

GRAPH CUT SEGMENTATION OF NEURONAL STRUCTURES FROM TRANSMISSION ELECTRON MICROGRAPHS

Nhat Vu and B. S. Manjunath

Center for Bio-image Informatics
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
<http://www.bioimage.ucsb.edu/>

ABSTRACT

In many neurophysiological studies, understanding the neuronal circuitry of the brain requires detailed 3D models of the nerve cells and their synapses. Typically, researchers build the 3D models by manually tracing the 2D cross-sectional profiles of the 3D structures from serial electron micrograph (EM) stacks and then construct the models from these 2D contours. While current computer-aided techniques can reduce the tracing time, they often require extensive user interaction. We propose a segmentation framework to extract the 2D profiles that is both fast and requires a minimal amount of user interaction. The framework uses graph cuts to minimize an energy defined over the image intensity and the flux of the intensity gradient field. Furthermore, to correct segmentation errors, our framework allows for efficient and intuitive editing of the initial results.

Index Terms— graph cuts, electron micrograph, flux

1. INTRODUCTION

Understanding the neural connectivity in the brain requires detailed 3D information of the neuroanatomy. Currently, transmission electron microscopy (TEM) is the preferred modality to obtain high magnification views of neuronal membranes, synapses, and subcellular organelles. However to obtain 3D data, a tissue block must be shaved into ultrathin (40-60 nm) serial sections and imaged individually. 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 [1].

To facilitate more efficient information extraction, we propose a framework to segment neuronal structures from serial EM images that is fast and requires a minimal amount of user interaction. 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 within the object without having to label any background regions. We develop and test our algorithm using several serial EM image stacks from the Synapse Web [2]. Two example images are shown in Fig. 1.

There are several notable previous works on segmenting objects in TEM images. Carlbom *et al.* [3] developed a framework using parametric snakes to segment neurons, which requires the user to provide good initializations and allows the user to interact with the snakes to correct for errors. Fok *et al.* [4] also used 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 are not applicable to our dataset. Frangakis and Hegerl [5] used a spectral approach for segmentation, where the objects are segmented in descending order of their saliency. However, it is not apparent that their technique can allow for user interaction or editing. Recently, Chang *et al.* [6] used 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 that is not the case for our dataset. Finally due to space limitations, we refer the interested reader to [7, 8] and the references therein for a more thorough summary of seeded segmentation algorithms and their applications to various types of biomedical images.

2. GRAPH CUTS PRELIMINARIES

Before discussing our segmentation framework, we provide a brief introduction to graph cuts. Image segmentation can be posed as a pixel labeling problem such that, for a set of pixels \mathcal{P} and labels $\mathcal{L} = \{0, 1\}$, we seek a labeling $\mathbf{y} : \mathcal{P} \rightarrow \mathcal{L}$ that

This work is supported by NSF IGERT # DGE-0221713 and NSF ITR grant #ITR-0331697.

minimizes the energy

$$E(\mathbf{y}) = \sum_{p \in \mathcal{P}} \psi_p(y_p) + \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} \psi_{pq}(y_p, y_q). \quad (1)$$

Here $\psi_p(y_p)$ is the penalty for labeling pixel $p \in \mathcal{P}$ with label $y_p \in \mathcal{L}$, and $\psi_{pq}(y_p, y_q)$ is the penalty for the pairwise labeling of pixels $p \in \mathcal{P}$, $q \in \mathcal{N}_p$ with labels $y_p, y_q \in \mathcal{L}$. \mathcal{N}_p denotes the set of pixels that are neighbors of p . Graph cuts can be used to exactly and efficiently minimize Eq. (1) when ψ_{pq} is a submodular function satisfying $\psi_{pq}(0, 0) + \psi_{pq}(1, 1) \leq \psi_{pq}(0, 1) + \psi_{pq}(1, 0)$ [9].

Before discussing graph cuts, we define the graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed weighted graph composed of a set of nodes \mathcal{V} and a set of directed edges \mathcal{E} with nonnegative weights. The set \mathcal{V} consists of two types of nodes: neighborhood nodes made up of the image pixels \mathcal{P} and two terminal nodes s and t . A directed edge from node p to node q is denoted (p, q) and has weight w_{pq} . Note that $(p, q) \neq (q, p)$. We assume that for $p, q \in \mathcal{P}$, if $(p, q) \in \mathcal{E}$ then $(q, p) \in \mathcal{E}$ and $w_{pq} = w_{qp}$. In addition, all pixels $p \in \mathcal{P}$ are connected to s by terminal edge (t-edge) (s, p) and to t by t-edge (p, t) .

A subset of edges $\mathcal{C} \subset \mathcal{E}$ is called an st-cut if the terminal nodes are completely separated in the induced graph $\mathcal{G} = (\mathcal{V}, \mathcal{E} - \mathcal{C})$. That is there are no directed paths from terminal s to 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 $t \in \mathcal{T}$. For simplicity, we will refer to the st-cut simply as a cut. The cost of the cut $|\mathcal{C}|$ is the sum of all the edge weights in \mathcal{C} , and the minimum cost cut (mincut) can be found by solving an equivalent maximum flow problem [10].

In our notation, a pixel p is assigned label $y_p = 1$ (object) if $p \in \mathcal{S}$ and $y_p = 0$ (background) if $p \in \mathcal{T}$. As a result, each cut produces a labeling $\mathbf{y} = \{y_p \mid p \in \mathcal{P}, y_p \in \mathcal{L}\}$ and hence a corresponding energy $E(\mathbf{y})$. The goal is to assign weights to the graph's edges such that the mincut cost $|\mathcal{C}|$ is equal to the minimum energy $E(\mathbf{y})$. As such, the unary penalty is incorporated into the t-edge weights (t-weights) as: $w_{sp} = \psi_p(0)$ and $w_{pt} = \psi_p(1)$ [9]. The remaining neighborhood edge weights (n-weights) will be discussed subsequently.

3. INTERACTIVE SEGMENTATION

For a given EM image, the segmentation begins with the user marking a set of pixels as belonging to the object (see middle column of Fig. 1). By default, the pixels on the image borders are labeled as 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 t-weight assignments:

$$w_{sp} = K, \forall p \in \mathcal{O}, \quad (2a)$$

$$w_{pt} = K, \forall p \in \mathcal{B}, \quad (2b)$$

where K is set to some large constant (10^6) to ensure that $\mathcal{O} \subset \mathcal{S}$ and $\mathcal{B} \subset \mathcal{T}$ after the cut.

3.1. Pairwise penalty using image intensity

In the EM image, the object of interest usually does not exhibit a distinct regional characteristic, such as intensity or texture, that discriminates it from the surrounding background objects. The most distinguishing feature is the dark, albeit thin membrane surrounding the object. To take advantage of this feature, we use the region coherence penalty term [11]

$$\psi_{pq}(y_p, y_q) = |y_p - y_q| \cdot g(x_p, x_q), \quad (3)$$

where

$$g(x_p, x_q) = \frac{1}{\|p - q\|} \exp\left(-\frac{(x_p - x_q)^2}{2\sigma_x^2}\right). \quad (4)$$

Here $x_p \in [0, 255]$ is the intensity value at p , $\|p - q\|$ is the Euclidian distance between p and q , and σ_x is a parameter set to 20 in all experiments. Eq. (3) is minimized by setting the n-weight $w_{pq} = g(x_p, x_q) \forall p \in \mathcal{P}, q \in \mathcal{N}_p$. Accordingly a penalty $g(x_p, x_q)$ is incurred only when the pairwise labels $\{y_p, y_q\}$ are different, which should be the case at the object boundary. However at the boundary, $g(x_p, x_q)$ is small due to large intensity differences between the object and membrane, and thus the cut is more likely to occur here than anywhere else. 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 Eq. (2)) and proceed with graph cuts. But as the middle column of Fig. 1 shows, there are significant portions of the objects that are mislabeled as background when using Eq. (3) alone.

3.2. 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 [12] and graph cut [11] 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.

The flux of a vector field \mathbf{v} through a continuous hypersurface S is given by [11]

$$\varphi(S) = \int_S \langle \mathbf{v}, \hat{\mathbf{n}} \rangle dS, \quad (5)$$

where $\hat{\mathbf{n}}$ is the unit normal to the surface element dS and $\langle \cdot, \cdot \rangle$ is the Euclidian dot product. Here, the field \mathbf{v} is the normalized gradient of the gaussian smoothed image, *i.e.* $\frac{\nabla I_\sigma}{\|\nabla I_\sigma\|}$. We use $\sigma = 3$ for all results in this paper. 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 .

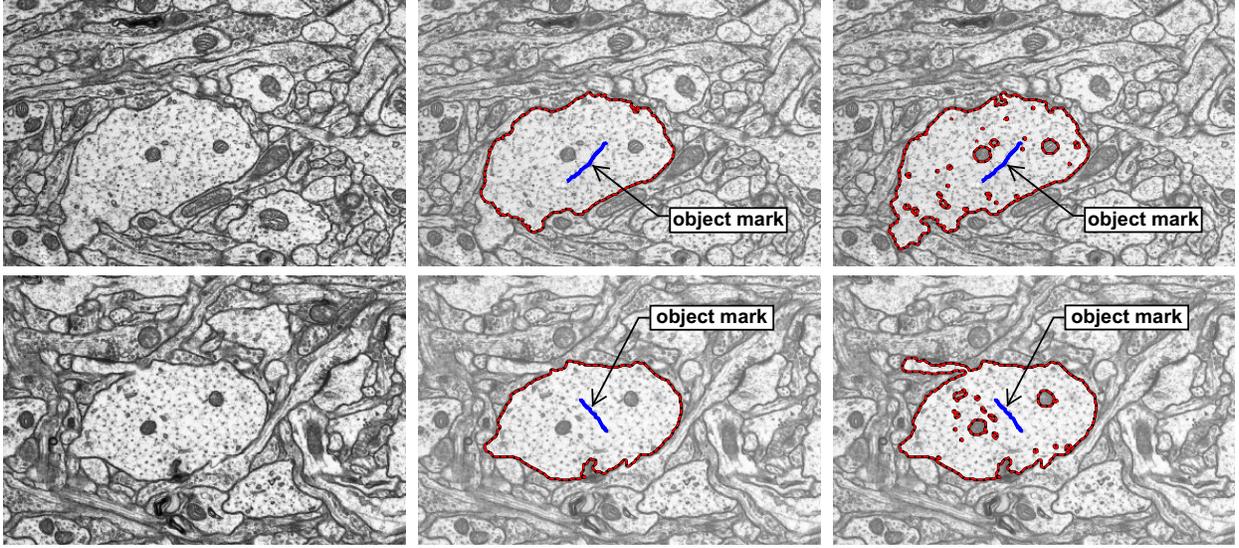


Fig. 1. Original images 255×382 (left), results with intensity alone (middle), and with intensity and flux (right).

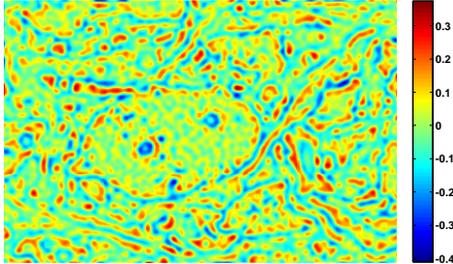


Fig. 2. Flux of gradient field of bottom left image in Fig. 1.

The flux for the bottom left image in Fig. 1 is shown in Fig. 2. The flux is more positive in the lighter intensity regions adjacent to the darker membranes and enhances the contrast between the foreground object and its surrounding membrane. We incorporate the flux into the graph using the t -weight assignments:

$$w_{sp} = \max(0, \varphi(p)), \quad \forall p \in \mathcal{P} \setminus \mathcal{O}, \quad (6a)$$

$$w_{pt} = -\min(0, \varphi(p)), \quad \forall p \in \mathcal{P} \setminus \mathcal{B}. \quad (6b)$$

This formulation favors a cut in which the object's flux is maximized, and encourages the inclusion of convoluted portions of the object.

The right column of Fig. 1 shows the results using both flux and intensity information, with the same user input as before. As shown, the segmented objects contain convoluted regions that were previously missed. There are some minor isolated background blobs inside the objects, but they can be removed with simple morphological operations.

4. INTERACTIVE EDITING

In general, the segmentation may produce unintended results because the algorithm parameters may be difficult to tune for

an entire dataset or because the energy formulation fails to sufficiently capture the user's domain knowledge. Allowing the user to edit the results can be very beneficial and less time consuming than parameter tuning. There are several graph cut methods that allow for editing of the original solution [8, 13]. We base our editing framework on the energy formulation presented in [13], but make some important and necessary modifications in order to adapt it for our application.

Similar to [13], our editing energy makes use of the previous segmentation output, since it is assumed that that result is nearly correct. Second, only pixels that change their labels during editing are penalized. Contrary to the formulation in [13], 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 Euclidean distance used in [13]. 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.

Assume that the previous binary segmentation result is \mathbf{y} with object and background pixel sets \mathcal{S} and \mathcal{T} , respectively. The user edits \mathbf{y} by specifying pixels that should be relabeled (see Fig. 3). Then we seek a new labeling \mathbf{y}' that minimizes the following energy:

$$E(\mathbf{y}') = \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p} \psi_{pq}(y'_p, y'_q) + \sum_{p \in \mathcal{P}} w_{sp} y_p (1 - y'_p) + \sum_{p \in \mathcal{P}} w_{pt} (1 - y_p) y'_p. \quad (7)$$

The pairwise penalty ψ_{pq} is given by Eq. (3), and $w_{pq} = g(x_p, x_q)$ as before. The second term sums the costs of relabeling pixels $p \in \mathcal{S}$ as background, and the third term sums

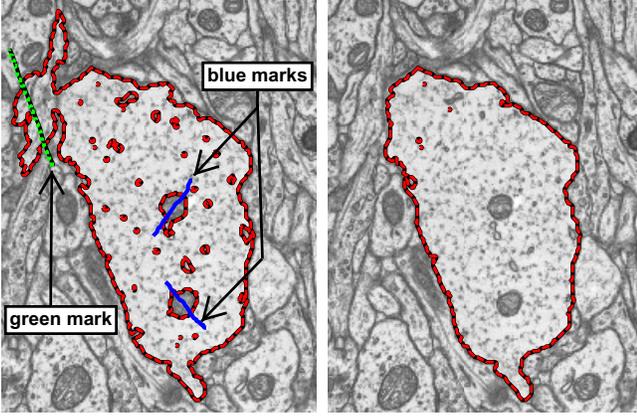


Fig. 3. Blue and green marks in the left image indicate pixels that should be relabeled as background and object, respectively. The edited result is shown on the right.

the costs of relabeling pixels $p \in \mathcal{T}$ as object. Denote the user relabeled pixel sets as \mathcal{O}' for object and \mathcal{B}' for background. The t-weights are then defined as:

$$w_{sp} = \begin{cases} 1 - \exp\left(-\frac{d_g(p, \mathcal{B}')}{\sigma_g}\right), & p \in \mathcal{S} \setminus (\mathcal{O}' \cup \mathcal{B}') \\ K, & p \in \mathcal{O}' \end{cases} \quad (8a)$$

$$w_{pt} = \begin{cases} 1 - \exp\left(-\frac{d_g(p, \mathcal{O}')}{\sigma_g}\right), & p \in \mathcal{T} \setminus (\mathcal{O}' \cup \mathcal{B}') \\ K, & p \in \mathcal{B}' \end{cases} \quad (8b)$$

Here, $d_g(p, \mathcal{O}')$ and $d_g(p, \mathcal{B}')$ are the geodesic distances from pixel p to the closest pixel in the user edited pixel sets \mathcal{O}' and \mathcal{B}' . The geodesic distance is computed using the Fast Marching algorithm [14], with the speed term a function of the image gradient magnitude, *i.e.* $\|\nabla\|$. Fig. 3 shows an example of the editing process. The left image contains edit marks for a previous segmentation, and the right image shows the edited result. In all experiments, σ_g is set to 20. For both the results without and with flux in Fig. 1, after the initial user input the segmentation takes less than a second to run using a Matlab/C++ mex implementation. The editing process is approximately ten times faster.

5. CONCLUSION

We presented a segmentation framework to extract 2D contours of neuronal structures from serial EM images. Using graph cuts, we minimize an energy based on the image intensity and flux of the gradient field. We also proposed an editing procedure, allowing the user to correct segmentation errors. Our method is computationally fast and requires a minimal amount of user input. Future work includes applying our method to a more diverse set of EM images and obtaining quantitative validation of our method against manually segmented ground truth.

6. REFERENCES

- [1] J. C. Fiala, “Three-dimensional structure of synapses in the brain and on the web,” in *IJCNN 2002*.
- [2] Synapse Web, Kristen M. Harris, PI, <http://synapse-web.org>.
- [3] I. Carlbom, D. Terzopoulos, and K. M. Harris, “Computer-assisted registration, segmentation, and 3d reconstruction from images of neuronal tissue sections,” *IEEE Trans. Med. Imag.*, vol. 13, no. 2, pp. 351–362, June 1994.
- [4] 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, June 1996.
- [5] A. S. Frangakis and R. Hegerl, “Segmentation of biomedical images with eigenvectors,” in *ISBI 2002*.
- [6] H. Chang, Q. Yang, M. Auer, and B. Parvin, “Modeling of front evolution with graph cut optimization,” in *ICIP 2007*.
- [7] L. Grady, “Random walks for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.
- [8] 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.
- [9] 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.
- [10] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- [11] V. Kolmogorov and Y. Boykov, “What metrics can be approximated by geo-cuts, or global optimization of length/area and flux,” in *ICCV 2005*.
- [12] A. Vasilevskiy and K. Siddiqi, “Flux maximizing geometric flows,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1565–1578, Dec. 2002.
- [13] L. Grady and G. Funka-Lea, “An energy minimization approach to the data driven editing of presegmented images/volumes,” in *MICCAI 2006*.
- [14] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” *Proc Natl Acad Sci U S A*, vol. 93, no. 4, 1996.