

Automatic 3D Single Neuron Reconstruction with Exhaustive Tracing

Zihao Tang¹, Donghao Zhang^{1,*}, Siqi Liu¹, Yang Song¹, Hanchuan Peng², and Weidong Cai^{1,*}

¹School of Information Technologies, University of Sydney, NSW, Australia

{ztan1463, dzha9516*}@uni.sydney.edu.au, {siqi.liu, yang.song, tom.cai*}@sydney.edu.au

²Allen Institute for Brain Science, Seattle, WA, United States

hanchuanp@alleninstitute.org

Abstract

The digital reconstruction of neuronal morphology from single neurons, also called neuron tracing, is a crucial process to gain a better understanding of the relationship and connections in neuronal networks. However, the fully automation of neuron tracing remains a big challenge due to the biological diversity of the neuronal morphology, varying image qualities captured by different microscopes and large-scale nature of neuron image datasets. A common phenomenon in the low quality neuron images is the broken structures. To tackle this problem, we propose a novel automatic 3D neuron reconstruction framework named exhaustive tracing including distance transform, optimally oriented flux filter, fast-marching and hierarchical pruning. The proposed exhaustive tracing algorithm shows a robust capability of striding over large gaps in the low quality neuron images. It outperforms state-of-the-art neuron tracing algorithms by evaluating the tracing results on the large-scale First-2000 dataset and Gold dataset.

1. Introduction

Neuron morphology is a crucial topic in neuroscience [2]. The morphological property of the axons and dendrites is a key determinant in neuronal network connectivity [1, 19]. The quantitative analysis allows neuroscientists to study the relationship between neuron morphology and corresponding function [6], and this can be applied for reverse-engineering to achieve a better understanding of brain functions [30].

Neuron reconstruction is a process to extract the neuron morphology from single neuron images captured by optical microscopes with different neuron labeling techniques. These techniques include bulk dye loading [12], intracellular injections [10], immunilabeling [20] and genetic la-

beling [18]. The neuron tracing method consists of distinguishing the foreground neuron signals from the background, finding the main skeleton of the neuron, linking broken patterns and pruning redundant structures [7].

The traditional approach of neuron tracing is to label the neuron manually with the assistance of computers [22]. Due to the large-scale characteristic of the neuron image datasets, reconstructing the neuron morphology manually is an extremely laborious and time-consuming task and it is not practicable for large datasets. Hence, many semi-automatic or automatic neuron tracing algorithms have been proposed in recent years to deal with this problem. Open-curve Snake [37, 35, 9] preprocesses a gradient vector flow to force the tracing algorithm to move along the centreline of the neuron bidirectionally. MOST algorithm [21] is based on the Rayburst sampling [29] using prediction and refinement strategy. All Path Pruning algorithms, which including APP1 [25] and APP2 [36], build an over-reconstruction neuronal tree then prune the redundant branches and nodes based on the coverage ratio. Based on the result of All Path Pruning algorithms, Smart tracing [5] uses wavelet embryo feature to perform the feature selection and train the Support Vector Machine (SVM) [4] classifier to identify the neuron structures. Ensemble Neuron Tracer [34] combines the data perturbation and model selection. Rivulet [38, 15] and Rivulet2 [16] back trace the longest uncovered neuron branches in each iteration with a confidence score. Probability Hypothesis Density Filtering Tracer [27] identifies the branch centrelines of the neuron structure using Bayesian probabilistic model and the accuracy of this method is further improved by introducing the sequential Monte Carlo estimation [28]. Triple-crossing 2.5D Convolutional Neural Network has been recently proposed to reduce the large computation cost of the general patch-based 3D learning to a practical level and train a model to detect the neuron arbours [17].

However, due to varying qualities of the neuron images captured by different microscopes, the fully automation of

*Corresponding author

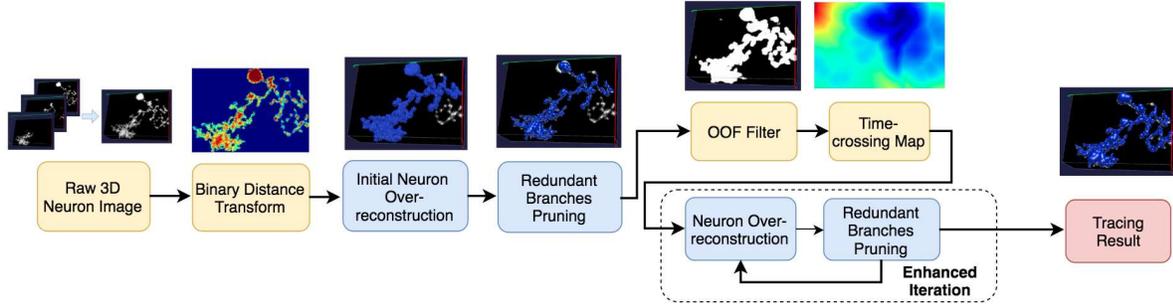


Figure 1. The pipeline of the proposed exhaustive tracing framework.

the neuron tracing algorithm remains a big challenge. The major obstacles during neuron tracing are broken structures and noises in the low quality neuron images. Most existing algorithms do not have a robust capability to bridge the gaps in the broken structures.

To solve the problems stated above, we have proposed a 3D neuron tracing framework called exhaustive tracing. Our framework takes in a raw 3D large-scale neuron image and generates the corresponding 3D neuron morphology. The tracing neuron is initially reconstructed by over-reconstruction and redundant branches pruning. This initial tracing result is expected to have miss-traced structures when the quality of input neuron image is low. Then we propose an enhanced iteration approach to further enhance the initial tracing result. For the exhaustive tracing, the time-crossing map of original neuron image is computed and the image is then enhanced by Optimally Oriented Flux (OOF) [13] filter to bridge the broken structures which are miss-traced in the initial reconstruction. During each iteration, an uncovered foreground voxel with the largest time-crossing value is picked as the new seed point to apply the over-reconstruction and redundant branches pruning, and the iteration will stop immediately if the over-reconstruction reaches the voxel covered in the previous tracing. The neuron images in First-2000 and Gold datasets from BigNeuron project [24, 26] are used to evaluate the performance of our method and compare with other state-of-the-art algorithms. The results show that our method outperforms the state-of-the-art algorithms with a more robust capability to stride over the gaps in low quality neuron images and a better balancing on the completeness of reconstruction.

2. Methods

In our method, Binary Distance Transform (BDT) [3] is first applied on the original neuron image and the node with the maximum distance transformed value is selected as a seed point. Based on this seed point, an exploration method using the idea of fast-marching to reach all possible foreground nodes is performed to generate an over-reconstruction tree graph of the traced neuron. Next, since

only the skeleton of the neuron needs to be kept, the hierarchical branches are constructed and redundant spurs are recursively pruned. Our method provides a solution to trace the discontinuous patterns in the neuron images based on exhaustive tracing framework with images enhanced by OOF filter after the initial reconstruction. The pipeline of the proposed method is summarized in Fig. 1 with the visualization of intermediate steps.

2.1. Seed Detection

BDT is first applied to highlight the centre of the neuron for generating a better quality of initial over-reconstruction. BDT is capable of enlarging the intensity value for the voxels close to the main skeleton of the neuron and decreasing the intensity value for the voxels close to the boundary. The Euclidean distances between each foreground voxel and the closest background voxel are used to compute the distance-transformed map, as shown in the following equation:

$$BDT(x_i) = \min(\sqrt{(x_i - b)^2}), \quad (1)$$

where x_i and b are the spatial coordinates of foreground voxels and the background voxels respectively after applying a rough threshold based on intensity distribution of the 3D neuron image.

Another important purpose to apply BDT is that distance-transformed map of the original image stack can provide the location information about soma due to the highest-intensity characteristic of the somas in the neuron images. Hence the geometric centre of the soma can be simply found by selecting the voxel with the maximum distance-transformed value. Since major dendrites and axons are connected at soma [33], the location of soma is normally chosen as the seed point (or start point) for the neuron reconstruction algorithm.

2.2. Exhaustive Tracing

2.2.1 Neuron Over-reconstruction

With the seed point selected from BDT, an over-reconstruction tree of the tracing neuron is generated

Algorithm 1 Neuron Over-reconstruction

- 1: *Input*: Seed location (x_s, y_s, z_s) , 3D neuron image stack I , intensity threshold t_i
 - 2: *Output*: Tree graph G
 - 3: Define *candidates* as the nodes of candidates stored during fast-marching
 - 4: **Procedure**
 - 5: Initialize the distance d for all voxels in I to be infinity
 - 6: Store seed into G and *candidates*
 - 7: Initialize *par* for every voxels I_x in I to it to be itself, so $par(I_x) = I_x$
 - 8: **while** *candidates* is not empty **do**
 - 9: Extract C_{min} which has the minimum $d(c)$ in the *candidates*
 - 10: Store C_{min} into G
 - 11: **for** each neighbor N_c within offset of two units of voxel length from C_{min} **do**
 - 12: **if** $I(N_c) \geq t_i$ **then**
 - 13: $d(N_c) \leftarrow \min(d(N_c), d(C_{min})$
 - 14: $+e(N_c, C_{min}))$
 - 15: **if** $d(C_{min}) + e(N_c, C_{min}) < d(N_c)$ **then**
 - 16: $par(N_c) \leftarrow C_{min}$
 - 17: where $e(x, y) = |x - y| \cdot \frac{(g_1(x) + g_1(y))}{2}$,
 - 18: and $g(x) = \exp(\lambda_1(1 - \frac{I(x)}{I_{max}})^2)$
 - 19: $I(x)$ indicates the intensity value of voxel x ,
 - 20: I_{max} is the maximum intensity value in I
 - 21: **if** $I(N_c) \geq t_i$ **then**
 - 22: Store N_c into *candidates*
 - 23: $par(N_c) \leftarrow C_{min}$
 - 24: **Return** G
-

by traversing from the seed location based on fast-marching [32]. The neuron image is treated as a graph G where two nodes next to each other in G are defined as neighbors. Fast-marching is an exploring algorithm which examines the neighbours of current processed node based on geodesic distance [36], then adds the valid neighbours into the candidates and the current processed node into resultant tree graph G . The details are described in Algorithm 1. The algorithm stops when none of the neighbours of the candidates is valid.

2.2.2 Redundant Branches Pruning

As we are only interested in the main skeleton of the neuron, the redundant branches in the over-reconstructed tree graph G need to be pruned, the pruning process involves hierarchical branches construction and coverage pruning.

The hierarchical branches construction starts by tracing all the leaf nodes without any children and backtracking to the nearest branch node, with each branch node having a minimum degree of 2. The path between each leaf node and

the corresponding branch node is defined as a hierarchical branch. The branch nodes are merged to the joint branch which has the longest path distance, and other hierarchical branches connected to this branch nodes are redefined as child branches of the joint branch. This merging process is iterated until the seed point is reached.

The importance factor of the hierarchical branches is determined by their path lengths and longer branch has higher importance factor, all the constructed hierarchical branches are sorted in the decreasing order according to their path lengths. For the coverage pruning, the branch has the longest path length in the sorted branches collection is selected in each iteration, the pruning decision depends on the intensity coverage ratio of the current processed branch within its coverage area. If the coverage ratio is less than a pre-defined coverage threshold, we keep this branch in the result, mask all the nodes within the coverage area of this branch and remove it from the sorted collection of the branches; otherwise this branch and all its related child branches are pruned. The pruning stops when all the branches are removed from the sorted collection.

We define the coverage area of a branch as the union of the coverage area of all the nodes in this branch. And the coverage area of a node is defined as the volume of a sphere which the centre point is the node with an estimated radius. The estimated radius r is calculated by an incremental method based on unit voxel length where r is initialized as 1 unit voxel length and increased by 1 unit voxel length in each iteration. If the ratio of background voxels within the volume of the sphere centre at the node with radius r respect to the foreground voxels is larger than 0.1%, then r is the estimated radius for the node, otherwise r is increased by 1 unit voxel length. The intensity coverage ratio is the sum of the intensity values of all nodes in the current process branch with respect to that of all nodes in the coverage area.

2.2.3 Optimally Oriented Flux

Before applying exhaustive iteration, an enhanced image needs to be obtained from the low quality image. This step is to link the broken structures in the low quality test images and can be omitted when the quality of the neuron image is high. The enhanced image is generated by Optimally Oriented Flux (OOF) which detects the 3D curvilinear like structure and connects the broken structures belonging to the same branch [13]. To be more specific, OOF finds the direction of optimal projection which minimizes the inward oriented flux [31]. Computation of OOF can be achieved analytically by filtering an image with a set of linear filters $\psi_{r,i,j}$,

$$q_{r,x}^{i,j} = \psi_{r,i,j}(x) * I(x) \quad (2)$$

By employing a step function $b_r(x)$ and the divergence theorem, the volume integral is extended to the entire image domain Ω :

$$\begin{aligned} q_{r,x}^{i,j} &= \int_{\Omega} b_r(y)((g_{\hat{a}_i \hat{a}_j} * I)(x+y))dV \\ &= ((b_r * g_{\hat{a}_i \hat{a}_j})(x)) * I(x) \end{aligned} \quad (3)$$

where $b_r(x) = \begin{cases} 1, & \|x\| \leq r \\ 0, & \text{otherwise} \end{cases}$ is the step function and $g_{\hat{a}_i \hat{a}_j}$ is the second derivative of Gaussian kernel along the orthogonal direction \hat{a}_i and \hat{a}_j . The last step is to apply Fourier transforms and Hankel transform [14] on b_r to get:

$$\begin{aligned} \beta &= \frac{4\pi r u_i u_j e^{-2(\pi \|u\|)^2}}{\|u\|^2} \\ \psi_{r,i,j}(u) &= \beta \left(\cos(2\pi r \|u\|) - \frac{\sin(2\pi r \|u\|)}{2\pi r \|u\|} \right). \end{aligned} \quad (4)$$

The eigenvalues and eigenvectors extracted from the computation of the OOF are on the local sphere surface which are grounded on the analysis of image gradient. The values of oriented flux along the eigenvectors are equal to the eigenvalues extracted from OOF:

$$\begin{aligned} \lambda_i(x; r) &= [\omega_i(x; r)]^T Q_{r,x} \omega_i(x; r) \\ &= f(x; r, \omega_i(x; r)) \end{aligned} \quad (5)$$

where $\omega_i(x; r)$ is the optimal direction matrix and $Q_{r,x}$ is the matrix form of $q_{r,x}^{i,j}$.

For the multiscale detection of OOF, a set of radii is required as input parameters, radii are respect to the r in Eq. (5) and the range of the set should cover between the smallest estimated radius and largest estimate radius in the neuron image volume. The broken structures in the original neuron image are connected after applying the OOF filter.

2.2.4 Enhanced Iteration

If the neuron over-reconstruction and redundant branches pruning are performed only once, the corresponding tracing result terminates at the broken neuronal fibres due to discontinuities in the neuron image and neighbouring exploration design of fast marching algorithm. Discontinuities in the neuron image are enhanced by introducing OOF filter described in Section 2.2.3. And the neighbouring exploration design of the fast marching is improved by selecting the new seed location and performing neuron over-reconstruction in Section 2.2.1 and pruning in Section 2.2.2 iteratively.

For the enhanced iteration, each time we select the voxel which has the maximum time cost as a new seed location, then apply the over-reconstruction of the neuron and hierarchical pruning. Furthermore, a threshold for the number of nodes of each branch is set as 50 to eliminate possible

noisy structures. The remaining foreground voxels in the candidates included in this iteration of tracing are deleted.

In order to select the new seed location, unlike the BDT only considers the local information, the time-crossing map of the original neuron image is calculated to take the global information into account. The time-crossing map is obtained by calculating the geodesic distance of neuronal foreground to the somatic centre, defined as:

$$BDT(x) = \frac{dx}{dT}, |\nabla T(x)| = \frac{1}{BDT(x)}, T(x_{soma}) = 0 \quad (6)$$

where BDT is binary distance transforms and x_{soma} is the point coordinate with largest $BDT(x)$ defined in Section 2.1. The time-crossing map is implemented using the multi-stencils fast marching (MSFM) [11] by applying BDT as its speed image starting from the somatic centre. The gap between a broken structure on $(T(x))$ is normally larger than the value on the centreline. Then the foreground voxels not included in the initial neuron reconstruction are found as candidates and they are sorted in decreasing order in terms of time cost. The overall exhaustive tracing framework is

Algorithm 2 Exhaustive Tracing

- 1: *Input*: 3D neuron image stack I , Intensity threshold t_i , Iteration number n
 - 2: *Output*: Tree graph G
 - 3: **Procedure**
 - 4: $seed \leftarrow BDT_{max}(I)$
 - 5: $G \leftarrow NO(seed, I, t_i)$
 - 6: $G \leftarrow RBP(G)$
 - 7: $E(I) \leftarrow OOF(I)$
 - 8: $T(I) \leftarrow MSFM(I)$
 - 9: $B(I) \leftarrow I(x > t_i)$
 - 10: $candidates \leftarrow \{x : x \in B(I) \text{ and } x \notin G\}$
 - 11: Sort $candidates$ according to the $T(I)$ in descending order
 - 12: $n_i \leftarrow 0$
 - 13: **while** $candidates$ is not empty and $n_i \leq n$ **do**
 - 14: Extract the first voxel c_i from $candidates$
 - 15: $G_i \leftarrow NO(c_i, E(I), t_i)$
 - 16: (where the over-reconstruction in exhaustive iteration will stop immediately if it reaches any voxel in G)
 - 17: $G_i \leftarrow RBP(G_i)$
 - 18: Add G_i into G
 - 19: $n_i \leftarrow n_i + 1$
 - 20: **Return** G
-

stated in Algorithm 2, where NO indicates Neuron Over-reconstruction method stated in Section 2.2.1 and RBP indicates Redundant Branches Pruning method stated in Section 2.2.2.

3. Experiments and Results

3.1. Dataset and Setup

The dataset used for evaluating our method is from the BigNeuron project [24]. The first part of BigNeuron is called First 2000 which contains 2000 preprocessed large-scale neuron images of fly, for which however there is no ground truth annotation available. The second part is the Gold dataset and each neuron image in this dataset has a corresponding gold standard reconstruction verified by three computational neuroscientists. These neuron images come from large-scale neuroninformatics projects and directly contributed from neuroscientists worldwide. The dataset has been captured under varying types of microscopes and neuron labeling techniques for different animal species. Each neuron image in this dataset has different volume sizes. The results of reconstructing 3D neuron images are visualized by Vaa3D [23].

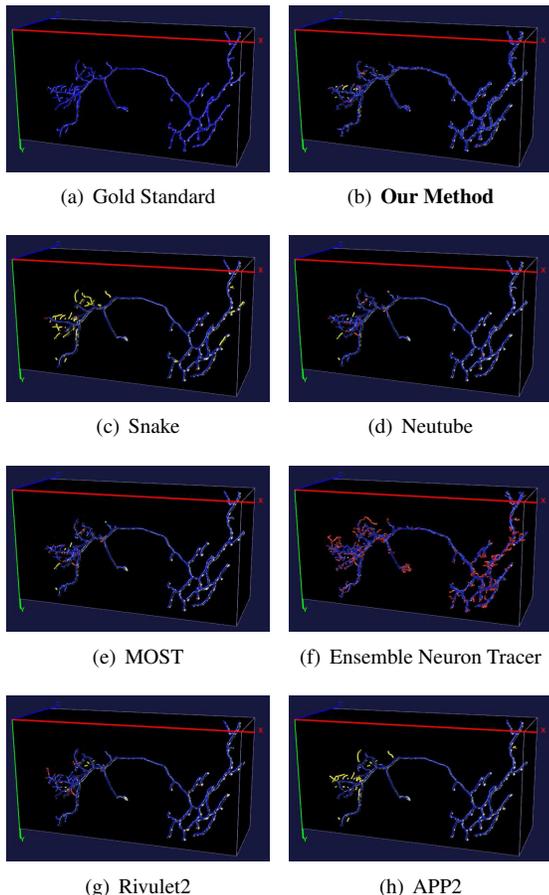


Figure 2. Comparison of the neuron reconstruction with other state-of-art algorithms on one neuron image from Janelia-Fly dataset in BigNeuron: Snake, Neutube, MOST, APP2, Ensemble Neuron Tracer and Rivulet2. The nodes marked in red indicate the over-traced structures and the nodes marked in yellow indicate the miss-traced structure.

3.2. Analysis of Results on Gold Dataset

	Precision	Recall	F-score
Our method	92.2936%	95.9544%	93.0690%
Snake	92.7854%	66.3830%	76.1651%
Neutube	92.4260%	49.0801%	62.2028%
MOST	91.6606%	83.7389%	86.5277%
ENT	73.1409%	91.1554%	74.7675%
Rivulet2	91.3611%	92.8872%	91.4173%
APP2	69.9866%	95.3009%	79.5407%

Table 1. Performance of different methods on reconstructing 42 neuron images in Janelia-Fly. ENT represents for Ensemble Neuron Tracer.

A quantitative evaluation based on the precision, recall and F-score value is defined as:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (8)$$

where TP is defined as the number of matched nodes between result and gold standard, the value of FP is simply $n - TP$, and n is the total number of the nodes in the result. FN is defined as the nodes in gold standard but not in reconstruction result and this value indicates the coverage ratio of the result in gold standard.

In particular, all 42 neuron images of the larval *Drosophila* are selected from the Janelia-Fly dataset to be reconstructed in our evaluation. Most of these images have visual distinguishable gaps or broken structures. The volume sizes of these 3D neuron images range from $89 \times 93 \times 111$ to $217 \times 490 \times 257$ and the average number of the voxels to be examined for each image stack is 6868496.5. Each voxel is $0.38 \mu m$ along Z axis and since the images are captured under isotropic sampling, the voxel size along X axis and Y axis is the same. We have tuned the corresponding intensity thresholds of each test image provided by BigNeuron project for all state-of-the-art algorithms and our method to achieve their best performances during the experiments. The number of iteration is fixed to 50 for the enhanced iteration in our exhaustive tracing framework. Table 1 shows the average precision, average recall and corresponding F-score of reconstruction results of neuron images in Janelia-Fly by using the state-of-the-art neuron tracing algorithms and our method. Among all the methods, APP2 is the most fastest algorithm to complete the tracing. From the table, our method achieves the best balance between tracing broken structures in the low-quality images and avoiding the interference of noise since it has the highest F-score values. However, the precision is penalized due to the noise in these low quality images.

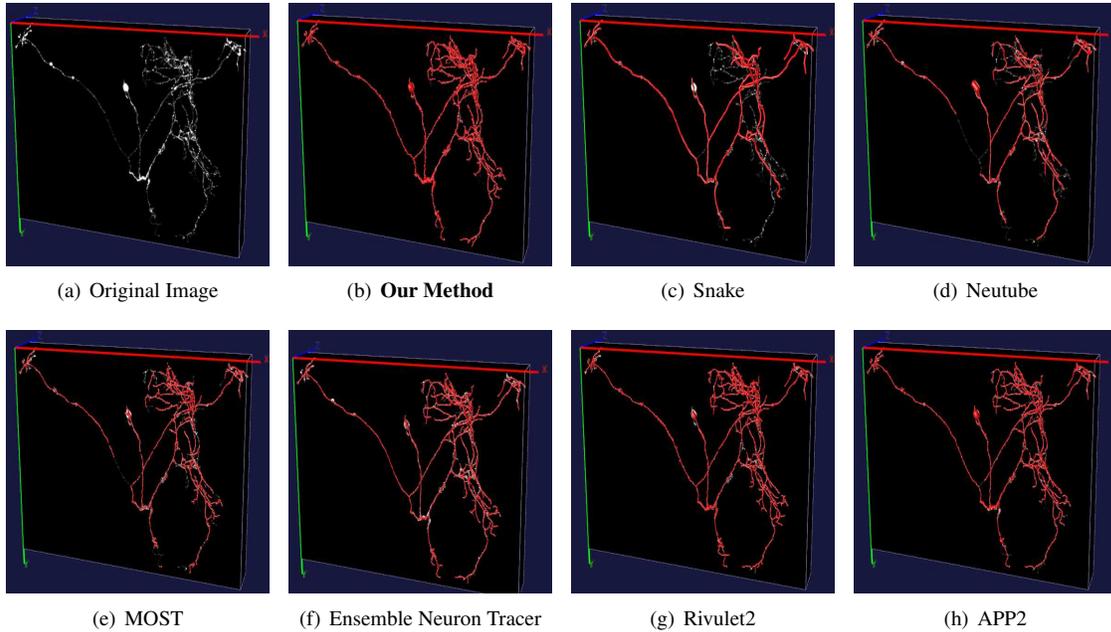


Figure 3. Reconstruction results of a highly discontinuous fly neuron image from First-2000 dataset. All reconstruction results shown in the above image are generated by the neuron reconstruction algorithm without any manual editing.

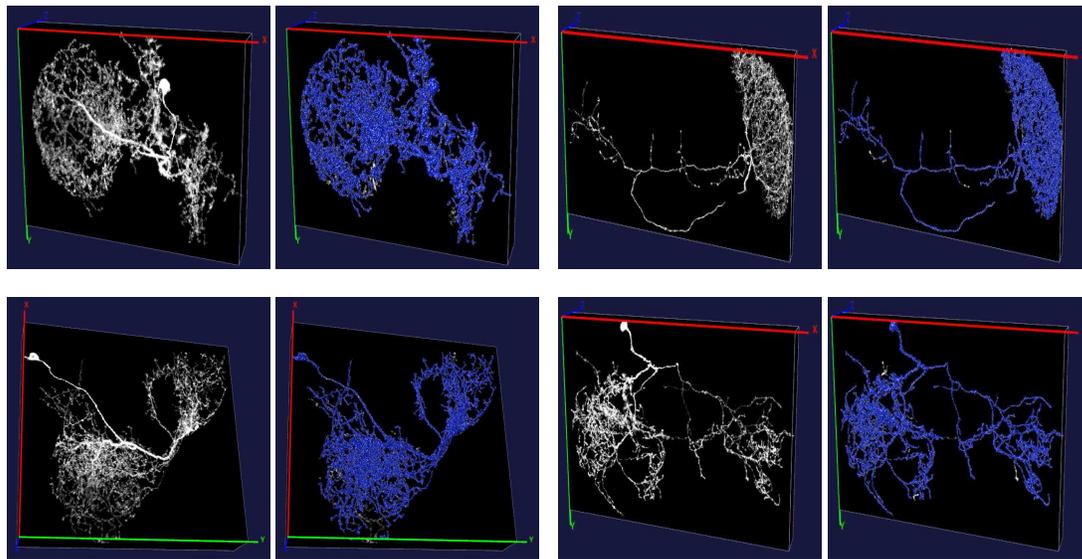


Figure 4. Reconstruction results using exhaustive tracing on extremely complicated neurons regarding the close-by dendrites and large number of nodes. Our method achieves robust results on these complicated neurons.

Fig. 2 shows the reconstruction result using our method compared to the state-of-the-art algorithms on one *Janelia-Fly* neuron image. The nodes marked in red indicate the over-traced structures and the nodes marked in yellow indicate the miss-traced structure. It can be seen that our method achieved a robust reconstruction result on this image. The result generated by Ensemble Neuron Tracer over-traced a large amount of neuron structures and Snake and

APP2 miss-traced neuron branches close to the leaf nodes.

3.3. Analysis Results on First-2000 Dataset

To further evaluate the robustness of our method, we tested it on the large-scale First-2000 dataset. The volume size of the neuron image in this dataset is from $78 \times 106 \times 19$ up to $770 \times 1024 \times 100$ and the average number of the voxels to be processed for each image is 9689032.8. Since

the neuron images have already been preprocessed in this dataset, the intensity threshold t_i for the over-reconstruction is fixed to 5 in the exhaustive tracing framework. Our visual inspection shows that our method is able to generate reasonable results of 2000 fly neuron images. Fig. 3 shows the reconstruction results of a fly neuron image with broken structures by Snake [35], NeuTube [8], MOST [21], Rivulet2 [16], Ensemble Neuron Tracer [34], APP2 [36] and our method. Fig. 3 demonstrates the reconstruction results generated from these methods. Snake, Neutube and MOST failed to derive a generally complete reconstruction since there exist large amount of miss-traced structures by visual inspection. In addition, the visualization of the results shows that Snake, MOST, Ensemble Tracer and Rivulet2 under-estimated the radius of the somatic structure. Note that there are no computational labelled ground truth available for this dataset, so no quantitative comparison is conducted on this dataset. Some reconstruction results of our method of neuron images with large number of nodes and complicated morphology are shown in Fig. 4.

3.4. Investigation of Enhanced Iteration Setting

For the extremely low quality neuron image as demonstrated in Fig. 6 (a), our method is able to cover majority of the foreground voxels. With the increasing number of iterations, more percentage of foreground voxels are covered, however, the ratio of errors is also increasing. In this particular neuron image captured from frog, there exists a large region of noises on the top left, which affects the result of exhaustive tracing. Fig. 5 illustrates the ratio of neuron signal coverage and errors caused by noises of this testing image with highly discontinuous neuronal fibres. The neuron reconstruction results of our method at different iterations are visualized in Fig. 6.

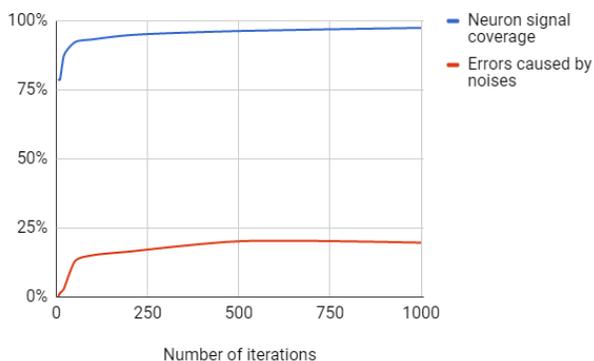


Figure 5. Neuron signal coverage and the errors caused by noises. The blue line and the red line indicate the neuron signal coverage ratio and the ratios of errors caused by noises respectively.

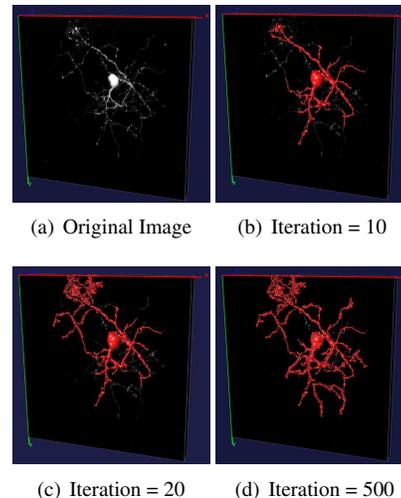


Figure 6. (a) Frog neuron image with highly discontinuous structures. Reconstruction results at different iterations are (b) initial reconstruction with a few neuronal segments traced, (c) intermediate reconstruction with more neuronal segments traced, and (d) final reconstruction with the majority of neuronal segments traced.

4. Conclusion

The neuron images from different datasets can have completely different qualities mainly because of the acquisition of the images is under varying conditions. The phenomenon of different broken and noisy patterns existing in the neuron images is caused by different lab conditions and neuron labeling techniques. OOF filter used for exhaustive tracing is capable of connecting broken structures together, but in these neuron images which have high-intensity noises existed, it enlarges the influence of these noises. To improve the robustness of OOF to noises, both global and local information are considered after obtaining the time-crossing map. In this study, an exhaustive neuron tracing framework is proposed to overcome the discontinuous patterns. This method has been tested on the large-scale First-2000 dataset and Gold dataset. The results show that our method achieved the best balance between tracing broken structures and avoiding inferences of noises compared to state-of-the-art neuron tracing algorithms for low quality images.

References

- [1] G. A. Ascoli. Neuroinformatics grand challenges. *Neuroinformatics*, 6(1):1–3, 2008.
- [2] M. F. Bear, B. W. Connors, and M. A. Paradiso. *Neuroscience: Exploring the brain*. Lippincott Williams & Wilkins, 2007.
- [3] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, 1995.

- [4] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [5] H. Chen, H. Xiao, T. Liu, and H. Peng. Smarttracing: self-learning-based neuron reconstruction. *Brain Informatics*, 2(3):135–144, 2015.
- [6] L. d. F. Costa, E. T. M. Manoel, F. Faucereau, J. Chelly, J. van Pelt, and G. Ramakers. A shape analysis framework for neuromorphometry. *Network: Computation in Neural Systems*, 13(3):283–310, 2002.
- [7] D. E. Donohue and G. A. Ascoli. Automated reconstruction of neuronal morphology: an overview. *Brain Research Reviews*, 67(1):94–102, 2011.
- [8] L. Feng, T. Zhao, and J. Kim. neutube 1.0: a new design for efficient neuron reconstruction software based on the swc format. *Eneuro*, 2(1):0049, 2015.
- [9] S. Gulyanov, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis. CRF formulation of active contour population for efficient three-dimensional neurite tracing. In *ISBI*, 2016.
- [10] M. Halavi, K. A. Hamilton, R. Parekh, and G. Ascoli. Digital reconstructions of neuronal morphology: three decades of research trends. *Frontiers in neuroscience*, 6:49, 2012.
- [11] M. S. Hassouna and A. A. Farag. Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9), 2007.
- [12] J. L. Lanciego and F. G. Wouterlood. A half century of experimental neuroanatomical tracing. *Journal of chemical neuroanatomy*, 42(3):157–183, 2011.
- [13] M. Law and A. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *ECCV*, 2008.
- [14] J. W. Layman. The hankel transform and some of its properties. *J. Integer Seq*, 4(1):1–11, 2001.
- [15] S. Liu, D. Zhang, S. Liu, D. Feng, H. Peng, and W. Cai. Rivulet: 3D neuron morphology tracing with iterative back-tracking. *Neuroinformatics*, 14(4):1–15, 2016.
- [16] S. Liu, D. Zhang, Y. Song, H. Peng, and W. Cai. Automated 3D neuron tracing with precise branch erasing and confidence controlled back-tracking. *bioRxiv doi: 10.1093*, 2017.
- [17] S. Liu, D. Zhang, Y. Song, H. Peng, and W. Cai. Triple-crossing 2.5D convolutional neural network for detecting neuronal arbours in 3D microscopic images. In *MICCAI Workshop on Machine Learning in Medical Imaging*, 2017.
- [18] J. Livet, T. A. Weissman, H. Kang, R. W. Draft, J. Lu, R. A. Bennis, J. R. Sanes, and J. W. Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450(7166):56–62, 2007.
- [19] E. Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77(7):693–704, 2010.
- [20] K. D. Micheva and S. J. Smith. Array tomography: a new tool for imaging the molecular architecture and ultrastructure of neural circuits. *Neuron*, 55(1):25–36, 2007.
- [21] X. Ming, A. Li, J. Wu, C. Yan, W. Ding, H. Gong, S. Zeng, and Q. Liu. Rapid reconstruction of 3D neuronal morphology from light microscopy images with augmented rayburst sampling. *PloS One*, 8(12):e84557, 2013.
- [22] R. Parekh and G. A. Ascoli. Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron*, 77(6):1017–1038, 2013.
- [23] H. Peng, A. Bria, Z. Zhou, G. Iannello, and F. Long. Extensible visualization and analysis for multidimensional images using Vaa3D. *Nature Protocols*, 9(1):193–208, 2014.
- [24] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli. Bigneuron: large-scale 3D neuron reconstruction from optical microscopy images. *Neuron*, 87(2):252–256, 2015.
- [25] H. Peng, F. Long, and G. Myers. Automatic 3D neuron tracing using all-path pruning. *Bioinformatics*, 27(13):i239–i247, 2011.
- [26] H. Peng, E. Meijering, and G. A. Ascoli. From diadem to bigneuron. *Neuroinformatics*, 13(3):259–260, 2015.
- [27] M. Radojević and E. Meijering. Automated neuron tracing using probability hypothesis density filtering. *Bioinformatics*, 33(7):1073–1080, 2017.
- [28] M. Radojević and E. Meijering. Neuron reconstruction from fluorescence microscopy images using sequential monte carlo estimation. In *ISBI*, 2017.
- [29] A. Rodríguez, D. B. Ehlenberger, P. R. Hof, and S. L. Wearne. Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images. *Nature Protocols*, 1(4):2152–2161, 2006.
- [30] B. Roysam, W. Shain, and G. A. Ascoli. The central role of neuroinformatics in the national academy of engineering's grandest challenge: reverse engineer the brain. 7:1–5, 2009.
- [31] H. M. Schey. *Div, grad, curl, and all that*. Norton, 1973.
- [32] J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999.
- [33] L. Squire, D. Berg, F. E. Bloom, S. Du Lac, A. Ghosh, and N. C. Spitzer. *Fundamental Neuroscience*. Academic Press, 2012.
- [34] C.-W. Wang, Y.-C. Lee, H. Pradana, Z. Zhou, and H. Peng. Ensemble neuron tracer for 3D neuron reconstruction. *Neuroinformatics*, 15:185–198, 2017.
- [35] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam. A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, 9(2-3):193–217, 2011.
- [36] H. Xiao and H. Peng. App2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11):1448–1454, 2013.
- [37] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.
- [38] D. Zhang, S. Liu, S. Liu, D. Feng, H. Peng, and W. Cai. Reconstruction of 3D neuron morphology using rivulet back-tracking. In *ISBI*, 2016.