Automatic Neuron Tracing in Volumetric Microscopy Images with Anisotropic Path Searching

Jun Xie, Ting Zhao, Tzumin Lee, Eugene Myers*, and Hanchuan Peng*

Janelia Farm Research Campus,
Howard Hughes Medical Institute, Ashburn, Virginia USA

Abstract. Full reconstruction of neuron morphology is of fundamental interest for the analysis and understanding of neuron function. We have developed a novel method capable of tracing neurons in three-dimensional microscopy data automatically. In contrast to template-based methods, the proposed approach makes no assumptions on the shape or appearance of neuron's body. Instead, an efficient seeding approach is applied to find significant pixels almost certainly within complex neuronal structures and the tracing problem is solved by computing an graph tree structure connecting these seeds. In addition, an automated neuron comparison method is introduced for performance evaluation and structure analysis. The proposed algorithm is computationally efficient. Experiments on different types of data show promising results.

1 Introduction

The tasks of extracting neuron structures from microscope images and accurately measuring their topologic properties are important for a large number of studies in biology and medicine. Although this is an extensively studied problem in the literature and there are some commercial products available, no automated method has promised reconstruction that requires no or few manual corrections. This prevents the implementation of high-thoroughput neuron structure analysis, without which biologists can be easily overwhelmed by their data.

Most existing methods are based upon the fact that a neuron is a branching tree structure. They formulate the problem of reconstructing a neuron as determining where the branches go and how they bifurcate. For this reason, the problem is also called neuron tracing. The branches can extend in any direction in the three-dimensional (3D) space. Therefore a successful strategy for realistic tracing applications has to operate in 3D. In 1994, Cohen *et al.* [1] introduced a 3D tracing method based on skeletonization and graph extraction. The same strategy was applied by Koh *et al.* [2] but with a modified medial axis to locate dendritic branch centerlines. However, 3D skeletonlization is computationally expensive and the noise in the microscopy data presents a significant challenge

* Equal senior authors.

T. Jiang et al. (Eds.): MICCAI 2010, Part II, LNCS 6362, pp. 472–479, 2010.
© Springer-Verlag Berlin Heidelberg 2010

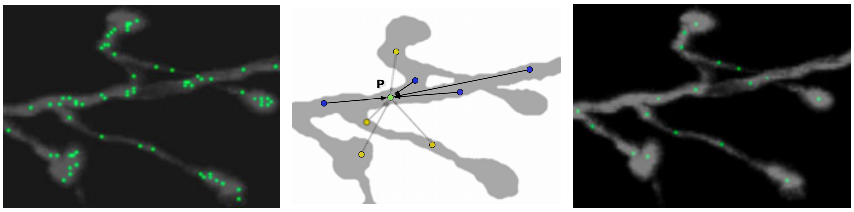

Fig. 1. An example of seeding. (Left) Initial seed candidates (green) from local maximum. (Middle) The diagram showing seed refinement. The ray between one candidate to candidate p determines whether it is an valid voter (blue) or not (yellow). (Right) The final seeds for tracing.

for the thinning operation. Based on structure continuity, parametric models were proposed for filament tracing [3]. Alkofahi *et al.* [4] proposed applying predetermined deformable templates to detect the local dendritic boundary and then performing an exploratory tracing of the dendrite. He *et al.* [5] proposed combining local color-based estimates of the probability that an image pixel belongs to a filament with the global tree properties of the complete set of filaments. If the neurite structures in the stack are all connected, template based tracing works well. However, for common realistic cases, the structures of interest appear to be discontinuous due to imaging conditions and noise. An optimization scheme based on the Minimum Spanning Tree (MST) [6] is usually applied to tackle this challenge.

2 Method

In this work, we follow the optimal tree idea to reconstruct the 3D topology of neuronal structures. Similar to the human operator's exercise, our method first detects a set of seeds which should be dense enough to describe the major characteristics of the neuronal structures, but also be considerably compact for redundancy control and computational efficiency.

Optimal Seeding. The ideal set of seeds consists of the points on the underlying objects which stands for a good representative to each neurite segment. Before detecting seed points, a global thresholding on the input 3D data is applied to remove noise (isolated particles) and save computation time. Based on the fact that noise often appears as small local maximal regions in the background area of microscope images, the global threshold is obtained by applying the triangle-thresholding method [7] on the histogram of local intensity maxima in the input image.

Our seeding method consists of two steps. In the first step, candidate seeds are detected by searching local intensity maixma. The local window is a cube on the stack grid and only one candidate is identified in each valid cube. A cube is considered as valid only if it contains sufficient foreground voxels which can be measured with the signal-to-noise ratio $\text{SNR}_c = (\mu_f - \mu_b)/\sigma_b$, where μ_f is the mean intensity value of the foreground in the cube, μ_b and σ_b denotes

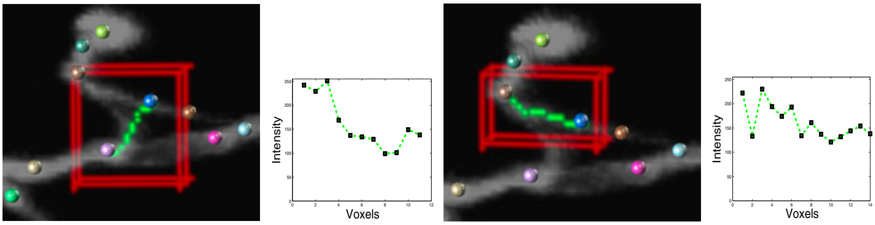

Fig. 2. An example of path searching. (Left) The strongest path detected for seed #37 (blue) with the isotropic weight function in Eq. (1). The corresponding intensity profile of the path is shown beside. (Right) The strongest path detected with the anisotropic weight function in Eq. (2) which favors smoothness.

the background's mean and standard deviation, respectively. In order to remove the high-frequency components and improve the detection accuracy, a Gaussian kernel is applied to filter the local intensity maxima. An example of this initial sampling is presented in Fig. 1(Left).

Secondly, a decimation operation is applied to those candidates for final reliable seeds. The local scale of the neurite structure is first estimated with the distance transformation [8]. Then for candidates with lower scale values, indicating they are isolated speckles or close to boundary, votes from other candidates are collected for refinement decision. In specific, for candidate p which has the lowest scale in the set, other candidates can recommend for elimination if they have a strong ray connection R to p ($\bar{I}_R > I_t$) (Fig. 1(Middle)). Candidate p will be removed from the set if there are recommendations from diverse orientations (*e.g.*, the maximum included angle of the voter rays is considerably large). This condition lets us avoid eliminating terminal seeds mistakenly. This seeding approach is robust to the intensity variation and leads to varying seeding density (Fig. 1(Right)) according to the local complexity of the structures, thus simplifying the tracing problem.

Linking Strength among Seeds. With the representative seeds detected above, tracing neuronal structures can be considered as finding the optimal connections of those seeds. Considering the image grid as a discretized weighted grid graph $G = (V, E, W)$, where vertex set V consists of the grid points, E contains the regular straight line segments in the grid and W are some features associated with the segments. By modeling the stack grid as a graph, one voxel is connected with its 26-connected neighbors. Given a pair of nodes v_0 to v_n, a path between them is a list of unique nodes. The cost of a path is the sum of each edge weight in the path.

Let $\mathcal{P}$ denote the set of all unit-speed paths in the search region $\mathcal{R}$ from v_0 to v_n, *i.e.*, the set of functions $\ell : [0, N] \rightarrow \mathcal{R}$, for which $\ell(0) = v_0$ and $\ell(N) = v_n$. The formulation of finding a shortest path between two vertices is to compute a path $\ell^* \in \mathcal{P}$ such that $\mathcal{W}(\ell^*) = \min_{\ell \in \mathcal{P}} \mathcal{W}(\ell)$, where $\mathcal{W} : \mathcal{P} \rightarrow [0, \infty]$ denotes the overall cost of the path: $\mathcal{W}(\ell) = \sum_{t=0}^{N} \rho(\ell(t))$, for each ℓ in $\mathcal{P}$. One of the efficient algorithm to find shortest path on the sparse graph is the one introduced

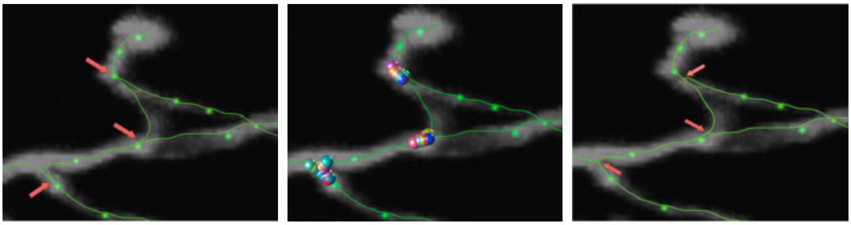

Fig. 3. Path Merging. (Left) Initial paths among seeds. (Middle) The 'touching' positions between nearby paths. (Right) The updated paths with the new branching locations (red arrows).

by Johnson [9] by storing the path lengths with a priority queue. The complexity of Johnson's algorithm is $O(|E|\log|V|)$.

The cost weighting function ρ varies according to the voxel's relative positions $\rho(v_i) \propto d(v_i)$ where $d(v_i) = |v_i - v_{i+1}|$. To emphasize the difference between foreground and background regions, a region-based cost term $\mathcal{I}$ is incorporated in the weighting function as:

$$\mathcal{W}(\ell) = \sum_{i=0}^{N-1} d(v_i)\mathcal{I}(v_i) = \sum_{i=0}^{N-1} d(v_i)\frac{\mathcal{S}(v_i) + \mathcal{S}(v_{i+1})}{2}, \qquad (1)$$

where $\mathcal{S}(\cdot)$ is a Sigmoid-filtered version of the input stack which generates a large weight value if two voxels belong to different regions, indicating the path is crossing a significant boundary.

One limitation of the above cost-weighting function is that it considers only situations where the metrics are isotropic. It cannot account for the inherent properties of neurite structures and doesn't provide a smoothing constraint for the path searching. In our case, the neurite structures have a tree-liked topology and most of the components are of considerable smoothness. One way to apply this prior knowledge is to search paths with the isotropic metric first and then apply a post-processing on the resulting graph. This strategy is straightforward but error-prone. See Fig. 2 for an example. A more efficient way to solve this problem is to introduce a smoothness constraint into the weighting function as:

$$\mathcal{W}(\ell) = \sum_{i=0}^{N-1} d(v_i)\frac{\mathcal{S}(v_i) + \mathcal{S}(v_{i+1})}{2}\mathcal{K}(v_i), \qquad (2)$$

where term $\mathcal{K}(v_i)$ is a smoothness measurement of path segment defined as

$$\mathcal{K}(v_i) = \sum_{t=0}^{i} \gamma^t k_t = \sum_{t=0}^{i} \gamma^t \frac{\overrightarrow{v_{t-1}v_t} \cdot \overrightarrow{v_t v_{t+1}}}{|\overrightarrow{v_{t-1}v_t}||\overrightarrow{v_t v_{t+1}}|}, \qquad (3)$$

and γ is a constant determining the influence scale of a grid point to its neighbors. Term $\overrightarrow{pq}$ denotes the vector from p to q. With this incremental smoothness term, the orientation of the previous path flow and the bending force will be involved to determine the cost of the new extension. With the anisotropic weighting function in Eq. (2), the path detected between each pair of the seeds would be a short and smooth path riding on regions with similar properties (the foreground in our case). The total cost of the path is defined as the linking strength between the pair of seeds.

Computing the Optimal Tree. Once the linkage paths between each pair of seeds are determined, a weighted graph $\mathcal{N}(S, C, L)$ is constructed, where set S contains all the seeds. Set C consists of the optimal paths $\ell_{ij}(s_i, s_j \in S)$ with the associated edge weights in L which could be the corresponding linking strength computed above. Then a globally optimal tree with the minimum sum of edge distances can be discovered by the MST algorithm. In order to avoid the misconnection between close seeds with a sharp dark gap, a gap indicator function is incorporated in the the final vertex distance function as:

$$\mathcal{F}(\ell) = \alpha\mathcal{W}(\ell) + (1 - \alpha)\mathcal{D}(\ell), \tag{4}$$

where $\mathcal{D}$ is a gap indicator function defined as

$$\mathcal{D}(\ell) = 2/(1 + e^{I_m(\ell)/\bar{I}}). \tag{5}$$

Term I_m denotes the minimum intensity along the path and $\bar{I}$ is the mean intensity of the foreground. All the metrics are normalized for the combination.

In realistic applications, it cannot be guaranteed that there would be seeds in the critical locations of the neurite structures such as bifurcations or terminals. Actually, it is difficult to detect the branching locations in the 3D microscopy data due to the various size of structures, imaging resolution and noise. Instead of pursuing detecting those positions directly from the images, our method solves this challenge by considering the paths between the seeds. As demonstrated in Fig. 3, there are multiple paths near the bifurcation positions that are very close to each other. This indicates a converge in those locations. By merging those 'touching' path segments, more accurate bifurcation nodes are derived.

3 Experimental Results

We tested our method on 24 stacks consisting of three different data sets. The first data set consists of 16 3D stacks of projection neurons of adult Drosophila with GH146-Gal4 carrying UAS>CD2, y^+ >CD8-GFP and Hs-flp transgenes. The CD2 Flip-out cassette is removed with a mild heat shock in the third instar larvae and individual neurons are visualized with anit-GFP. The second testing set contains five confocal microscopy stacks of drosophila olfactory axonal projection neurons labelled with GFP. The last data set consists of three confocal stacks showing the arborizations of individual neurons with the main presynaptic sites (button-liked structures) in the fly lamina component. Most of the stacks have a clear view of a single neuron in a cluster. Figure. 4 demonstrates one example from each of the testing data sets. For all tests, we set $\gamma = 0.9$ and $\alpha = 0.7$.

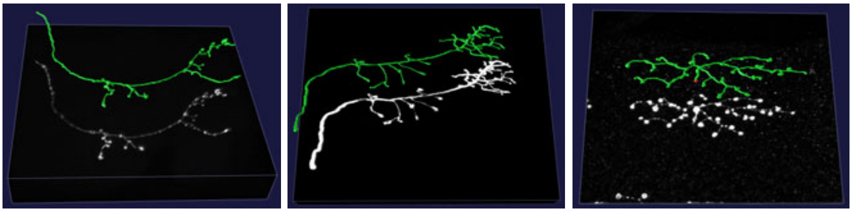

Fig. 4. (Left) Projection Neuron of adjust drosophila. (Middle) Confocal section of drosophla olfactory axonal projection. (Right) Confocal section shows the aborisations of neurons in the lamina of drosophila.

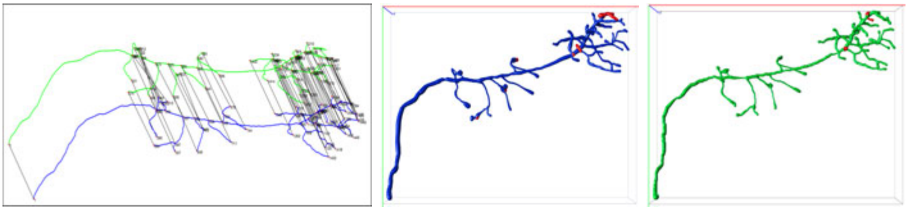

Fig. 5. An example of matching between two traces. Only bifurcation and terminal matching are demonstrated for clear view (Right). The red components in (Middle) and (Right) show the unmatched parts on the ground-truth and automatic result, respectively. In this case: MES = 0.93, $ADE(xy_d, z_d) = (5.54 \pm 1.180, 2.24 \pm 0.431)$.

To quantitatively study the performance of the tracing system, it is essential to select reliable metrics to validate the automatic traces in comparison with some gold-standard or ground-truth such as manual annotation. In our experiment, the ground-truth is generated by different people or by the same person at different times with the aid of the visualization tool V3D [10] . Instead of applying the average euclidean distance in [11], which is intuitive but vulnerable to the biocomplexity, we pursue the correspondences between key nodes in the neurite structure (such as branchings and terminals) that are of the major interest. In particular, for one node p_i, consider the set of paths in the trace originating from p_i to all other nodes. These paths provide a rich description of the entire topology of the trace relative to p_i. Assuming there are sufficient nodes in the trace, this representation is highly precise.

For computational efficiency, two features are calculated to describe each path ℓ_{ij}, including the accumulated length and the included angle between vector $\overrightarrow{p_ip_r}$ and $\overrightarrow{p_ip_j}$, where p_r refers to the root node. The statistics of those two features is a histogram h_{p_i} counting the relative locations of the other nodes regarding to p_i. Then given two different traces $\mathcal{T}_1$ and $\mathcal{T}_2$, the matching cost $\mathcal{C}(p, q)$ of two nodes $p \in \mathcal{T}_1$ and $q \in \mathcal{T}_2$ is estimated by comparing their descriptors using the χ^2 test. In all the experiments demonstrated in this paper, we choose 20 bins for included angle and 10 bins for path length respectively. Therefore, the total number of bins for each histogram is $M = 200$. This descriptor is robust since it is translation and scale invariant. In order to maintain the spatial ordering of those nodes, the

Table 1. The Evaluation on Automatic Tracing for Different Data Sets

Dataset	Avg. Size	# Stacks	MES(std.)	XY-dist(std.)	Z-dist(std.)	Time(s)
1	512x512x70	16	0.87(0.073)	3.42(1.402)	1.18(0.403)	24.6
2	512x512x73	5	0.86(0.042)	2.36(0.723)	0.76(0.181)	28.1
3	1024x1024x32	3	0.99(0.011)	1.04(0.867)	0.45(0.588)	56.2
Overall	N/A	24	0.89(0.074)	2.90(1.194)	1.00(0.380)	29.3

dynamic programming is applied to find the optimal matching π which minimizes the total matching cost: $\mathcal{Z}(\pi) = \sum_{\substack{p\in\mathcal{T}_1 \\ \pi(p)\in\mathcal{T}_2}} \mathcal{C}(p, \pi(p))$, where $\mathcal{C}(p, 0) = \eta$ is the skipping penalty which can be an instant or be determined based on the size of lost branches (directly from p). By maintaining the spatial ordering, crossing or overlapping matches are avoided. The resulting corresponding critical nodes then serve as a set of landmarks, partitioning the trace into independent segments for further analysis. Fig. 5 illustrates an example of the matching results.

With the established correspondences, we calculate two metrics to evaluate the automated tracing results. One is the Miss-Extra-Score (MES) which is defined as MES $= (S_G - S_{miss})/(S_G + S_{extra})$, where S_G is the total length of all segments in the Ground-Truth trace, S_{miss} and S_{extra} are the total lengths of missing and extra segments in the automatic trace respectively (compared with the Ground-Truth). MES provides a global view on how many parts of the neuron have been traced and how many undesired components are introduced by the automatic tracing program.

Another metric is the Average-Displacement-Error (ADE) which is defined as the average displacement of those matched components. It is a numerical measurement to describe the local accuracy of the automatic trace. ADE is valid based on the assumption that the ground-truth and automatic trace have the same root position. Because the resolution on Z-dimension is usually lower than that on XY-dimension in microscopy data, the measurement is calculated in xy- and z- dimension separately.

The results on each tested stack are plotted in Fig. 6. The majority of failure cases are found in data with blur or broken dendritic fragments, either due to insufficient dye penetration or an imaging deficiency. Table 1 summarizes the

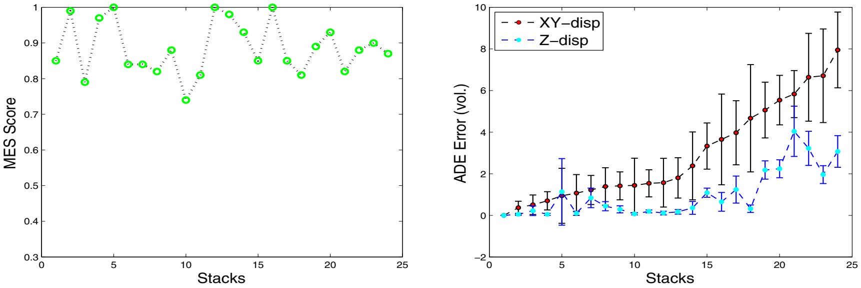

Fig. 6. The comparisons between automatic traces with the ground-truth. (Left) The MES score. (Right) The ADE error measured in XY and Z dimension respectively.

overall performances on the three testing data sets. Compared to the manual annotations, the automated 3D tracing system yielded overall MES of 0.89. The overall ADE is 2.90 and 1.00 on xy- and z- dimension respectively. The computation time varies with the stack size and the density of the neuron network. For the testing data, the average computation time is about 30s. Our system is implemented in C++ and the average computation time is obtained on a Mac Pro with 2.8GHz intel Xeon CPU and Leopard operation system.

4 Conclusion

We have developed an automated system capable of tracing neuronal structures in 3D microscopy data. Incorporating anisotropic path searching and merging, this approach is efficient for neuron tracing and bifurcation localization. In addition, a robust method is introduced for trace comparison, which can also be used for neuron recognition and structure analysis in other applications. As demonstrated in the paper, the proposed method can be applied to a variety of biomedical image tracing problems because of the robustness of global optimality and the flexibility to design a problem-specific objective function.

Acknowledgments. We thank Aljoscha Nern for generating the presynaptic site brain images and Margaret Jefferies for help of text editing the manuscript.

References

1. Cohen, A.R., Roysam, B., Turner, J.N.: Automated tracing and volume measurements of neurons from 3d confocal fluorescence microscopy data. Journal of Microscopy 173(1), 103–114 (1994)
2. Koh, I.Y.Y., et al.: An image analysis algorithm for dendritic spines. Neural Comput. 14(6), 1283–1310 (2002)
3. Shen, H., et al.: Optimal scheduling of tracing computations for real-time vascular landmark extraction from retinal fundus images. IEEE Trans. Inform. Technol. Biomed. 5, 77–91 (2001)
4. Al-kofahi, K.A., et al.: Rapid automated three-dimensional tracing of neurons from confocal image stacks. IEEE Trans. Inform. Technol. Biomed. 6, 171–187 (2002)
5. He, W., et al.: Automated three-dimensional tracing of neurons in confocal and brightfield images. Microscopy 9, 296–310 (2003)
6. Aho, A.V., Ullman, J.D., Hopcroft, J.E.: Data Structures and Algorithms. Addison-Wesley, Reading (1985)
7. Zack, G.W., Rogers, W.E., Latt, S.A.: Automatic measurement of sister chromatid exchange frequency. J. Histochem Cytochem 25(7), 741–753 (1977)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science (2004)
9. Johnson, D.B.: Efficient algorithms for shortest paths in sparse networks. J. ACM 24(1), 1–13 (1977)
10. Peng, H., et al.: V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. Nature Biotechnology 28(4), 348–353 (2010)
11. Al-kofahi, K.A., et al.: Median-based robust algorithms for tracing neurons from noisy confocal microscope images. IEEE Trans. Inform. Technol. Biomed. 7, 302–317 (2003)