TREE2TREE: NEURON SEGMENTATION FOR GENERATION OF NEURONAL MORPHOLOGY

Saurav Basu[1], *Alla Aksel*[1], *Barry Condron*[2] *and Scott T. Acton*[1]

1. Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia, USA 22904.
2. Department of Biology, University of Virginia, USA 22904.

ABSTRACT

The knowledge of the structure and morphology of neurons is a central part of our understanding of the brain. There have been concerted efforts in recent years to develop libraries of neuronal structures that can be used for multiple purposes including modeling the brain connectivity and understanding how cellular structure regulates function. However, at present, tracing neuronal structures from microscopy images of neurons is very time consuming and somewhat subjective and therefore not practical for the current datasets. Current automatic state of the art algorithms for neuron tracing fail to work in neuron images which have low contrast, amorphous filament boundaries, branches, and clutter. In this paper, we develop *Tree2Tree*, a robust automatic neuron segmentation and morphology generation algorithm. It uses a local medial tree generation strategy for visible parts of the neuron and then uses a global tree linking approach to build a maximum likelihood global tree by combining the local trees. Tests on cluttered confocal microscopy images of Drosophila neurons give results that correspond to ground truth within ±5.3 pixel RMSE margin of error.

Index Terms**—** Segmentation, neuron tracing, morphology, filament tracking

1. INTRODUCTION

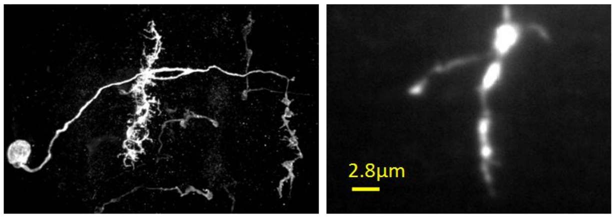
Automated neuron segmentation remains one of the critical open problems in biological image analysis. The neuron hypothesis of Cajal states that the functional unit of the brain is the neuron and that “all computation is anatomical.” It follows then that our knowledge of the structure or morphology of neurons is a central part of our understanding of the brain [1]. A concerted plan is currently underway to develop libraries of neuronal structures that can be used for multiple purposes including modeling [2]. This approach is to identify all the shapes, the *neurome*, and the connectivity, the *connectome* [3], of at least a part of a brain and use this knowledge to understand how cellular structure regulates function.

The only combined neurome/connectome described to date is for *C.elegans* [4] and is arguably that critical tool which established this organism as one of the central pillars of modern developmental and behavioral neurobiology. However, neurons in *C.elegans* tend to be structurally simple and likely do not represent the repertoire of morphological variation seen in animals from *Drosophila* to humans. Therefore, there is a strong argument to develop a similar set of neuronal atlases for other organisms. Given the concentration of genetic, behavioral and physiological data collected for *Drosophila*, it makes most sense to start with this animal.

The *Drosophila* central nervous system (CNS) consists of a brain, associated with higher sensory processing, and contains about 40,000 neurons, and a ventral nerve cord (VNC) containing about 5000 cells [5]. The VNC contains a set of fused subesophageal segments, three large thoracic segments and 7 central reiterated and relatively simple abdominal segments. Each segment of the VNC contains about 130 neurons and many of these have been imaged at high resolution in different animals. The entire data set consists of about 1000 cells yielding about 1GB of information. In order to acquire more useful quantitative information, each of the cells needs to be traced and coordinates entered into a public database [2]. Such information can be used to model electrical conduction in large realistic networks as well as be used to develop methods to rapidly scan for structural features. However, at present, tracing neuronal structures is very time consuming and somewhat subjective and therefore not practical for the current datasets. With the development of newer imaging tools [1, 2] much larger datasets will soon become available. Therefore, the need to develop new tracing strategies will soon become critical.

Current state of the art automatic algorithms [6, 7, 8] for tracing and segmenting neuronal structures are unreliable with neuron images which have low contrast, amorphous filament boundaries, branches, and significant clutter. Al-Kofahi et al. [6] use the median value of a directional filter to estimate the best possible direction of the neuron filament. This algorithm depends on high contrast as well as continuous and parallel boundaries of neuronal structures for success. In addition to it being semi-automatic with user defined seed points, this algorithm does not handle branching reliably.

Wolf et al. [7] manually initialize a quadrilateral directional filter over a seed point and use local relative grey value differences and local maxima of intensity to trace the axons. The starting and stopping criteria are heuristic in nature and the algorithm is dependent on considerable manual adjustments to give acceptable results. In addition, poor contrast can easily throw the algorithm off track. Cai et al. [8] approach the neural tracing problem by segmenting neuronal cross sections in 2D image stacks with active contours [9] and stitching corresponding segmented contours across the slices. The strategy for correspondence between multiple contours across stacks is not well defined, and this method requires optimal alignment of neuronal branches along the z axis for the segmentation to be meaningful.



(a) (b)

Figure 1: (a) GFP stained Drosophila CNS neuron. (b) Close-up of a neuronal branch structure from the same image with the corresponding scale.

In our research, we have used confocal microscopy images of individual sets of GFP-labeled *Drosophila* neurons [10]. Each image is acquired in focal planes from dorsal to ventral. As can be seen from figure 1(b), the neuron images are characterized by low contrast, filament discontinuity, and poorly defined boundaries. Therefore, usual seed growing and edge finding algorithms described above fail to give reliable segmentation. We have developed *Tree2Tree*, an automatic neuron segmentation and morphology generation algorithm that does not require manual seed points, is not dependent on edge consistency and filament continuity, and handles neuron branching naturally. *Tree2Tree* gets its name from the fact that it attempts to fit a graph-theoretic tree to the neuronal tree.

The rest of the paper is arranged as follows. Section 2 introduces *Tree2Tree* in detail. Section 3 shows results of applying *Tree2Tree* to our dataset. Section 4 discusses future extension of this algorithm to the neuronal atlas project.

2. TREE2TREE

Instead of following brightness maxima along filaments, *Tree2Tree* builds a maximum likelihood tree that optimally explains the orientation and connectivity of visible brightness patterns in the neuron images. Brightness patterns due to clutter in specimen preparation are removed by passing through a novel graph based pruning method.

Starting from a 2D intensity image of a neuron, our algorithm outputs the centerline of the neuron as a tree (for a quick reference to graph theoretic concepts, see [11]), with adjacent nodes placed at user defined resolution. The sequential steps, the rationale behind them and the mathematical formulation are presented next.

Pixel Classification (Step 1): The raw neuron segmentation can be treated as a binary classification problem into the foreground neuron pixels (brighter) and the background pixels (darker). A global 2-class classification over the entire image does not give satisfactory results because the relative brightness of the neuron compared to the background varies widely over the image. Therefore, we employ a local binary classifier at each pixel, the end result being a disjoint sets of connected binary 'islands' which suggest possible neuron segments.

Let our intensity image be denoted by $\mathcal{I}: \Lambda \to \mathcal{L}$ where $\Lambda = \{(x, y) | x \in (1, .., N), y \in (1, .., M)\}$, and $\mathcal{L} = \{1,2, .., L\}$. If $p: \mathcal{L} \to [0,1]$ be the normalized histogram of $\mathcal{I}$, and $\tau \in \mathcal{L}$ and $0 \leq N_\tau \leq 1$ be such that

$$\tau = \max_{\tau}(\sum_{i=1}^{\tau} p(i) \leq 1 - N_\tau), \quad (1)$$

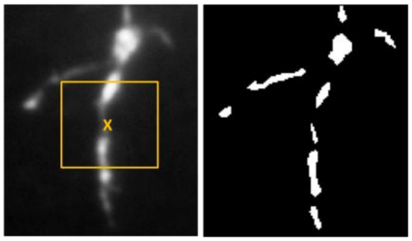
then, we define a class boundary for the binary pixel classification problem at every pixel position as

$$\rho(x, y) = \max\ \{\mathcal{O}(x, y), \tau\} \quad (2).$$

$\mathcal{O}: \Lambda \to \mathcal{L}$ is a local binary classifier performing pixel classification in a $w \times w$ window centered around each pixel (x, y). N_τ denotes the proportional area the neuron occupies in the image frame.

Classifying pixels with the help of local classifier $\rho(x, y)$ has the advantage of treating the relative brightness of neuron pixels by taking the average background intensity of the neighborhood into consideration, while not allowing the class boundary to fall below a minimum level τ (preventing trivial classification over non-neuron areas with uniformly dark pixels). Figure 2 shows the result of local pixel classification of the neuron image into isolated islands or connected components.

Local Medial Trees (**Step 2**): Let $\mathcal{C} \equiv \{C_1, C_2, .., C_n\}$ be n connected components obtained from step 1. The goal of this step is to characterize the connected components according to their filamentous properties, orientation and area.



(a) (b)

Figure 2: (a) Pixel classifier evaluated at every pixel by considering intensity information within a $w \times w$ window centered on that pixel (b) Result of binary classification is a disjoint set of isolated connected component.

We extract the *skeleton* of each connected component by applying the *binary skeletonization algorithm* [12] on a boundary- smoothed version of each connected component. For each connected component, the raw skeleton is then resampled with a user defined resolution r to generate individual medial trees. Each connected component can then be described by the following pair

$$C_j \equiv \{\mathcal{M}_j, \mathcal{A}_j\} \quad (3)$$

where

$$\mathcal{M}_j \equiv \{V_j, E_j\}, V_j \equiv \{\overline{p_j^1}, .., \overline{p_j^{m_j}}\},$$

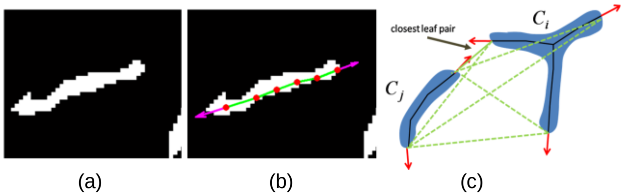
$$E_j \equiv \{(\overline{p_j^k}, \overline{p_j^l}) | \text{ with some } k, l \in \{1,2, \dots, m_j\}, k \neq l\},$$

$$\text{and } \mathcal{A}_j = area(C_j)\ (4)$$

In equation (4), $\mathcal{M}_j$ is the medial tree of C_j, with the set V_j of m_j vertex nodes (with adjacent nodes placed at resolution r) and the set E_j of $(m_j - 1)$ edges (since $\mathcal{M}_j$ is a tree). Thus at the end of step 2 we have n connected components each of which is represented by its medial tree and area.

Global k-NN Graph (**Step 4**): We utilize a graph theoretic approach in estimating the connectivity between the connected components in order to approximate a complete neuron. In order to evaluate the weighted distance between two connected components, we need to characterize the alignment and proximity between components.

For a connected component C_j with medial tree $\mathcal{M}_j$ that has f_j number of leaf nodes (a leaf node is a terminating node in a tree and is connected to only one other node), we can define f_j tangent vectors $\{\overline{t_1^j},..,\overline{t_{f_j}^j}\}$ with $\overline{t_l^j} = \overline{p_j^l} - a(\overline{p_j^l})$ ($a(\overline{p_j^l})$ is the ancestor node for node $\overline{p_j^l}$ in $\mathcal{M}_j$). Refer figures 3(a) and 3(b) to understand the medial tree and the leaf tangents for a connected component.



(a) (b) (c)

Figure 3: (a) A single connected component (b) The medial tree of (a) with nodes shown in red, edges in green and leaf tangents in magenta. (c) 2 connected components C_i and C_j shown in blue, leaf tangents shown in red and medial trees in black. 6 pairs of leaf node connectivity (shown in dotted green) are tested for the closest distance between the components. The closest leaf pair is indicated.

If 2 connected components are in fact part of the same branch of a neuron, then we can expect at least one leaf pair (with each connected component contributing one leaf) to be close to each other and optimally aligned. Thus, for connected components C_i and C_j, we define the distance d_{ij} between them as

$$d_{ij} \equiv \min_{k\in\{1,..,f_i\},l\in\{1,..,f_j\}} \{|\overline{p_i^k} - \overline{p_j^l}|^2 + \pi - angle(\overline{t_k^i}, \overline{t_l^j})\} \quad (5)$$

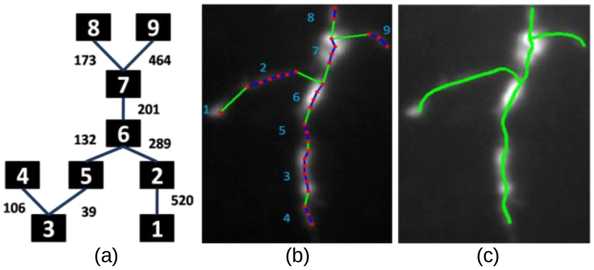
where $angle(\overline{t_k^i}, \overline{t_l^j})$ is the acute angle between tangent vectors $\overline{t_k^i}$ and $\overline{t_l^j}$. For a leaf pair (k, l) that gives the minimum distance in (5), the distance squared $|\overline{p_i^k} - \overline{p_j^l}|^2$ between leaf nodes is small and the tangent vectors face each other (hence angle between them is close to π). See figure 3(c) for understanding the distance calculation in (5).

We can now build a global k-Nearest Neighbor graph $\mathcal{G}$ with the connected components C_j themselves considered as nodes. $\mathcal{G}$ is defined as

$$\mathcal{G} \equiv \{\mathcal{V}, \mathcal{E}\}, \mathcal{V} \equiv \{C_1, .., C_n\},$$
$$\mathcal{E} \equiv \{(C_i, C_j, d_{ij})| \text{ with some } i, j, \in \{1,2, \dots, n\}, i \neq j\} \quad (6).$$

Edge (C_i, C_j, d_{ij}) with edge weight d_{ij} only exists if C_j is one of the k nearest nodes to C_i based on the distance d_{ij} calculated from (5) (Note that $\mathcal{G}$ is directed).

Global Connectivity Tree (**Step 5**): We can now form an optimal connectivity tree by constructing a *minimum spanning tree* $\mathcal{M}$ [11] of the k-NN global graph $\mathcal{G}$ such that the sum of the edge weights d_{ij} over the entire tree is a minimum.



(a) (b) (c)

Figure 4: (a) $\alpha - \beta$ graph pruned global connectivity tree of components from figure 2(b), with connected component numbers in black and edge weights shown besides edges. $\alpha = 0.9$ and $\beta = 0.4$. (b) Pictorial depiction of 4(a) with medial trees for corresponding connected components in blue edges and red nodes. Connections between the medial trees are shown in green. (c) The merged global neuron tree from (b) has been represented with splines.

***$\alpha - \beta$* Graph Pruning** (**Step 6**): The vertex set $\mathcal{V} \equiv \{C_1, .., C_n\}$ in step 5 might have some connected components contributed by image clutter. If we define the likelihood $\mathcal{W}(C_i)$ of each vertex as

$$\mathcal{W}(C_i) = |E_j| + \sqrt{\mathcal{A}_j} \quad (7)$$

then components with longer medial graphs and higher areas have higher likelihood. We can expect noisy connected components to have low node likelihood, at the same time having poorer proximity and alignment with their neighboring connected components, thus contributing to a high edge weight. Therefore, the global tree from step 5 is used as an input to the $\alpha - \beta$ *graph pruning* function $\mathcal{F}$ which is defined as

$\mathcal{F}(\mathcal{M}) = \mathcal{M}_f$ where $\mathcal{M}_f = \{\mathcal{V}_f, \mathcal{E}_f\}, \mathcal{M}_f$ is connected, $\mathcal{M}_f \subseteq \mathcal{M}, \mathcal{V}_f = \mathcal{V} \setminus \mathcal{V}_N, \mathcal{V}_N = \{C_{i_1}, .., C_{i_k}\} \subseteq \mathcal{V}$ such that $\mathcal{V}_N = \arg\min_{\mathcal{V}_N \subseteq \mathcal{V}} |\mathcal{V}_N|$ satisfying $\frac{\sum_k \mathcal{W}(C_{i_k})}{\sum_j \mathcal{W}(C_j)} \geq \alpha$ and $\frac{d(\mathcal{M}_f)}{d(\mathcal{M})} \leq \beta$. Here $d(\mathcal{M})$ denotes the sum of edge weights of the tree $\mathcal{M}$.

In other words, the $\alpha - \beta$ graph pruning removes a minimal set of connected components from the tree $\mathcal{M}$ such that the proportional net node likelihood does not fall below α but the proportional net edge weight is reduced below β. The pruning thus achieves a tradeoff between highly likely nodes and highly likely connections. Figure 4(a-b) shows $\mathcal{M}_f$ calculated on the connected component set from image 2(b). In figure 4(b), if there is an edge between C_i and C_j in $\mathcal{M}_f$, a green edge connects the closest leaf pair from $\mathcal{M}_i$ and $\mathcal{M}_j$ in figure 4(b) obtained from (5) . The local medial trees $\mathcal{M}_i$ are shown in blue.

Global Node Numbering and Spline Fitting (**Step 7**): Finally, the individual medial trees $\{\mathcal{M}_j\}_{j=1}^{|\mathcal{V}_f|}$ from step 6 are merged together using the connectivity information from the global tree $\mathcal{M}_f$ to form the **final neuronal tree** $\mathcal{N} = \bigcup_{j=1}^{|\mathcal{V}_f|} \mathcal{M}_j$. A global node number is assigned to the nodes of the merged graph and cubic splines are fit to *individual* branches of $\mathcal{N}$ to generate the neuronal morphology. Figure

4(c) shows the final version of the spline fitted tree from figure 4(b).

3. RESULTS AND DISCUSSIONS

Results of applying *Tree2Tree* to Drosophila neuron parts in 2D images are described in detail in this section. In all our experiments, we have fixed $N_\tau = 0.15$, window size $w = 15$, neighborhood size k =10, filter parameters $\alpha = 0.9$ and $\beta = 0.4$.

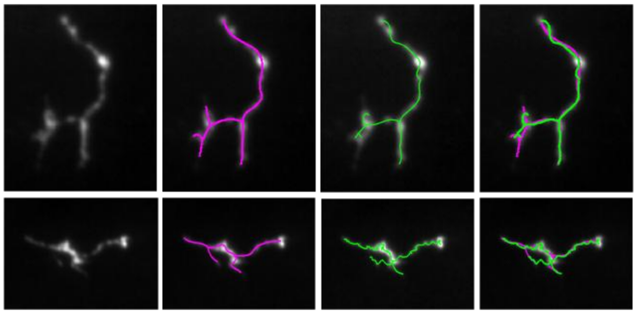


Figure 5: Top and bottom rows show the result of application of *Tree2Tree* to 2 sample neuron images. Leftmost column shows the original images, followed by the ground truth in magenta in the 2nd column, the automatic segmentation by *Tree2Tree* in the 3rd column, and the automatic results overlaid on the ground truth in the last column. The size parameter $N_\tau = 0.15$ in both cases, window size for local classification is $w = 15$, and the graph pruning parameters are $\alpha = 0.9$ and $\beta = 0.4$.

Figure 5 shows the final output of *Tree2Tree* compared with ground truth. It demonstrates the effectiveness of *Tree2Tree* in case of ambiguous boundaries (top row) and complicated and discontinuous filaments (bottom row).

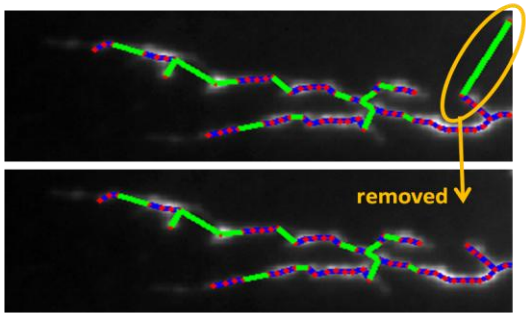


Figure 6: Top image shows a global graph connection before passing it through the $\alpha - \beta$ graph pruning. Bottom image results after passing through the filter with $\alpha = 0.9$ and $\beta = 0.4$. The unlikely edge that has been removed has been circled.

Figure 6 shows the application of the $\alpha - \beta$ graph pruning to a neuron global connectivity tree. Top image shows the unfiltered tree $\mathcal{M}$ (from step 5), and bottom image shows the filtered tree $\mathcal{M}_f$. The highly unlikely node with large edge distance and small weight has been removed after the filtering (shown in yellow).

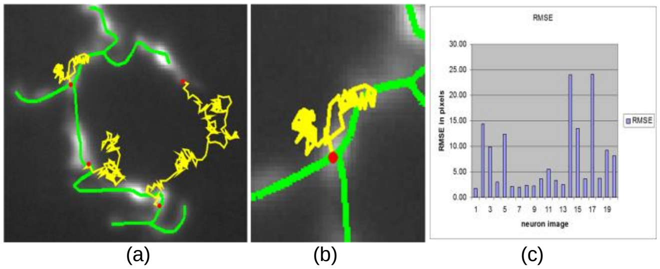


Figure 7: (a) *Tree2Tree* gives reliable segmentation of neuron in green, but method of [6] fails to successfully trace neuronal branches (traces shown in yellow) given 3 separate starting points in red. (b) Close-up of a failed trace by [6]. Low contrast and inconsistent neuronal thickness confounds the tracking process. (c) RMSE pixel errors of *Tree2Tree* on a set of neurons when compared to ground truth.

Figure 7(a-b) shows that *Tree2Tree* can easily estimate branches while the method of [6] has no automatic mechanism to do so. Furthermore the low contrast and irregular thickness of the neuronal filaments preclude successful tracking of the branches. Figure 7(c) shows RMSE pixel errors of *Tree2Tree* on a set of neuron images, with an average RMSE of 5.3 pixels over the set.

4. CONCLUSION

We have presented an automatic neuron segmentation and morphology generation algorithm that can reliably trace cluttered, low contrast, and irregularly shaped neurons. In future, we will use the morphology generated by this algorithm to compare similar neurons in a neuron database.

5. REFERENCES

[1] C. Koch, and I. Segev, “The role of single neurons in information processing”, *Nat Neurosci 3 Suppl*, pp. 1171-7, 2000.

[2] G. A. Ascoli, D. E. Donohue, and M Halavi, “NeuroMorpho.Org: a central resource for neuronal morphologies”, *J Neurosci*, vol 27, pp. 9247-51, 2007.

[3] G. A. Ascoli, “Neuroinformatics grand challenges**”,** *Neuroinformatics*, vol 6, pp. 1-3, 2008.

[4] J. G. White, E. Southgate, J. N. Thomson and S. Brenner, “The Structure of the Nervous System of the Nematode Caenorhabditis elegans,” *Philosophical Transactions of the Royal Society of London, Series B, Biological Sciences*, vol 314, pp. 1-340, 1986.

[5] B. D. Pfeiffer et al., “Tools for neuroanatomy and neurogenetics in Drosophila,” *Proc. Natl Acad Sci USA* 105, pp. 9715-20, 2008.

[6] K. A. Al-Kofahi, A. Can, S. Lasek, D. H. Szarowski, N. Dowell-Mesfin, W. Shain, J. N. Turner, and B. Roysam, “Median based robust algorithms for tracing neurons from noisy confocal microscope images”, *IEEE Trans. Information Technology in Biomedicine*, vol 7, no. 4, pp 302-16, 2003.

[7] A. Wolf, A. Herzog, S. Westerholz, B. Michaelis, and T. Voigt, “Improving fuzzy-based Axon segmentation with Genetic Algorithms: The IEEE Congress on Evolutionary Computation”, *Proc. IEEE Congress on Evolutionary Computation*, pp. 1025-31, 2009.

[8] H. Cai, X. Xu, J. Lu, J. Lichtman, S. P. Yung, and S. T. C. Wong, "Shape-constrained repulsive snake method to segment and track neurons in 3D microscopy images," *Proc. 3rd IEEE Intl. Symposium on Biomedical Imaging*, 2006.

[9] T. F. Chan and L. A. Vese, “Active Contours without Edges”, *IEEE Trans. on Image Processing*, vol 10(2), pp. 266-77, 2001.

[10] J. Chen, and B. G. Condron, “Branch architecture of the fly larval abdominal serotonergic neurons”, *Developmental Biology* (in press), (2008).

[11] R. Diestel, *Graph theory* (*Graduate Texts in Mathematics*), Springer, 3rd edition, 2006.

[12] H. I. Choi, S. W. Choi and H. P. Moon, “Mathematical Theory of Medial Axis Transform”, *Pacific Journal of Mathematics*, vol-181, pp. 57-88, 1997.