TREE2TREE2: NEURON TRACING IN 3D

Suvadip Mukherjee[1], *Saurav Basu*[3], *Barry Condron*[2], *Scott T. Acton*[1]

[1]Electrical & Computer Engineering, Biomedical Eng., University of Virginia, Charlottesville, VA USA
[2]Biology, University of Virginia, Charlottesville, VA USA
[3]Carnegie Mellon University, Pittsburgh, PA USA

ABSTRACT

We seek a complete description for the neurome of the *Drosophila*, which involves tracing more than 20,000 neurons. The currently available tracings are sensitive to background clutter and poor contrast of the images. In this paper, we present *Tree2Tree2*, an automatic neuron tracing algorithm to segment neurons from 3D confocal microscopy images. Building on our previous work in segmentation [1], this method uses an adaptive initial segmentation to detect the neuronal portions, as opposed to a global strategy that often results in under segmentation. In order to connect the disjoint portions, we use a technique called *Path Search*, which is based on a shortest path approach. An intelligent pruning step is also implemented to delete undesired branches. Tested on 3D confocal microscopy images of GFP labeled *Drosophila* neurons, the visual and quantitative results suggest that *Tree2Tree2* is successful in automatically segmenting neurons in images plagued by background clutter and filament discontinuities.

Index Terms— Biological image analysis, neuron segmentation, automatic tracing.

1. INTRODUCTION

Proper understanding of the brain requires an understanding of the morphology of the neurons. While the neural anatomy of the worm *C. elegans* has been fully analyzed [2], development of a set of neuronal atlases for the more complicated organisms like *Drosophila* and the mouse is still a field of ongoing research.

As far as image analysis is concerned, automated neuron segmentation still remains a critical hurdle. In the past few years, there has been significant contribution in this field ([3], [4], [5], [6], [7]). Poor contrast and low SNR values of the 3D neuron images pose difficulties, especially to achieve complete automation. Overall, the existing methods are sensitive to the background clutter and lack of contrast in the image, and often perform unpredictably in presence of significant background noise. Our approach extends the state-of-the-art by way of an adaptive initial segmentation, an optimized path searching process for merging neuronal components and an automated method for pruning spurious components.

In Section 2 of this paper, we discuss our method (*Tree2Tree2*). In Section 3 we compare the performance of our algorithm to the previous graph theoretic method [1] treating the solutions generated by the semi-automatic tracer *V3D* [5] as ground truth.

2. TREE2TREE2

Tree2Tree2 is an automatic neuron tracing method that does not rely on manually chosen seed points to initialize the tracing process. Figure 1 explains the different stages of *Tree2Tree2*. Salient novel features are highlighted in the figure by enclosing rectangles. The first stage involves a Hessian based enhancement method to calculate the vesselness measure $\mathcal{N}(\mathbb{p})$ for each voxel $\mathbb{p}$. Once we have the vesselness image that contains the neuron (and some background clutter), adaptive method is used to acquire an initial (disjoint) segmentation. Then, we use a two-step procedure to create the global neuronal tree: an initial global graph connectivity algorithm followed by a path search technique for efficient component merging and, finally, a specialized pruning process.

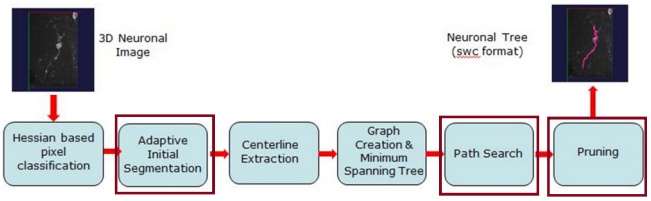


Figure 1: Schematic of *Tree2Tree 2*

2.1. Adaptive Initial Segmentation

The first step, as depicted in Figure 1, is an initial segmentation. The neuronal structure is assumed to be tubular, and a Hessian based analysis of the image isolates the neuron from the background that is typically plagued by blob-like clutter. Inspired by Frangi's method [8] to enhance tubular structures, we analyze the eigenvalues ($|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$) of the 3D Hessian matrix at each voxel position $\mathbb{p}$. We obtain a vesselness score by transforming the intensities in the confocal stack at a given point $\mathbb{p}$ (at Gaussian scale σ) to:

$$\mathcal{N}_\sigma(\mathbb{p}) = \begin{cases} \dfrac{|\lambda_1(\mathbb{p}) - \lambda_2(\mathbb{p})|^2}{(|\lambda_1(\mathbb{p})||\lambda_2(\mathbb{p}) - \lambda_3(\mathbb{p})|) + \epsilon} & \text{if } \lambda_2(\mathbb{p}), \lambda_3(\mathbb{p}) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

To capture tubes of different sizes, a scale space approach is used, where we look for the maximum vesselness response over a range of scales. The final vesselness measure of the image is given by

$$\mathcal{N}(\mathbb{p}) = \max_{\sigma_{min}<\sigma<\sigma_{max}} \mathcal{N}_\sigma(\mathbb{p}).$$

Since the fixed global thresholding technique of Otsu [9] is unreliable to segment the image in our spatially varying imaging scenario, we apply an adaptive technique based on surface evolution ([10], [11]) to obtain an initial segmentation.

If T is a threshold surface of an image I (i.e. the intensity levels above the surface T are treated as foreground), then the total energy functional E is written as:

$$E(T,\alpha) = \sqrt{1+\alpha^2}\, E_1(T) + \alpha\, E_2(T)$$

$\alpha \in [0,1]$ is a weighing parameter that regulates the two energy functionals E_1 and E_2 which are defined as

$$E_1(T) = \frac{1}{2}\iiint f(x,y,z)\big(I(x,y,z) - T(x,y,z)\big)^2\, dxdydz$$

$$E_2(T) = \frac{1}{2}\iiint |\nabla T(x,y,z)|^2 dxdydz$$

$\nabla T(.)$ denotes the gradient of the threshold surface. The functional $E_2(T)$ regulates the shape of the threshold surface and discourages sharp changes of the surface profile. Choosing the indicator function $f(x,y,z) = |\mathcal{N}(x,y,z)|^p$ makes it high at locations in which the vesselness score dominates. p is a parameter that determines the contribution of the indicator function. It is evident that in order to minimize $E_1(T)$ at points where $f(.)$is high, the threshold surface T needs to be close to the image surface I.

The motivation behind the adaptive segmentation technique is as follows: An analysis of the energy functional $E(T,\alpha)$ reveals that it is concave w.r.t. α and convex w.r.t. T. Here α regulates the influence of the regularization term $E_2(T)$ and the data term $E_1(T)$. If the regularization term dominates, the threshold surface is smooth, and hence results in under segmentation, whereas a low contribution of the regularization term produces a non-smooth surface with more sensitivity to clutter. The optimal α can be obtained by using a minimax method, to get the optimal threshold surface T'as

$$T' = \arg\max_{\alpha}\min_{T} E(T,\alpha).$$

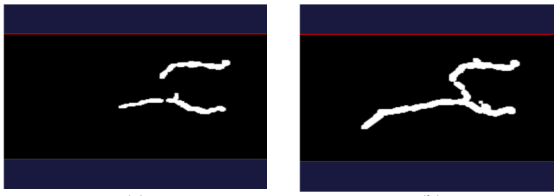
Solving the above mini-max optimization problem, we obtain the maximum value of α as $\alpha' = \frac{E_2(T)}{\sqrt{E_1(T)^2+E_2(T)^2}}$. The evolution of the threshold surface T,w.r.t. time (τ), can be obtained by using gradient decent technique as follows ([11]):

$$\frac{\partial T}{\partial \tau} = \sqrt{1-\alpha'^2 f(.)\big(I(.)-T(.)\big) + \alpha'\big(\nabla^2 T(.)\big)}.$$

Note that the method does not use any tuning parameters and is completely automated. The convexity of the energy functional ensures convergence. Figure 2 provides evidence that the adaptive initial segmentation outperforms the global thresholding method.

2.2. Path Search

The initial segmentation procedure classifies the image voxels into neuronal (foreground) and non-neuronal (background) portions. Since the obtained foreground segments may be disjoint, the next step is to connect the components to obtain a global graph, and eventually, the neuronal tree. To ensure proper connectivity, we adopt a two stage approach. Let us denote the set of disjoint components as $C = \{C_1, \dots, C_n\}$.



(a) (b)

Figure 2: Otsu's method (a) vs. adaptive segmentation (b). The adaptive method successfully identifies the neuronal segment.

The first step involves a global graph connectivity algorithm that decides on the connectivity between these components based on their relative distance and orientation [1]. A minimum spanning tree of the global graph produces an initial neuronal tree. This method is computationally simple and provides efficient and accurate connectivity information between the components. The objective now is to detect the actual path that connects them in the 3-D image. (Figure 3).

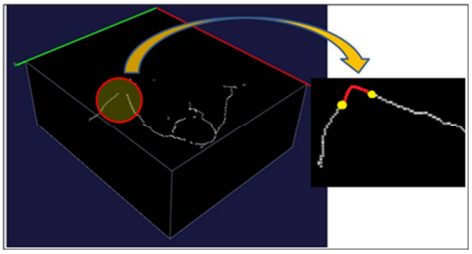


Figure 3: Two disjoint components are joined by the global connectivity algorithm (shown in red). But the actual path is unknown.

The Path Search algorithm to find the actual path between two disjoint components is essentially a dynamic programming approach to find the shortest path between two voxels. Let $\mathcal{N}$ be the 3D vesselness image of size $M \times N \times D$, i.e. there are D number of slices per stack, each slice being $M \times N$ in dimension. Let s and t be two predefined voxels in $\mathcal{N}$. We are interested in finding the best path connecting s and t.

Let $G = (V,E)$ be an undirected connected graph (where V is the set of vertices and E is the set of edges) with no self-loops and no negative edge cycles. The weight of a path $p =< v_0, v_1, \dots, v_k >$ is defined as $W(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$, where w: E $\rightarrow \mathbb{R}$ is the edge weight function. The shortest path between the node pair (v_i, v_j) is the path $P =< v_i, \dots v_j >$ having a path cost $\delta(v_i, v_j) = min\big(W(p)\big), \forall$ paths p $b$etween v_i and v_j. In our case, the

nodes v_i and v_j correspond to the leaf voxels s and t of two components $(C_i, C_j) \in C$. A path connecting s and t is represented by the set of voxels $V =< v_1, \dots, v_n >$, where $v_1 = s$ and $v_n = t$. The intermediate points are obtained by finding the most likely path between s and t in a manner similar to finding the shortest path between two nodes in a graph by *Dijkstra's* algorithm [12]. To solve this problem, we need to define the graph $G = (V, E)$ and the edge weight function w. We construct the graph in the following way:

$V = \{v = (i, j, k): 1 \le i \le M, 1 \le j \le N \; and \; 1 \le k \le D\}$

$$E(v_i, v_j) = \begin{cases} 1 & \text{if } v_i \in \aleph(v_j) \\ 0 & \text{otherwise} \end{cases}$$

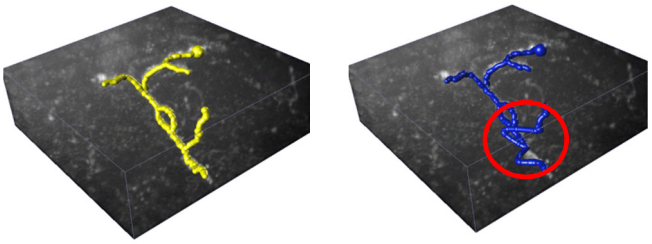
$$w(v_i, v_j) = \begin{cases} \dfrac{1}{\lambda + \dfrac{\mathcal{N}(v_i) + \mathcal{N}(v_j)}{2}} & \text{if } E(v_i, v_j) = 1 \\ \infty & \text{otherwise} \end{cases}$$

Here $\aleph(\mathrm{v_j})$ denotes the set of neighbors of a voxel v_j in the 26-connectivity sense. λ is a constant.

This graph has the following properties

(a) G is a connected graph, i.e. $\forall\, v_i, v_j \in V,\ \exists$ a path $\mathcal{P} = < v_i, \dots, v_j >$ between (v_i, v_j), with path cost $W(\mathcal{P})$.

(b) The neuronal portions in the image are characterized by relatively higher vesselness measure than the background. Hence if a voxel v_{int} belongs to the neuron and v_{ext} is a background voxel, $\mathcal{N}(v_{int}) > \mathcal{N}(v_{ext})$. Hence, $w(v_i, v_j) < \mathrm{w}(u, v)$ if both v_i and v_j are neuron voxels and either one or both of u and v belongs to the background.



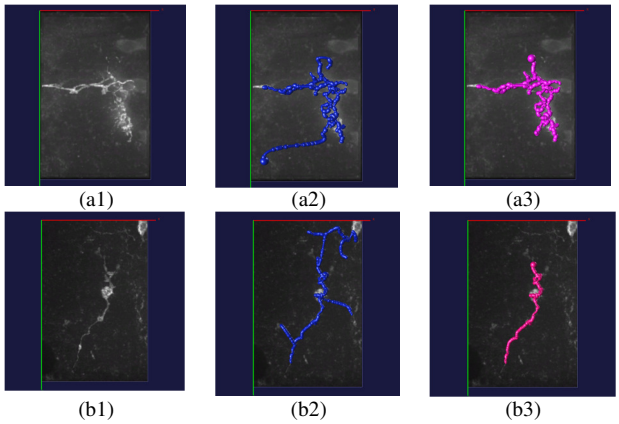
(a) (b)

Figure 4: Comparison of Path Search with spline fitting. (a) The obtained segmentation by using path search. (b) The segmentation result corresponding to the spline fitting approach. The red circle in (b) shows the region where the spline fitting algorithm finds improper paths, which is rectified by the path search algorithm in (a) .

For each edge in the initial neuronal tree connecting two components $(C_m, C_l) \in C$, the path search algorithm determines the shortest path (in the 3-D image) $P_{ml} =< v_m', \dots, v_l' >$, with path cost $W(P_{ml})$. We can now construct the global neuron tree $\Gamma = (V_T, E_T)$, where $V_T = \{C_1, \dots, C_n\}$, the set of disjoint components, $E_T = \{P_{ml}\}$, the set of shortest paths connecting the components with path cost $W(P_{ml})$, $\forall$ paths $P_{ml} \in E_T$.

2.3. Intelligent Pruning

Although our algorithm requires minimal human intervention, a pruning step provides the user some control over the tracing process. This step is necessary to eliminate the false nodes which may have appeared due to the presence of background clutter. The Path Search algorithm creates a global neuronal tree Γ. We designate the volume occupied by a tree in the 3D binary image as $\psi(\Gamma)$. Assuming that the neuronal portions occupy more volume than the clutters, which are present as sporadic blob like structures, we devise the following method to prune the tree:



(a1) (a2) (a3)

(b1) (b2) (b3)

Figure 5: Pruning results. (a1) and (b1) are the MIP images of two neurons. The non-pruned raw neuron trees are shown in (a2) and (b2). (a3) and (b3) are the final results obtained after pruning ($\boldsymbol{\varepsilon} = 0.1$).

At iteration t, let $e \in E_T$ be the heaviest edge, i.e. $w(e) \ge w(e')\ \forall\, e' \in E_T$. Removing this edge from the tree Γ creates two disconnected subtrees $\Gamma_1$and Γ_2. Let Γ_i be the subtree such that $\psi(\Gamma_i) = \min(\psi(\Gamma_1), \psi(\Gamma_2))$, $i = 1,2$. Γ_i is deleted from the tree if $\frac{\psi(\Gamma_i)}{\psi(\Gamma)} \le \varepsilon$. So, a subtree is deleted if it is connected by a significantly high cost path, and the volume occupied by the subtree is small compared to the overall volume. Setting ε to a low value ($\sim$0.1) encourages removal of clutters over a neuron segment (Figure 5).

3. RESULTS AND DISCUSSION

The dataset used in the experiments consists of one neuron

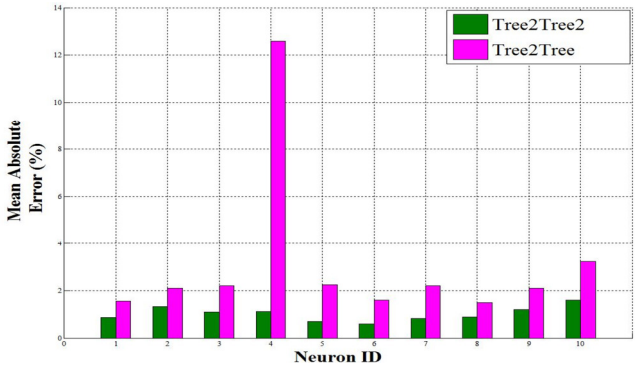


Figure 6: Mean absolute error (%) of *Tree2Tree* (pink) and *Tree2Tree2* (green) is computed against the tracing result of *V3D* (ground truth).

per image from a collection of 3-D stacks of confocal microscopy images generated in the Condron lab at UVA. We have compared our results against our previous approach *Tree2Tree* [1]. The manually traced neurons using *V3D* [5] serves as the ground truth against which the Mean Absolute Error (MAE) [1] of the segmented neuron is computed (see Figure 6). It is observed that in all the cases, *Tree2Tree2* outperforms *Tree2Tree*.

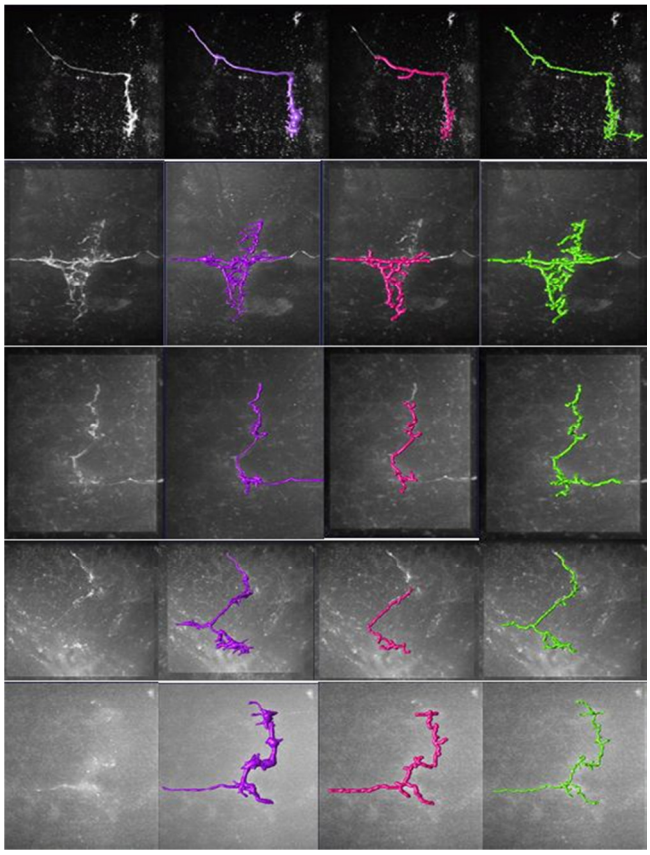


Figure 7: Comparative results: The first column from left shows the MIP images of some neurons from our dataset. The second column shows the tracing results of *V3D* (violet). The third column shows the output using the previous version of *Tree2Tree* (pink). The last column is the final tracing result of the current implementation (green).

Visual comparison of the tracing results of five (out of ten) neurons in our dataset (Figure 7) reflects that *Tree2Tree 2* is successful in detecting branches and neuron segments in cases where the previous implementation fails.

Even though the semi-automated tracing via $V3D$ provides a reasonable basis for ground truth, this seed-based segmentation can result in erroneous self-connections, as witnessed in the bottom row of Figure 7. In the future, we plan on investigating the potential of electron microscopy data as ground truth.

4. CONCLUSION

In this paper we have presented a robust automatic neuron tracer. The initial adaptive segmentation and the two stage technique to connect the components contribute to the robustness of *Tree2Tree2*. Undesired nodes can be removed by the intelligent pruning method that uses only single parameter ε (as opposed to two parameters in *Tree2Tree*). The visual and quantitative comparison results suggest that *Tree2Tree2* performs better than *Tree2Tree* in non-ideal images. The traced neuron is represented by a graph theoretic tree, which serves as the mathematical representation of its morphology. This is useful to compare the morphological structures of two neurons, which in turn leads to a technique to retrieve morphologically similar neurons from a database.

5. REFERENCES

[1] S. Basu, A. Aksel, B. Condron and S. T. Acton, "Segmentation and tracing of neurons in 3D," *IEEE Transactions on Info. Tech. Biomedicine,* in press,2012.

[2] 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.

[3] M. H. Longair, D. A. Baker and J. D. Armstrong, "Simple Neurite Tracer: Open Source software for reconstruction, visualization and analysis of neuronal processes," *Bioinformatics,* vol. 27, no. 17, pp. 2453-2454, 2011.

[4] E. Meijering, "Neuron Tracing in Perspective," *Cytometry Part A,* vol. 77, pp. 693-704, 2010.

[5] H. Peng, Z. Ruan, F. Long, J. Simpson and E. and Myers, "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnology,* vol. 28, pp. 348-353, 2010.

[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, pp. 302-16, 2003.

[7] J. Lu, J. Fiala and J. L. Lichtman, "Semi-automated reconstruction of neural processes from large numbers of fluorescence images," *PLoS One,* vol. 4, 2009.

[8] A. Frangi, W. Niessen, K. Vincken and M. Viergever, "Multiscale vessel enhancement filtering," *MICCAI,* vol. 1496, pp. 130-137, 1998.

[9] N. otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man and Cybernetics,* vol. 9(1), pp. 62-66, 1979.

[10] F. Chan and H. Z. F.K. Lam, "Adaptive thresholding by variational method.," *IEEE Trans Image Process. ,* pp. 468-473, 1998.

[11] B. N. Saha and N. Ray, " Image Thresholding by Variational Minimax Optimization," *Pattern Recognition,* 2008.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, Second ed., The MIT Press.