Neuroinformatics, 2015 (in press)

TReMAP: Automatic 3D Neuron Reconstruction Based on Tracing, Reverse Mapping and Assembling of 2D Projections

Zhi Zhou, Xiaoxiao Liu, Brian Long, and Hanchuan Peng*

Allen Institute for Brain Science, Seattle, WA, USA.

*Corresponding Author. Email: hanchuanp@alleninstitute.org

Abstract:

Efficient and accurate digital reconstruction of neurons from large-scale 3D microscopic images remains a challenge in neuroscience. We propose a new automatic 3D neuron reconstruction algorithm, TReMAP, which utilizes 3D Virtual Finger (a reverse-mapping technique) to detect 3D neuron structures based on tracing results on 2D projection planes. Our fully automatic tracing strategy achieves close performance with the state-of-the-art neuron tracing algorithms, with the crucial advantage of efficient computation (much less memory consumption and parallel computation) for large-scale images.

Availability: This method has been implemented as an Open Source plugin for Vaa3D (http://vaa3d.org).

1. Introduction

Digital neuron reconstruction (tracing) is an important way to quantify neuronal morphology (Kawaguchi, et al., 2006; Krahe, et al., 2011; DeFelipe, et al., 2013; Peng, et al., 2013; Peng, et al., 2015; Peng, et al., 2015). For better understanding the detailed morphology of neurons, 3D images of neurons are often acquired in high resolution, resulting in large volume datasets, which have posed substantial challenges in efficient and accurate reconstruction of complicated neuron morphology (Zhou, et al., 2015).

Researchers have developed several methods to reconstruct 3D neuron images from 2D image sequences (Fiala, 2005; Lu, et al., 2009; Myatt, et al., 2012). These methods are mostly semi-automatic. The basic idea was to trace neuron structures on each individual 2D plane, followed by assembling 2D fragments into a complete 3D reconstruction. Such processes often require identification of morphology landmarks (e.g. soma, branch terminal points, etc.) or human interactions.

Many 3D reconstruction algorithms have also been proposed for automated neuron tracing, including ray casting (Wearne, et al., 2005; Ming, et al., 2013), tube-fitting model (Feng, et al., 2015), pruning of over-reconstructions (Peng, et al., 2011; Xiao and Peng, 2013), deformable curves (Peng, et al., 2010; Wang, et al., 2011), and others. These methods work when the volume of image data is within the range of hundreds of megabytes to a few gigabytes. However, when the size of a neuron image is greater than 10 GB, most of the current methods would become inapplicable on typical single-PC platforms, due to both intolerable computation time and excessive memory requirements.

One common solution is to divide-and-conquer a large-scale image by only processing a small image tile each time and then assembling all the pieces of reconstruction results together. This approach is clearly not efficient, as it requires tracing all image tiles and assembling all respective reconstructions. A recently developed method, NeuronCrawler, was developed to reconstruct a neuron from a large-scale image by iterating a depth-first search strategy over a number of image-tiles and thus only trace in image tiles that contain neurites (Zhou, Sorensen and Peng, 2015). NeuronCrawler can efficiently trace a single neuron's morphology from a large-scale image. However, when the neuron has broken pieces in the image, NeuronCrawler will not be able to reconstruct the complete neuron structure.

Human annotators are able to produce high quality neuron reconstructions based on purely manual or semi-automatic methods. One recent efficient tool called Virtual Finger, which also bears the name 3D-WYSIWYG ("what you see in 2D is what you get in 3D") (Peng, et al., 2014), has been used in several large neuroscience initiatives worldwide (e.g. Allen Institute for Brain Science and the European Human Brain Project) to reconstruct neurons quickly in a semi-automatic way. The key idea is to annotate and detect 3D curves and reconstruct neurites effectively based on only 2D inputs using a regular computer mouse and screen. Inspired by this annotation technique, we wanted to address the challenge of automatic neuron tracing using a 2D-to-3D reverse mapping approach.

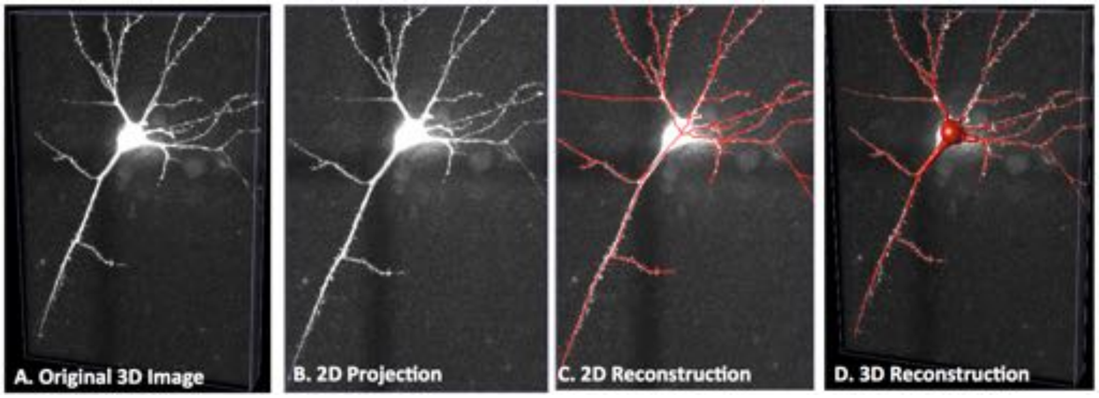


Figure 1. Schematic illustration of TReMAP neuron reconstruction method: A) A confocal microscopy image of the fluorescently-labeled mouse neuron at Allen Institute for Brain Science. The neuron was labeled via iontophoretic microinjection of 6% Lucifer Yellow in formalin-fixed coronal sections. After mounting in Fluoromount G, the Lucifer Yellow fluorescence was imaged on a confocal laser scanning microscope. The image voxel size is 0.18 x 0.18 x 0.5 um. The image stack size is 610 x 900 x 100 voxels. B) 2D maximum intensity projection on XY plane. C) 2D reconstruction which is overlaid on top of the B) (only the skeleton is shown). D) 3D reconstruction is overlaid on top of the original image A).

Here we present a new fully automated 3D reconstruction algorithm, called TReMAP, short for "Tracing, Reverse Mapping and Assembling of 2D Projections." Instead of tracing a 3D image directly in the 3D space as seen in majority of the tracing methods, we first trace the 2D projection trees in 2D planes, followed by reverse-mapping the resulting 2D tracing results back into the 3D space as 3D curves; then we use a minimal spanning tree (MST) method to assemble all the 3D curves to generate the final 3D reconstruction (Figure 1). Because we simplify a 3D reconstruction problem into 2D, the computational costs are reduced dramatically.

2. Method

The TreMAP method starts with tracing 2D projection trees and reversely mapping the 2D trees back to 3D space to produce 3D curves, which are then assembled together to form 3D neuron trees. The detailed steps are as follows.

2.1. 2D projections and connected components

We first generate the 2D maximum intensity projections (MIPs) from an input 3D image (Figure 2A). The projections can be produced based on either orthogonal planes or planes in arbitrary angles. Without losing generality, we use the typical XY, XZ, and YZ projection planes (Figure 2A) in this paper.

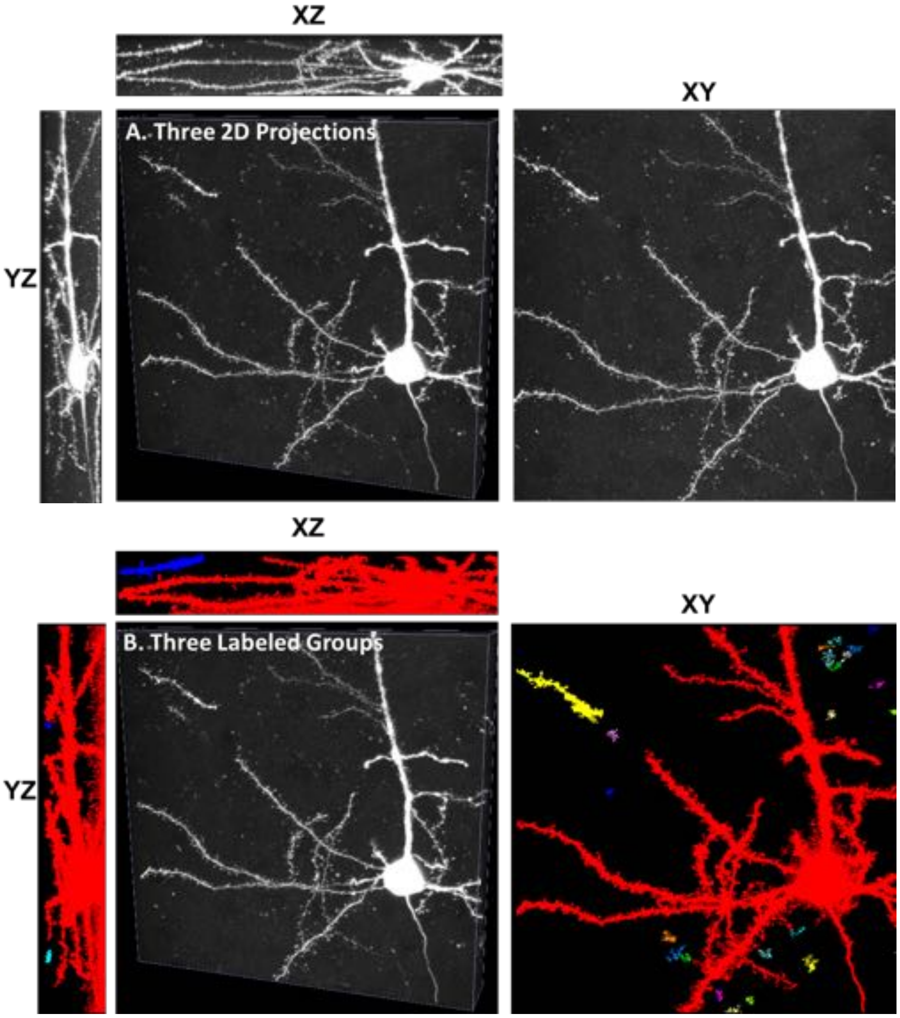


Figure 2. A) MIPs of a 3D confocal microscopy image on XY, XZ, and YZ planes. B) The corresponding connected groups. Different connected components are shown in different colors.

On each 2D MIP image, the foreground and background are separated based on intensity thresholding. We adopt the average intensity value of an entire 2D projection image as the default threshold to extract the foreground. Then in the foreground, all connected components are identified and labeled with different indexes (Figure 2B).

2.2. Tracing 2D Projection Trees

For each labeled group, we trace the signals in 2D projections (Figure 3) by using all-path-pruning tracing algorithms, which often produce a single neuron tree structure that has less false positives than most other tracing algorithms (Peng, Long and Myers, 2011; Xiao and Peng, 2013). Yet, we note that any general tracing algorithms that support neuron tracing in a 2D image may also be applied here (see Figure 3 for instances). In the cases of all-path-pruning algorithms, a neuron is reconstructed via first searching all possible signals from a single neuron image to generate an over-reconstructed tree. Then, several different pruning strategies are used to simplify the neuron tree (Peng, Long and Myers, 2011; Xiao and Peng, 2013). Since the tracing process of TreMAP is based on single 2D projections, the computation usually takes a few seconds (for a 1024 x 1024 2D image, it only takes 7 seconds for reconstruction at full resolution).

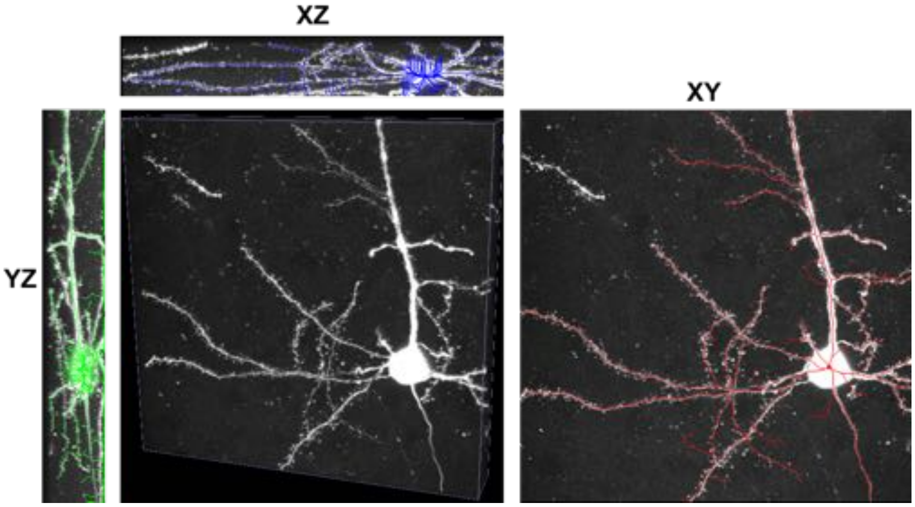


Figure 3. 2D tracing results on the largest connected group of three 2D projections. The skeletons of the reconstructions are overlaid on top of the image.

2.3. Reverse mapping and assembling

As mentioned above, the “3D Virtual Finger” technology (Peng, et al., 2014) allows annotators to detect and generate 3D objects by mapping users’ inputs in 2D displays back to 3D space. In the interactive neuron reconstruction, an annotator manually paints one or more 2D curves following the trajectories of neurites projected from a 3D image to a 2D screen. The Virtual Finger algorithm automatically reconstructs 3D neurites by combining both the 2D prior information input by human and the 3D image content.

To prepare for the reverse mapping, we first extract the 2D neuron tree by tracing a projection image (see Figure 4A) and then it is broken into a collection of 2D segments (see Figure 4B). For each segment, a 2D bounding box is identified and used to extract the corresponding 3D image content for reverse mapping. This divide-and-conquer strategy is crucial to cut down the memory consumption of reverse mapping and it enables parallel processing that can further cut down computation time.

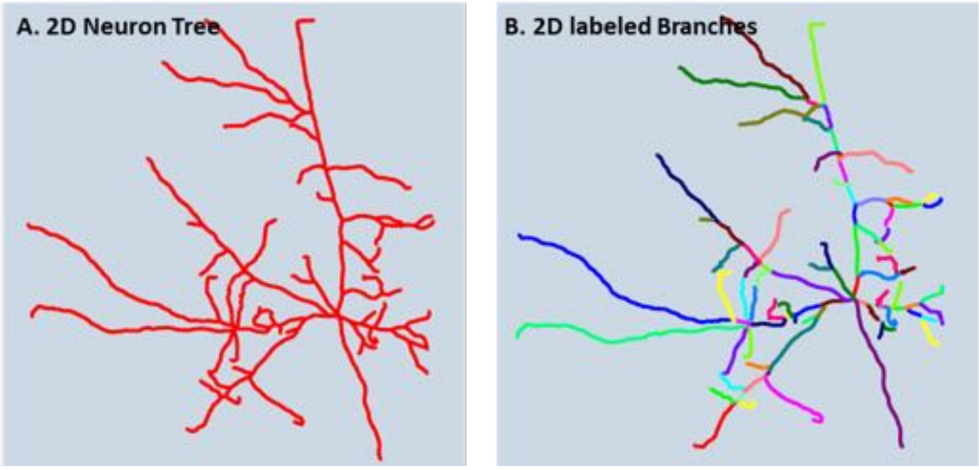


Figure 4. One example of a 2D neuron tree and its branches. In A), a single neuron tree structure has been generated by 2D tracing algorithm. In B), the neuron tree has been broken into multiple branches. Different branches are labeled in different colors.

To map the 2D projection tree onto 3D curves in the image space we use Curve-Drawing Algorithm 2 (CDA2) from the Virtual Finger method (Peng, Tang, Xiao, Bria, Zhou, Butler, Zhou, Gonzalez-Bellido, Oh and Chen, 2014). From each node of the 2D curve, a shooting ray is generated along orthogonal direction of the 2D projection plane. The starting 3D node is identified by the maximal intensity location along the first shooting ray (at the first 2D node). Then CDA2 searches for 3D curve path from the last 3D node to the current shooting ray using an adapted fast-marching algorithm. This process iterates over all the 2D nodes and forms the corresponding 3D curve. Figure 5 shows one example of the reverse mapping algorithm. Even though the two neurons are interlaced in the 2D projection, the reversely mapped 3D curves are precisely detected and correctly separated.

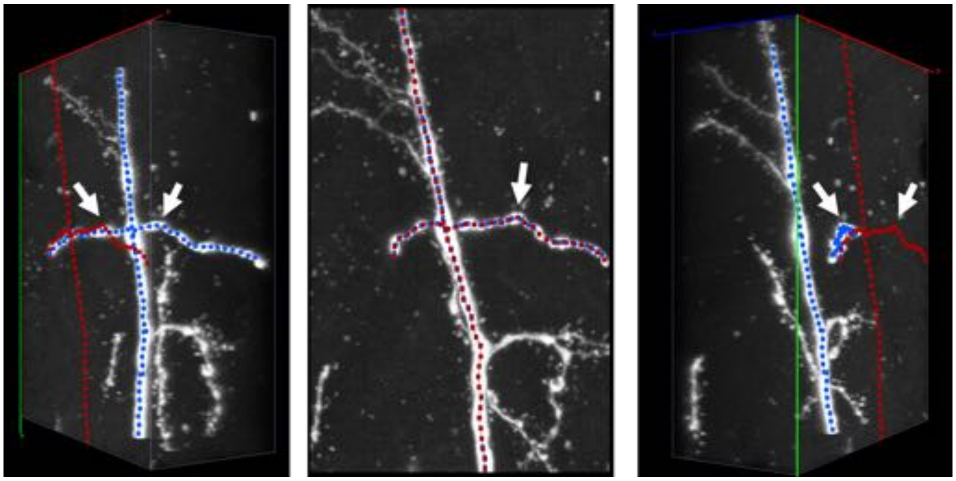


Figure 5. 3D curves detected using "3D Virtual Finger." 2D tracing results are marked as red, and the reversely mapped 3D tracing results are marked as blue. Three different angles are shown for better visualization.

A 3D reconstructed neuron is often represented using a tree graph, which is made up of a series of reconstruction nodes. Each node has its own 3D X, Y, Z locations, radius, as well as the information which other node is this node's topological "parent." In a simpler way, a neuron reconstruction can also be viewed as a connected graph of many "segments," each being a 3D curve consisting of a number of reconstruction nodes. In the case of TReMAP, each segment corresponds to a 3D curve produced using the Virtual Finger methods.

In the end, MST is used to connect all 3D curves to produce the final neuron reconstruction. Figure 6 shows the 3D tracing results using different 2D projection planes. TReMAP achieves better performance with the XY projection plane than XZ or YZ projection planes, because usually the neuron image has lower resolution in the Z direction, and a better signal to noise ratio (SNR) in the XY plane.

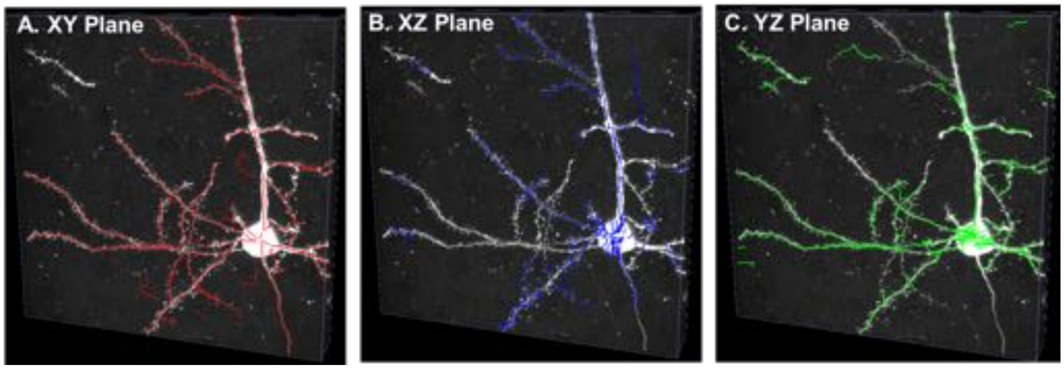


Figure 6. 3D reconstruction results for the largest connected group of each 2D projection. In A), the 3D reconstruction is generated from XY plane projection. In B), the 3D reconstruction is generated from XZ plane projection. In C), the 3D reconstruction is generated from YZ plane projection. More 3D neuron structures have been reversely mapped using 2D tracing result from XY plane than XZ or YZ plane.

2.4. Multi-projection approach

Our reverse mapping is based on the tracing results in 2D projections. If two or more 3D neuron structures overlap in one 2D projection, only one 3D structure will be reversely mapped using "3D Virtual Finger." To address this issue, multiple projections from different angles can be used to make sure that all neuron structures can be detected.

In this work, we introduced a multi-projection approach based on XY, XZ, and YZ planes for our TReMAP neuron reconstruction. First, we used the 3D reconstruction from XY plane as the basic tracing result, since the XY plane typically has better spatial resolution than XZ and YZ planes. Then all newly identified 3D segments traced from the XZ and YZ planes are added to the set of segments linked via MST to yield the final result. Figure 7 shows the original image, the 3D tracing result from multi-projections, and the 3D tracing result from XY projection only. From the zoom-in area we can see that two 3D neuron structures overlapped in XY plane can be traced precisely using our multi-projection approach.

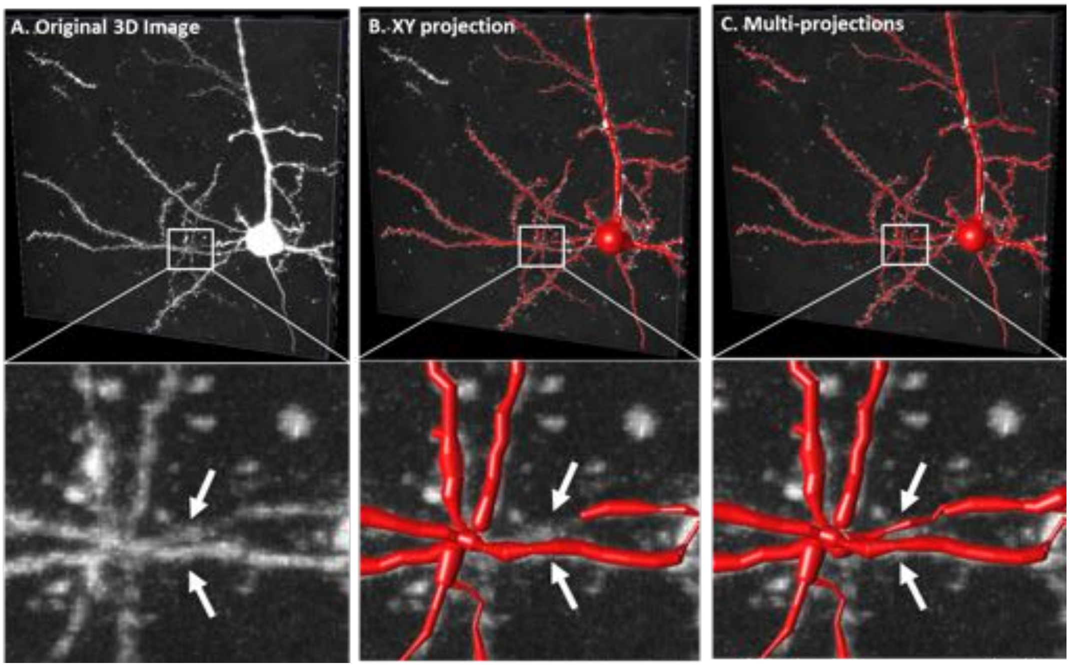


Figure 7. Comparison results of tracing multi-projections and XY projection. Two neuron structures overlapped in XY plane (pointed by two arrows in A) have been precisely reconstructed from multi-projections (pointed by two arrows in C). But only one structure is detected in XY projection (pointed by two arrows in B).

3. Experimental Results

To evaluate the reverse mapping performance of our TReMAP approach, we tested a confocal neuron image aligned from four sections along Z-axis (Figure 8). Unlike other single-stack 3D neuron images, the neuron structure of this aligned image (3D dimension is 2001 x 3183 x 975 voxels) is relatively complicated.

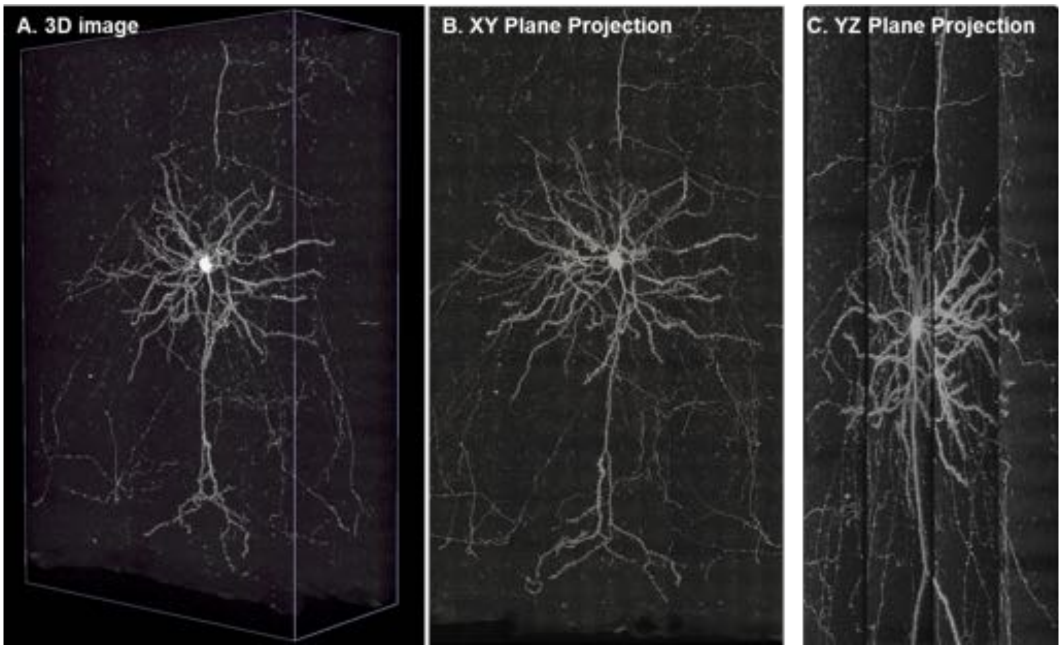


Figure 8. A confocal microscopy image of the fluorescently-labeled mouse neuron aligned from four sections. The neuron is electroporated with a GFP plasmid and imaged in the resonant confocal at the Allen Institute for Brain Science. The image voxel size is 0.143 um x 0.143 um x 0.280 um. The image stack size is 3099 x 3183 x 975. A) 3D view, B) XY projection, C) YZ projection.

We first divided this 3D image into *M* (*M* = 1, 2, 3, … , 10) equal 3D sub-images along the Z-axis. For each sub-image, we traced 2D structures on the XY projection plane. Then, we reverse-mapped the 2D tracing results onto the 3D sub-image to obtain 3D curves from each sub-image. The final 3D reconstruction is the fusion of the 3D curves from these *M* 3D sub-images. To quantify the variation in TReMAP results for different numbers of sub-images, we calculated the total length of the reconstructed neuron using *M* (*M* = 1, 2, 3, … , 10) 3D sub-images. The standard deviation is 265.92 µm, which is 3% of the average 8798.69 µm, showing that the reverse mapping algorithm is robust and is not sensitive to the number of sub-images.

3.1. Single neuron reconstruction comparison

To evaluate the accuracy of the TreMAP algorithm, we compared our results with four other state-of-the-art tracing algorithms on single neuron reconstructions: Micro-Optical Sectioning Tomography (MOST) ray-shooting tracing (Wu, et al., 2014), NeuTube tracing (Feng, Zhao and Kim, 2015), open-curve snake (Snake) tracing (Narayanaswamy, et al., 2011; Wang, Narayanaswamy, Tsai and Roysam, 2011) and the all-path pruning 1 (APP1) tracing (Peng, Long and Myers, 2011). For the completeness of the comparison, we also included the result using a 2D image sequence based method by directly assembling 2D tracing results using the MOST ray-shooting algorithm. These comparison methods cover major known categories of tracing algorithms, and thus form a suitable set of methods for comparison (Peng, Meijering and Ascoli, 2015).

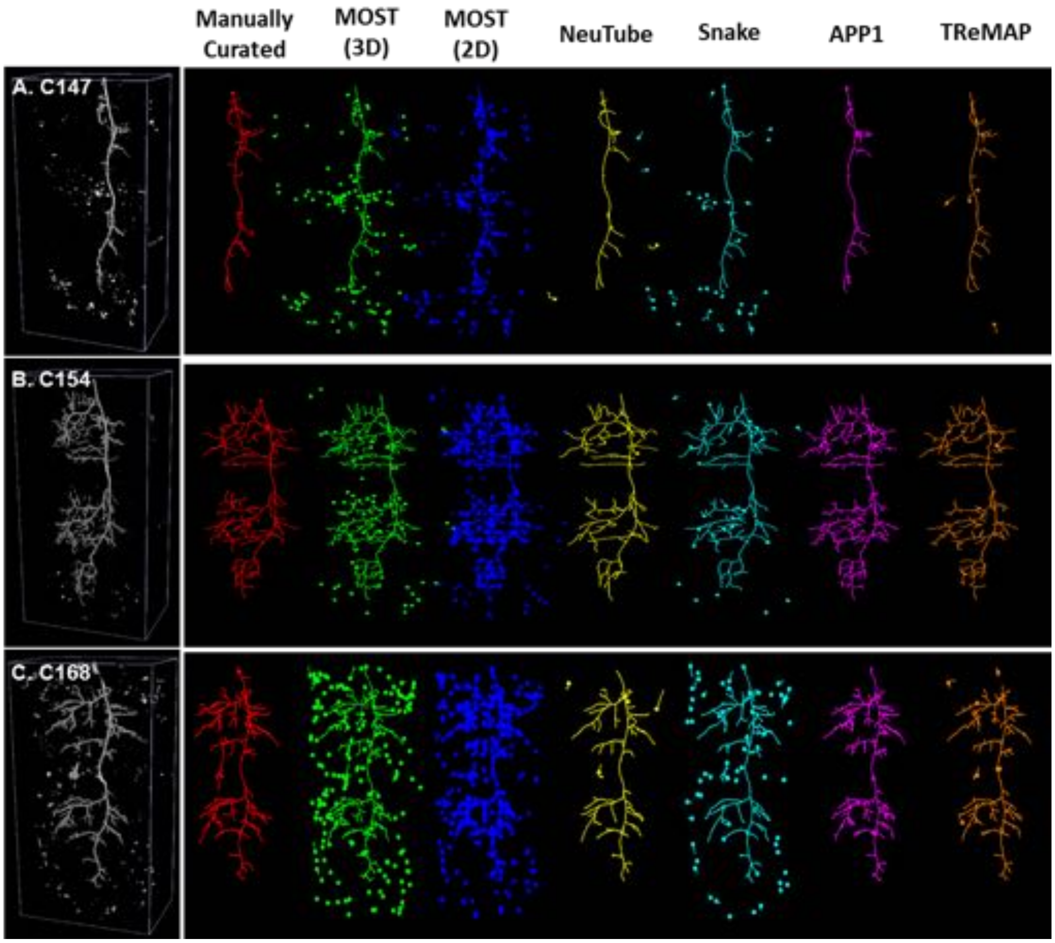
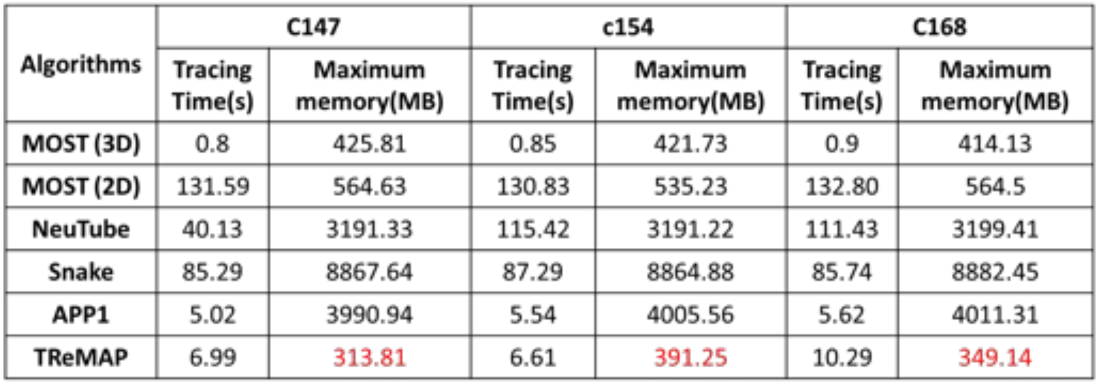


Figure 9. Reconstruction comparison results for three confocal dragonfly images. Reconstructions generated by APP2 (manually curated), MOST (both 2D and 3D), NeuTube, snake, APP1, and TReMAP are shown in different colors. Compared to the manually curated reconstructions, TReMAP's results have less false detection than those of MOST, NeuTube, and snake tracing, and have comparable performance to those of APP1.

Three confocal images of dragonfly neurons (C147, C154, and C168) from dragonfly neuron datasets (Gonzalez-Bellido, et al., 2013) were used as the testing images. We applied APP2 reconstruction with manual input terminal markers as the "gold standard," which had been shown to be comparable to human annotation (Xiao and Peng, 2013). Figure 9 shows the comparison results using these six reconstruction algorithms, as well as the manually curated result based on APP2. The best possible reconstruction result of each algorithm is shown.

Table 1 shows the comparison of computational costs in terms of tracing time and the maximum used memory of six different tracing algorithms. Our TReMAP has comparable tracing speed, and less maximum memory usage than other algorithms.

Table 1. Comparison of computational costs of TReMAP and other tracing algorithms



Algorithms	C147		c154		C168	
	Tracing Time(s)	Maximum memory(MB)	Tracing Time(s)	Maximum memory(MB)	Tracing Time(s)	Maximum memory(MB)
MOST (3D)	0.8	425.81	0.85	421.73	0.9	414.13
MOST (2D)	131.59	564.63	130.83	535.23	132.80	564.5
NeuTube	40.13	3191.33	115.42	3191.22	111.43	3199.41
Snake	85.29	8867.64	87.29	8864.88	85.74	8882.45
APP1	5.02	3990.94	5.54	4005.56	5.62	4011.31
TReMAP	6.99	313.81	6.61	391.25	10.29	349.14

To quantify the dissimilarity of a reconstruction compared to a "gold standard" reconstruction, two distance scores were calculated. First, we considered an "entire structure average score" that was defined as the average of the shortest distance between nodes in two reconstructions. We also used a "different structure average score," which was defined as the average visible spatial distance (>=2 voxels apart) between corresponding nodes in two reconstructions (detailed descriptions about these two scores can be found in (Peng, Ruan, Atasoy and Sternson, 2010)). Figure 10 shows evaluation of tracing accuracy for six different tracing algorithms. Entire structure average distance scores between the "gold standard" and each reconstruction in average of three images are 8.18 (MOST 3D), 8.57(MOST 2D), 4.42 (NeuTube), 8.59 (snake), 2.60 (APP1), and 3.98 (TReMAP). Different structure average distance scores between the "gold standard" and each reconstruction in average of three images are 15.06 (MOST 3D), 14.59 (MOST 2D), 12.21 (NeuTube), 19.76 (snake), 5.52 (APP1), and 9.06 (TReMAP). Overall, TReMAP has better performance than MOST 3D and 2D, NeuTube, and snake tracing, and has comparable performance to APP1 tracing.

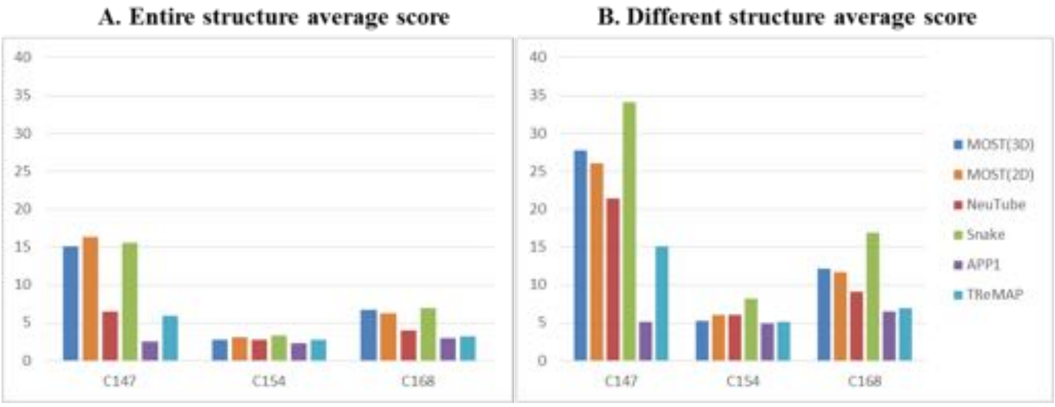
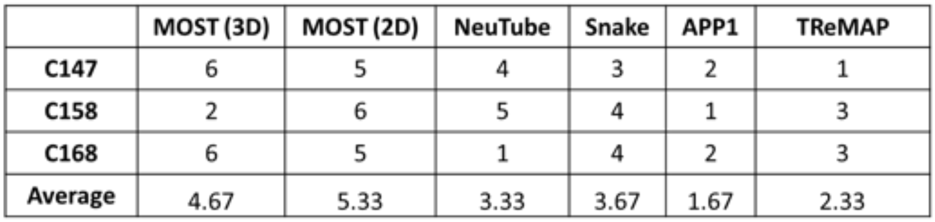


Figure 10. Evaluation of tracing accuracy for six tracing algorithms. The same color schemes were used in A and B.

To further test the topological precision, we applied BlastNeuron (Basic Local Alignment Search Tool for Neurons) for 3D neuron morphology evaluation (Wan, et al., 2015). BlastNeuron was designed to accurately and efficiently retrieve morphologically and functionally similar neuron structures based on global appearance, detailed arborization patterns, and topological similarity.

First, we used MST to generate the single neuron tree for these six reconstructions. Then the "gold standard" reconstruction is used as the query neuron to retrieve morphologically similar neurons from these six reconstructions. Comparing morphological and invariant moment features, BlastNeuron is able to rank the reconstruction candidates based on the given query neuron. Table 2 shows the ranking for all six reconstructions. Similar to the evaluation of distance scores, TReMAP has higher ranking score (2.33) than MOST 3D (4.67), MOST 2D (5.33), NeuTube (3.33), and Snake (3.67) on average.

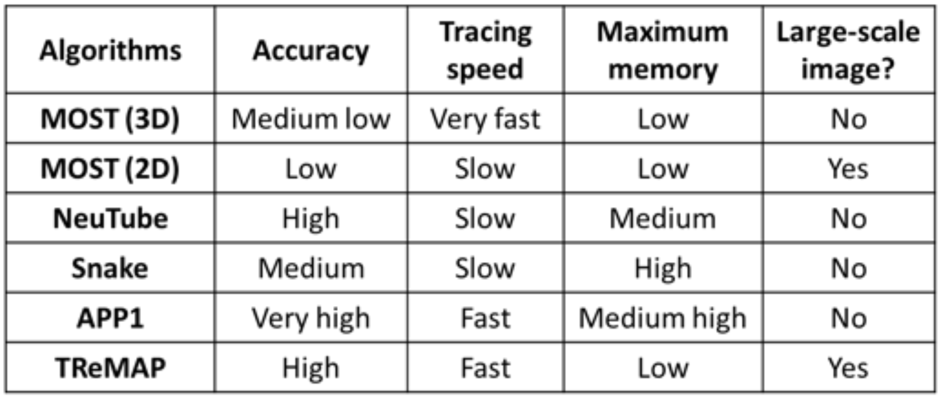
Table 2. Comparison of BlastNeuron ranking scores of reconstructions generated by six tracing algorithms



	MOST (3D)	MOST (2D)	NeuTube	Snake	APP1	TReMAP
C147	6	5	4	3	2	1
C158	2	6	5	4	1	3
C168	6	5	1	4	2	3
Average	4.67	5.33	3.33	3.67	1.67	2.33

Table 3 shows the comprehensive comparison of these six tracing algorithms in terms of accuracy, tracing speed, maximum memory usage, and the capability for large-scale image tracing. Even though the maximum memory usage of MOST 3D tracing is low, it still cannot be applied to the large-scale image directly because the entire image volume has to be preloaded first. Similar to MOST 3D tracing, NeuTube, Snake, and APP1 tracing algorithms have the same issue. For MOST 2D tracing, the 3D tracing result is assembled from 2D tracing of individual 2D image sequence, which makes it possible for large-scale image tracing. However, the tracing accuracy of MOST 2D is not satisfactory according to Figure 10 and Table 2. Overall, only TReMAP can be used for large-scale image tracing with comparable accuracy at much lower computational costs.

Table 3. Qualitative comparison of six tracing algorithms



Algorithms	Accuracy	Tracing speed	Maximum memory	Large-scale image?
MOST (3D)	Medium low	Very fast	Low	No
MOST (2D)	Low	Slow	Low	Yes
NeuTube	High	Slow	Medium	No
Snake	Medium	Slow	High	No
APP1	Very high	Fast	Medium high	No
TReMAP	High	Fast	Low	Yes

3.2. Application for large-scale microscopic images

One of the most notable advantages of TReMAP is the application for large-scale neuron images reconstruction, since it requires less computational time and much less memory usage. Figure 11A shows one section of a layer-2/3 neuron located in mouse V1. Since this entire mouse neuron crosses multiple sections, there are several separate neurite structures shown in this section.

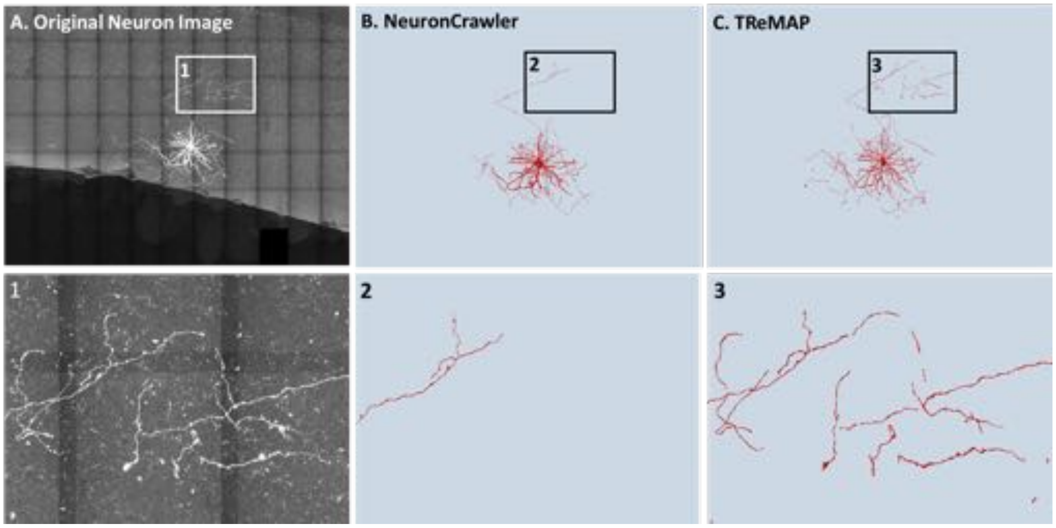


Figure 11. A large-scale neuron image and the automatic reconstructions by NeuronCrawler and TReMAP. The neuron is electroporated with a GFP plasmid and imaged in the resonant confocal at the Allen Institute for Brain Science. The image voxel size is 0.143 um x 0.143 um x 0.280 um. The image stack size is 10137 x 6451 x 313. The zoom-in views of three regions (1 2, and 3) are displayed respectively below A, B, and C.

Figure 11B and 11C show the reconstruction result using NeuronCrawler (Zhou, Sorensen and Peng, 2015) and our proposed TReMAP. From the figure we can see, TReMAP traced more structures than NeuronCrawler, since NeuronCrawler was not designed to trace multiple neuron structures. Moreover, for this large-scale image (image size is 19.063 GB with 10137 x 6451 x 313 voxels), the maximum used memory is only 1.45 GB, which means TReMAP can be used on common PCs.

4. Conclusions and Discussions

In this paper, we proposed a TReMAP algorithm for automated 3D neuron reconstruction. Our proposed algorithm can accurately and efficiently reverse 3D neuron structures from 2D tracing results and 3D image content. Experimental results showed that TReMAP achieves comparable accuracy to several recent 3D reconstruction algorithms on single neuron images via precise 2D tracing results from the APP2 algorithm and robust 3D reconstruction mapping from the Virtual Finger algorithm. In addition, because of 2D tracing and efficient local reverse mapping, TReMAP has less computational costs in terms of both running time and memory usage than other reconstruction algorithms.

Our TReMAP algorithm is tailored to efficiently handle the large-scale neuron image, which is contained in one 3D stack or multiple consecutive 3D sections. To further enhance the robustness of TReMap in the future, more projection angles could be used for reconstructing more complicated neuron structures.

Information sharing statement

The source code of TReMAP is openly available and is distributed along with the Vaa3D code repository (https://svn.janelia.org/penglab/projects/vaa3d_tools/hackathon/zhi/neurontracing_mip/).

Acknowledgement

We thank Nuno da Costa, Staci Sorensen, Julie Harris, Raina D'Aleo, and Soumya Chatterjee for providing the images of mouse neurons, Paloma Gonzalez-Bellido for providing the images of dragonfly neurons, Hanbo Chen and Yujie Li for comments. This work is supported by the Allen Institute for Brain Science.

References

DeFelipe, J., López-Cruz, P.L., Benavides-Piccione, R., Bielza, C., Larrañaga, P., Anderson, S., Burkhalter, A., Cauli, B., Fairén, A. and Feldmeyer, D. (2013) New insights into the classification and nomenclature of cortical GABAergic interneurons, *Nature Reviews Neuroscience*, **14**, 202-216.

Feng, L., Zhao, T. and Kim, J. (2015) neuTube 1.0: A New Design for Efficient Neuron Reconstruction Software Based on the SWC Format, *eneuro*, **2**, 0049-0014.

Fiala, J.C. (2005) Reconstruct: a free editor for serial section microscopy, *Journal of microscopy*, **218**, 52-61.

Gonzalez-Bellido, P.T., Peng, H., Yang, J., Georgopoulos, A.P. and Olberg, R.M. (2013) Eight pairs of descending visual neurons in the dragonfly give wing motor centers accurate population vector of prey direction, *Proceedings of the National Academy of Sciences*, **110**, 696-701.

Kawaguchi, Y., Karube, F. and Kubota, Y. (2006) Dendritic branch typing and spine expression patterns in cortical nonpyramidal cells, *Cerebral Cortex*, **16**, 696-711.

Krahe, T.E., El-Danaf, R.N., Dilger, E.K., Henderson, S.C. and Guido, W. (2011) Morphologically distinct classes of relay cells exhibit regional preferences in the dorsal lateral geniculate nucleus of the mouse, *The Journal of Neuroscience*, **31**, 17437-17448.

Lu, J., Fiala, J.C. and Lichtman, J.W. (2009) Semi-automated reconstruction of neural processes from large numbers of fluorescence images, *PLoS One*, **4**, e5655.

Ming, X., Li, A., Wu, J., Yan, C., Ding, W., Gong, H., Zeng, S. and Liu, Q. (2013) Rapid Reconstruction of 3D Neuronal Morphology from Light Microscopy Images with Augmented Rayburst Sampling, *PloS one*, **8**, e84557.

Myatt, D.R., Hadlington, T., Ascoli, G.A. and Nasuto, S.J. (2012) Neuromantic–from semi-manual to semi-automatic reconstruction of neuron morphology, *Frontiers in neuroinformatics*, **6**, 4.

Narayanaswamy, A., Wang, Y. and Roysam, B. (2011) 3-D image pre-processing algorithms for improved automated tracing of neuronal arbors, *Neuroinformatics*, **9**, 219-231.

Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E. and Ascoli, G.A. (2015) BigNeuron: Large-Scale 3D Neuron Reconstruction from Optical Microscopy Images, *Neuron*, DOI: 10.1016/j.neuron.2015.1006.1036.

Peng, H., Long, F. and Myers, G. (2011) Automatic 3D neuron tracing using all-path pruning, *Bioinformatics*, **27**, i239-i247.

Peng, H., Meijering, E. and Ascoli, G.A. (2015) From DIADEM to BigNeuron, *Neuroinformatics*, **13**, 9270, DOI: 9210.1007/s12021-12015-19270-12029.

Peng, H., Roysam, B. and Ascoli, G.A. (2013) Automated image computing reshapes computational neuroscience, *BMC bioinformatics*, **14**, 293.

Peng, H., Ruan, Z., Atasoy, D. and Sternson, S. (2010) Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model, *Bioinformatics*, **26**, i38-i46.

Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., Zhou, Z., Gonzalez-Bellido, P.T., Oh, S.W. and Chen, J. (2014) Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis, *Nature communications*, **5**, 4342.

Wan, Y., Long, F., Qu, L., Xiao, H., Hawrylycz, M., Myers, E.W. and Peng, H. (2015) BlastNeuron for Automated Comparison, Retrieval and Clustering of 3D Neuron Morphologies, *Neuroinformatics*, DOI: 10.1007/s12021-12015-19272-12027.

Wang, Y., Narayanaswamy, A., Tsai, C.-L. and Roysam, B. (2011) A broadly applicable 3-D neuron tracing method based on open-curve snake, *Neuroinformatics*, **9**, 193-217.

Wearne, S., Rodriguez, A., Ehlenberger, D., Rocher, A., Henderson, S. and Hof, P. (2005) New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales, *Neuroscience*, **136**, 661-680.

Wu, J., He, Y., Yang, Z., Guo, C., Luo, Q., Zhou, W., Chen, S., Li, A., Xiong, B. and Jiang, T. (2014) 3D BrainCV: Simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution, *NeuroImage*, **87**, 199-208.

Xiao, H. and Peng, H. (2013) APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree, *Bioinformatics*, **29**, 1448-1454.

Zhou, Z., Sorensen, S. and Peng, H. (2015) Neuron crawler: an automatic tracing algorithm for very large neuron images, *Proc. of IEEE 2015 International Symposium on Biomedical Imaging: From Nano to Macro*, 870-874.