 7–25, Aug. 2006. 20

[55] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l J. Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988. 4, 12, 19, 53, 161

[56] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in *Proc. IEEE Int'l Conf. Computer Vision*, Cambridge, MA, USA, Jun. 1995, pp. 810–815. 12, 20

[57] R. Kimmel, *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer New York, 2003, ch. Fast edge integration, pp. 59–77. 4

[58] P. Kohli, M. P. Kumar, and P. H. S. Torr, "P3 & beyond: Solving energies with higher order cliques," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition.*, Jun. 2007, pp. 1–8. 30, 32, 42, 73, 189

[59] P. Kohli, L. Ladicky, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition.*, Jun. 2008, pp. 1–8. 42, 73, 76, 189

[60] P. Kohli, J. Rihan, M. Bray, and P. H. S. Torr, "Simultaneous segmentation and pose estimation of humans using dynamic graph cuts," *Int'l J. Computer Vision*, vol. 79, no. 3, pp. 285–298, Sep. 2008. 27

[61] P. Kohli and P. H. S. Torr, "Dynamic graph cuts for efficient inference in markov random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2079–2088, Dec. 2007. 105, 115, 138

[62] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006. 37

[63] V. Kolmogorov and Y. Boykov, "What metrics can be approximated by geo-cuts, or global optimization of length/area and flux," in *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, Oct. 2005, pp. 564–571. 13, 56, 63, 64, 86

[64] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts-a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1274–1279, Jul. 2007. 138

[65] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004. 12, 30, 31, 32, 33, 36, 38, 41, 57, 88, 124, 138

[66] N. Komodakis, N. Paragios, and G. Tziritas, "MRF optimization via dual decomposition: Message-passing revisited," in *Proc. IEEE Int'l Conf. Computer Vision*, Oct. 2007, pp. 1–8. 37

[67] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "OBJ CUT," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2005, pp. 18–25. 86, 93

[68] S. Kumar and M. Hebert, "Discriminative random fields: a discriminative framework for contextual interaction in classification," in *Proc. IEEE Int'l Conf. Computer Vision.*, Oct. 2003, pp. 1150–1157. 27

[69] ——, "Discriminative random fields," *Int'l J. Computer Vision*, vol. 68, no. 2, pp. 179–201, Jun. 2006. 27, 28, 29, 30, 140

[70] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling," in *Proc. Int'l Conf. Machine Learning*, 2001. 27, 28

[71] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Hilton Head Island, SC, USA, 2000, pp. 316–323. 6, 83, 119

[72] S. Z. Li, *Markov random field modeling in computer vision*. London, UK: Springer-Verlag, 1995. 23, 25, 26

[73] Z. Li, M. O. Tso, H. M. Wang, and D. T. Organisciak, "Amelioration of photic injury in rat retina by ascorbic acid." *Invest. Ophthalmol. Vis. Sci.*, vol. 26, no. 1589, pp. 1589–98, 1985. 174

[74] X. Liu, O. Veksler, and J. Samarabandu, "Graph cut with ordering constraints on labels and its applications," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8. 123, 149

[75] J. Malcolm, Y. Rathi, and A. Tannenbaum, "A graph cut approach to image segmentation in tensor space," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8. 141

[76] ——, "Graph cut segmentation with nonlinear shape priors," in *Proc. IEEE Int'l. Conf. Image Process.*, 2007. 86

[77] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995. 12, 20

[78] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996. 143

[79] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application toevaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, Vancouver, BC, Canada, 2001, pp. 416–423. 117, 118

[80] J. J. Michon, Z. Li, N. Shioura, R. J. Anderson, and M. O. M. Tso, "A comparative study of methods of photoreceptor morphometry," *Invest. Ophthalmol. Vis. Sci.*, vol. 32, no. 2, pp. 280–284, 1991. 174

[81] E. N. Mortensen and W. A. Barrett, "Interactive segmentation with intelligent scissors," *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349–384, Sep. 1998. 161

[82] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989. 19

[83] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. of Comp. Phys.*, vol. 79, pp. 12–49, 1988. 20

[84] N. Paraagios and R. Deriche, "Geodesic active contours for supervised texture segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, Fort Collins, CO, USA, 1999. 4

[85] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, 1998. 37

[86] S.-C. Pei and C.-N. Lin, "Image normalization for pattern recognition," *Image and Vision Computing*, vol. 13, no. 10, pp. 711–723, Dec. 1995. 84, 92

[87] A. Protiere and G. Sapiro, "Interactive image segmentation via adaptive weighted distances," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1046–1057, Apr. 2007. 166

[88] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr, "Exact inference in multi-label crfs with higher order cliques," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8. 124, 125, 127, 129

[89] T. Riklin-Raviv, N. Sochen, and N. Kiryati, "Mutual segmentation with level sets," in *Proc. IEEE Workshop on Perceptual Organization in Computer Vision*, 2006. 7, 8, 70

[90] T. Riklin-Raviv, N. Kiryati, and N. Sochen, "Prior-based segmentation and shape registration in the presence of perspective distortion," *Int'l J. Computer Vision*, vol. 72, no. 3, pp. 309–328, May 2007. 83, 90

[91] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2006, pp. 993–1000. 7, 8

[92] M. Rousson, T. Brox, and R. Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, Jun. 2003, pp. 699–704. 143

[93] M. Rousson and N. Paragios, "Shape priors for level set representations," in *Proc. Euro. Conf. Computer Vision*, 2002. 6, 83, 119

[94] C. Russell, D. Metaxas, C. Restif, and P. Torr, "Using the pn potts model with learning methods to segment live cell images," in *Proc. IEEE Int'l Conf. Computer Vision*, Oct. 2007, pp. 1–8. 42

[95] S. Sarkar and K. L. Boyer, "Perceptual organization in computer vision: a review and a proposalfor a classificatory structure," *IEEE Trans. Systems, Man and Cybernetics*, vol. 23, no. 2, pp. 382–399, Mar./Apr. 1993. 4

[96] D. Schlesinger, "Exact solution of permuted submodular minsum problems," in *Proc. Int'l Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 4679, Ezhou, China, Aug. 2007, pp. 28–38. 33

[97] D. Schlesinger and B. Flach, "Transforming an arbitrary minsum problem into a binary one," Technical Report TUD-FI06-01, Dresden University of Technology, Tech. Rep., 2006. 32, 33, 124, 126, 130, 131

[98] T. Schoenemann and D. Cremers, "Globally optimal image segmentation with an elastic shape prior," in *Proc. IEEE Int'l. Conf. Computer Vision*, 2007. 6, 85, 157

[99] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proc. Int'l Conf. Pattern Recognition*, vol. 3, Aug. 2004, pp. 32–36. 5, 108

[100] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Natl. Acad. Sci. USA*, vol. 93, no. 4, 1996. 165

[101] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000. 4, 13, 21, 54

[102] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. Euro. Conf. Computer Vision*, vol. 3951, Graz, Austria, May 2006, pp. 1–15. 27

[103] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S. C. Zhu, "On advances in statistical modeling of natural images," *J. Math. Imaging Vis.*, vol. 18, no. 1, pp. 17–33, Jan. 2003. 3, 27

[104] Synapse Web, Kristen M. Harris, PI, *http://synapse-web.org*. 49, 81

[105] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008. 33, 34, 35, 36

[106] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proc. ACM SIGGRAPH Computer Graphics*, 1987. 12, 19

[107] S. R. Thiruvenkadam, T. F. Chan, and B.-W. Hong, "Segmentation under occlusions using selective shape prior," in *Proc. Int'l Conf. Scale Space and Variational Methods in Computer Vision*, 2007. 102, 103

[108] A. Torralba, R. Fergus, and W. Freeman, "80 million tiny images: a large dataset for non-parametric object and scene recognition," *IEEE Transactions on : Accepted for future publication Pattern Analysis and Machine Intelligence*. 3

[109] A. Torralba and A. Oliva, "Statistics of natural image categories," *Network: Computation in Neural Systems*, vol. 14, no. 3, pp. 391–412, August 2003. 3

[110] G. M. Treece, R. W. Prager, A. H. Gee, and L. Berman, "Surface interpolation from sparse cross sections using region correspondence," *IEEE Trans. Med. Imag.*, vol. 19, no. 11, pp. 1106–1114, Nov. 2000. 52

[111] A. Tsai, J. Yezzi, A., W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical

imagery using level sets," *IEEE Trans. Med. Imag.*, vol. 22, no. 2, pp. 137–154, Feb. 2003. 6, 83, 119

[112] A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1565–1578, Dec. 2002. 4, 63

[113] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *Int'l J. Computer Vision*, vol. 50, no. 3, pp. 271–293, Dec. 2002. 20, 94, 121

[114] N. Vu, P. Ghosh, and B. S. Manjunath, "Retina layer segmentation and spatial alignment of antibody expression levels," in *Proc. IEEE Int'l Conf. Image Process.*, vol. 2, Sep./Oct. 2007. 182

[115] N. Vu and B. S. Manjunath, "Graph cut segmentation of neuronal structures from transmission electron micrographs," in *Proc. IEEE Int'l Conf. Image Process.*, Oct 2008. 48, 162

[116] ——, "Shape prior segmentation of multiple objects with graph cuts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, USA, Jun. 2008, pp. 1–8. 84

[117] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on trees: message-passing and linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005. 37

[118] R. Whitaker, D. Breen, K. Museth, and N. Soni, "A framework for level set segmentation of volume datasets," in *Proc. of ACM Int'l Workshop on Volume Graphics*, 2001, pp. 159–168. 69

[119] L. R. Williams and D. W. Jacobs, "Stochastic completion fields: A neural model of illusory contour shape and salience," *Neural Computation*, vol. 9, no. 4, May 1997. 4

[120] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998. 4

[121] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast gauss transform and efficient kernel density estimation," in *Proc. IEEE Int'l Conf. Computer Vision*, Oct. 2003, pp. 664–671. 141

[122] J. Yang, L. H. Staib, and J. S. Duncan, "Neighbor-constrained segmentation with level set based 3-d deformable models," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 940–948, Aug. 2004. 9, 123

[123] A. Yezzi, L. Zollei, and T. Kapur, "A variational framework for joint segmentation and registration," in *Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, Kauai, HI, USA, 2001, pp. 44–51. 7, 8, 82

[124] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. of Comp. Phys.*, vol. 127, pp. 179–195, 1996. 20, 121

[125] S. C. Zhu and A. Yuille, "Region competition: unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996. 19

[126] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, Oct. 2003. 52