3D Neuron Tip Detection in Volumetric Microscopy Images

Min Liu*†, Hanchuan Peng*, Amit K. Roy-Chowdhury†, and Eugene Myers*
* *Janelia Farm Research Campus, Howard Hughes Medical Institute.*
Email: {pengh, myersg}@janelia.hhmi.org
†*Department of Electrical Engineering, University of California, Riverside.*
Email: mliu009@ucr.edu, amitrc@ee.ucr.edu

***Abstract*—This paper addresses the problem of 3D neuron tips detection in volumetric microscopy image stacks. We focus particularly on neuron tracing applications, where the detected 3D tips could be used as the seeding points. Most of the existing neuron tracing methods require a good choice of seeding points. In this paper, we propose an automated neuron tips detection method for volumetric microscopy image stacks. Our method is based on first detecting 2D tips using curvature information and a ray-shooting intensity distribution model, and then extending it to the 3D stack by rejecting false positives. We tested this method based on the V3D platform, which can reconstruct a neuron based on automated searching of the optimal 'paths' connecting those detected 3D tips. The experiments demonstrate the effectiveness of the proposed method in building a fully automatic neuron tracing system.**

***Keywords*-3D tips; neuron tracing; ray-shooting;**

I. Introduction

The study of the neuronal cells' structure is crucial in the endeavor to understand how the brain circuits work [7]. In this process, the neuronal morphology extraction and neuron reconstruction from microscopic imaging data is a key component [11], [16]. Most existing methods are based upon the fact that a neuron has a branching tree structure, and then the problem of reconstructing a neuron network can be formulated as determining where the neuron branches go and how they further branch into finer arbors. The problem of neuron reconstruction from image stacks is also called neuron tracing.

In the past decades, many computational methods and tools have been developed for digital reconstruction of neurons from images [7]. An ideal set of seeds is very important for most of those existing tracing methods [9], [11], [16]. The seeds can be any starting points to reconstruct the neuron tree, such as local maximum points, skeleton seeds or ridge seeds [11], [16], [17]. In fact, the most intuitive seed points would be the starting points and the ending points of the neuron structure, essentially, the neuron tips, which then can be fed into the existing tracing framework, such as the 'shortest-path augmented deformable model', which has demonstrated its tracing capability in V3D, a fast and versatile 3D/4D/5D Image Visualization Analysis System for Bioimages and Surface Objects [10], [11]. V3D is fast becoming a standard tool for research in this area [6], [7]. One important feature of V3D platform is its automatic neuron-tracing ability using V3D-Neuron, if the neuron tips are given. The contribution of this paper is in detecting the tips so that a fully automated system can be developed.

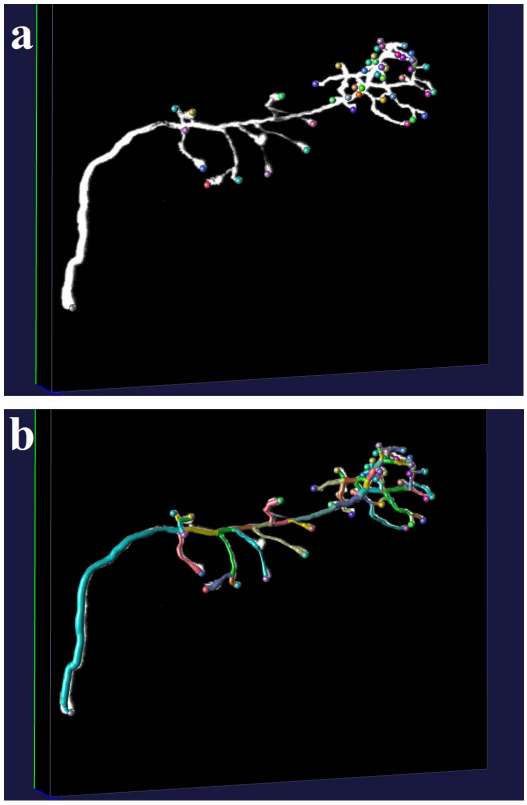

Figure 1. V3D-Neuron tracing based on 3D tips [11]. (a) Manually detected 3D tips of a *Drosophila* neuron [5]. 3D image stack: a Green Fluorescent Protein (GFP)-tagged neuron acquired by confocal microcopy; colored spheres: tips manually identified for this neuron. (b) Reconstructed neuron produced by V3D-Neuron based on those manually identified tips. Colored segments: the automatically reconstructed neurite structures.

V3D-Neuron is one of the state-of-the-art tracing tools to reconstruct the 3D structure of a neuron or a neurite tract from microscopy images. It can also immediately display the tracing results superimposed upon the raw image data, letting people proofread and correct the tracing results interactively. As shown in Figure 1, a *Drosophila* neuron is reconstructed using V3D-Neuron based on automated

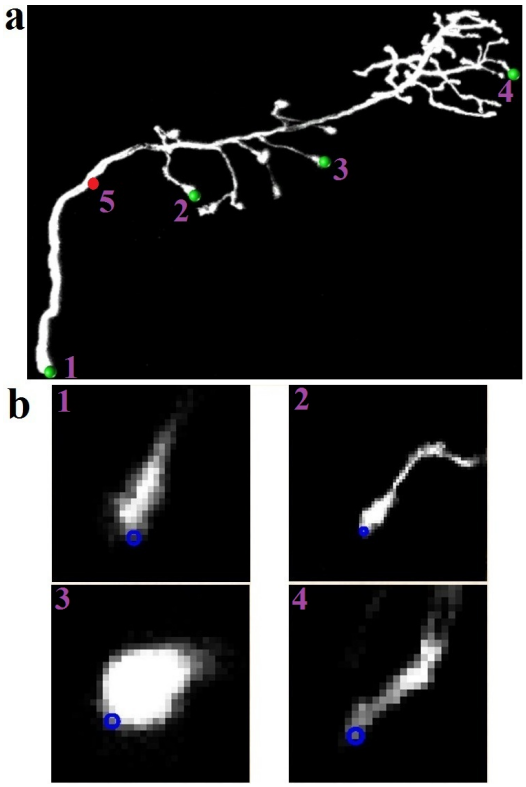

Figure 2. (a) 3D tips (numbers 1-4) denoted by green dots in the *Drosophila* neuron stack shown in Figure 1, and a fake 3D tip (number 5) denoted by red dot. (b) The 3D tips in slices–2D tips (blue circles). Usually a 3D tip will result in 2D tips in multiple slices, as shown in Figure 6. Here for each 3D tip, the 2D tips in just one slice are shown.

searching of the optimal 'paths' connecting a set of manually identified 3D tips. Those tips are locations pinpointed by the user to indicate where the tracing should begin and end (Figure 1a). The 'shortest path' algorithm finds a smooth tree-like structure in the image voxel domain to connect one tip (root) to all remaining tips with the least 'cost' [3]. Subsequently, the traced neuron is represented using the individual segments that connect tips and branching points (Figure 1b), which can be edited in 3D whenever needed.

As we know, neurons are usually very complicated in structure, for instance the human brain has 10^{11} neurons, so the manual tip identification is time-consuming and not practical. Furthermore, sometimes the arborization of a neuron tree is too complex and thus the tips are hard to visualize. If the 3D tips of a neuron stack can be automatically and accurately detected, this could be extremely useful for a series of neuron tracing algorithms such as V3D-Neuron.

A. Proposed Approach and Relation to Existing Work

The 3D neuron tips are small and very sensitive to noise, so it is a challenging task to robustly detect the 3D tips in neuron stacks. Although there is some work on neuron branching point (the location where the neuron tree bifurcates) detection and other kind of seed points detection [1], [16], [17], there is not much work on 3D neuron tip detection. However, there are some existing 2D corner detection methods, the idea of which can be borrowed to detect the 2D high curvature points, then further detect the 2D tip points, and then detect the 3D tips [4], [14].

B. Broad Overview of Solution Strategy

In this paper, we will show how to detect the 3D neuron tips for a given microscope image stack. Though the most intuitive way to detect the 3D tips would be to detect them directly in 3D space, it is not practical for two reasons. On one hand, the 3D high-curvature points detection is not robust, especially in the noisy case or when the neuron signal is weak; on the other hand, the high-curvature point detection in 3D is time-consuming, thus it is not suitable for real-time user-interactions. So we propose our tip detection method based on the 2D tip points detection.

The overall idea here is based on the fact that 3D tips (Figure 2a) will result in 2D tips in image slices (Figure 2b), and usually one 3D tip results in 2D tips in more than one slice; while 2D tips are not necessarily 3D tips. So we first find all 2D tips in every slice (xy-plane), then in the 3D space (z-direction) check whether those 2D tips are truly 3D tips or not. The diagram of this proposed 3D tip detection method is shown in Figure 3.

II. Detailed Methodology

As mentioned in Section I, one 3D tip may result in more than one 2D tips in a few slices (as illustrated in Figure 6). So we first do 2D tip detection, find the 2D tips as the 3D tip candidates, and then from which we pick the real 3D tips. The first step is to find the 2D tips, which can be detected by a high curvature point detection based method.

A. 2D High Curvature Point Detection

Methods of discrete curvature measures are based on the rate of tangent direction change. Generally, the tangent angle is measured by calculating the first difference of direction change over a certain curve segment, and then the curvature is measured by calculating the change of tangential direction over the curve segment.

A lot of existing methods can be modified to detect the 2D high curvature points, such as the corner detection approaches [4], [12]. Instead of using the curvature measures for corner detection, in this paper we detect the high curvature points based on the eigenvalues of the covariance matrix of data points on a curve segment, the detailed description of which can be found in [13], [14].

B. 2D Tip Point Detection

Each 2D high curvature point can be regarded as a 2D tip point candidate, which needs to be further checked to find whether it is indeed a tip point or not. Here we use the

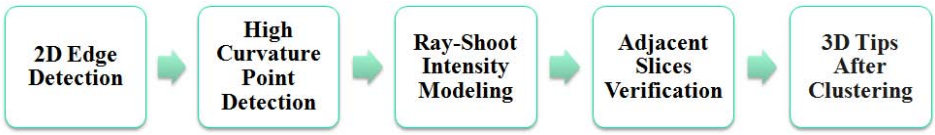

Figure 3. The diagram of the proposed automatic 3D neuron tips detection method in 3D microscopy image stacks.

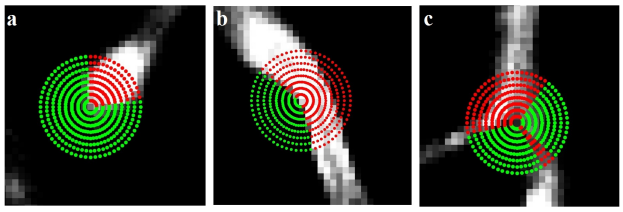

Figure 4. (a) A 2D tip with 16 'foreground rays' (in red color) out of 64 shooting rays, and the maximum angle between those 'foreground rays' is 0.47 π. By 'foreground rays', we mean those rays with average pixel intensity larger than a certain threshold, as described in Section 2.2. (b) A 2D non-tip with 39 'foreground rays', and the maximum angle between them is π. (c) A 2D branching point (non-tip) with 28 'foreground rays' and maximum angle π.

ray-shooting intensity distribution model to check whether a 2D high curvature point is a 2D tip point or not.

1) Ray-shooting Modeling: Not all high curvature points are 2D tips. Some of them are branching points and some of them are noise points, so we need a verification method to find the real 2D tips from those high curvature points. As shown in Figure 2 and Figure 4, the 2D tips have the 'tip' characteristic–they are located in the 'sharp peaks', from where if we shoot a number of rays outward, only a limited number of bright rays are within the foreground area. So we call such bright rays as 'foreground rays'. We propose to use the ray-shooting idea to model the intensity distribution of the neighboring pixels around a 2D tip candidate.

As shown in Fig 4, if we shoot a number of rays outward from the point of interest, for example, 64 rays, each with 8 pixels length, we can see that for a 2D tip (Fig 4a), the number of 'foreground rays' will be small, for example, 16. Here by 'foreground rays', we mean those rays with average intensity exceeding a certain threshold, as mathematically expressed in Equations (1)-(5). Moreover, those 'foreground rays' should stay together with the maximum angle between them also within a certain angle threshold, for example, $2\pi/3$. While in the non-tip and branching point cases (Figs 4b-c), the number of 'foreground rays' and the maximum angle between them are much larger.

Mathematically, for a point of interest P to be checked, the number of shooting rays are M, and each ray has N points. Let $I(i,j)$ denote the intensity matrix of those shooting rays $r(i)$, $i = 1:M, j = 1:N$, where each row represents the 'pixel' intensities along a ray. For those rays in the oblique direction, the 'pixels' are not in integer coordinates, so their intensities are computed through bilinear interpolation.

We first compute the average intensity $A(i)$ for every shooting ray,

$$A(i) = \frac{\sum_{j=1}^{N} I(i,j)}{N}, i = 1:M, \tag{1}$$

and find the maximum average intensity M_I,

$$M_I = \max_i(A(i)). \tag{2}$$

If it is lower than threshold T_0, it is considered to be a 'background area' point,

$$P \in BP_2, \quad \text{if } M_I < T_0, \tag{3}$$

where BP_2 stands for the 'background area' points set. So by 'background area' point, we do not mean a point is a background point (it can be a bright point due to noise), but it is in the background area and far from the foreground area. It has no shooting ray with average intensity exceeding the threshold T_0.

If $M_I > T_0$, a certain ratio $R(0 < R < 1)$ of the maximum average intensity, $M_I \times R$, is set as the threshold T to check whether a shooting ray is a 'foreground ray':

$$T = M_I \times R. \tag{4}$$

Those rays with average intensity values greater than T are regarded as the 'foreground rays', as those red rays shown in Figure 4, which are stored in the 'foreground ray' set Q, the size of which is n, as

$$\begin{cases} Q = \{i | A(i) > T, i = 1:M\}, \\ n = \#Q. \end{cases} \tag{5}$$

Furthermore, we find the maximum angle between those 'foreground rays',

$$M_A = \max_{p \in Q, q \in Q} \left(\text{ang}(r_p, r_q)\right), \tag{6}$$

where $\text{ang}(r_p, r_q)$ is the angle between two rays r_p and r_q.

For a 2D tip, n and M_A should be within certain ranges, i.e,

$$P \in TP_2, \quad \text{if } \begin{cases} T_1 < \frac{n}{M} < T_2, \\ M_A < T_3, \end{cases} \tag{7}$$

where TP_2 stands for the 2D tip set. A point P satisfying Equation (7) is considered to have the 'tip' characteristic.

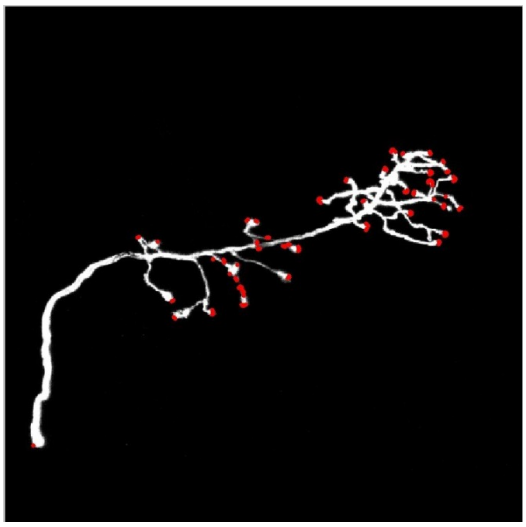

Figure 5. The detected 2D tips denoted by red dots in the maximum intensity projection [15] of the neuron stack shown in Figure 1, by the proposed 2D tip detection method.

In most cases, M_A is proportional to n. But it is not always true, such as in the branching point shown in Fig 4c. By this intensity distribution modeling, we can classify any point of interest (not just the 2D tip candidates in the object boundary) into one of three basic sets: 2D tip point TP_2, 'background area' point BP_2 and non-tip point NP_2.

The 2D tip detection result (red dots) on the maximum intensity projection [15] of the neuron in Figure 1 is shown in Figure 5. For a given neuron stack, we first detect the 2D tips in every slice, then using the adjacent slice verification method to find out which 2D tips are also 3D tips.

C. 3D Tip Detection

In fact, most of the 2D tips are formed by sectioning a 3D neuron into 2D slices in the xy-plane in the imaging process; such 2D tips are not the real starting point or ending point of a neuron branch. So we need to eliminate such fake 3D tips, and find the real 3D tips which are truly the starting point or ending point of a neuron signal.

After we get the 2D tips in every slice, we have some possible ways to find the real 3D tips. One possible approach is directly extending the 2D high curvature point detection method to detect the 3D high curvature point. However, usually the z resolution of the image stack is limited, so the computation of the z curvature value will be not easy. Another possible way is to do the 3D ray-shooting modeling after we get the 2D tips, but it would be computational expensive. So we propose a different approach that does not suffer from those shortcomings. It is a verification procedure on the adjacent slices to infer whether a 2D tip point satisfies the conditions of a 3D tip.

The basic idea here is that a real 3D tip should keep its 'tip' characteristic along the z direction for a couple slices, until it goes into the 'background area'. Specifically, for a 2D tip point P_k with coordinates (x_0, y_0) in the k_{th} slice S_k, if it is a real 3D tip, then in the adjacent slices $S_{k-m}, ..., S_{k-1}, S_{k+1}, ..., S_{k+m}$, those points $P_{k-m}, ..., P_{k-1}, P_{k+1}, ..., P_{k+m}$ in the same location (x_0, y_0) should be 2D tips, or 'background area' points,

$$P_k \in TP_3, \quad \text{if} \quad P_{k-m}, ..., P_{k+m} \in TP_2 \bigcup BP_2, \qquad (8)$$

where TP_3 stands for 3D tip point set. The logic behind this equation is that in the same location in the adjacent slices, the points should keep the 2D 'tip' characteristic until they gradually fall into the 'background area'. Those points can be either 2D tip point or 'background area' point, but cannot be non-tip point, as illustrated in Figure 6 (a)-(e). While a fake 3D tip can only keep its 'tip' characteristic just for very few slices and then becomes non-tip in the adjacent slices, as shown in Figure 6 (f)-(j).

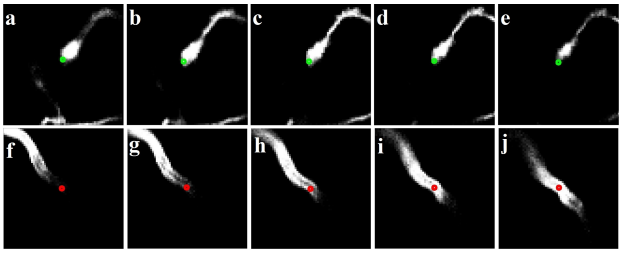

Figure 6. Real 3D tip and fake 3D tip. (c) A real 3D tip (denoted by number 2 in Fig 2a), with the points in the same xy-location denoted by green color in its adjacent slices –two upper slices (a)-(b) and two lower slices (d)-(e). (h) A fake 3D tip (number 5 in Fig 2a), with the same location points denoted by red color in the adjacent slices (f)-(g) and (i)-(j).

III. Experiment

The testing data are all from Digital Reconstruction of Axonal and Dendritic Morphology (DIADEM) Competition 2010 [2]. We test the proposed 3D tip detection method in this dataset and demonstrate the results mostly on Olfactory Projection Fibers (OP) images and Neocortical Layer (NL) images. The *Drosophila* OP stacks are acquired by confocal microscopy, with pixel size along the Z direction 3.03 times of that of the X and Y directions [5], while the NL image stacks are taken by two-photon lasers scanning microscopy in vivo, with pixel size along the Z direction is 3.40 times of that of the X and Y directions [8]. For a given stack, the preprocessing consists of median filtering and Gaussian filtering in every slice, and then each stack is patched with 3 all-black slices before the 1st slice and another 3 all-black slices below the last slice.

A. Choice of Parameters

In our proposed approach, the optimal values for the thresholds and other parameters in the equations are below,

$$\begin{cases} M = 64; N = 8; T_0 = 50; R = 1/2; \\ T_1 = 1/8; T_2 = 1/3; T_3 = 2\pi/3; m = 3. \end{cases} \qquad (9)$$

When it comes to the number of shooting rays, M, the larger M is, the more accurate our detection result is, and the default value in our experiment is 64. The choice of the ray length N depends on the thickness of the neuron branches. In the testing dataset, the average neuron branch thickness is round 8 pixels, so we choose 8 as the default ray length. About the threshold R, as a rule of thumb, we choose R to be 0.5. That is to say, we use the half-peak value of the maximum average intensity of all rays to separate the 'foreground rays' from the 'background rays'.

In the seed-based neuron tracing methods, false positive detection is preferable to false negative detection. The best values for T_0, T_1, T_2 and T_3 are learned in the experiment on real data, based on the criteria of minimum false negative detection and acceptable false positive detection.

For the verification in the adjacent slices, theoretically, the more adjacent slices taken into account, the more accurate of the 3D tip detection. But we cannot take too many adjacent slices, because we may run into a new neuron branch. In fact, the usual thickness of a neuron branch is 8-12 pixels in our dataset. Given the fact that the z-distance in image stack is about 3 pixels, a tip region may across 3-4 Z-slices, so 3 or 4 is a good choice for m in Equation (8), which has also proved to be effective in the experiments.

B. 2D Tip Detection Results

The 2D tip detection result on the maximum intensity projection of the OP1 stack (Figure 1a) is denoted by red color in Figure 5, from which we can see that most of the 2D tips are correctly detected. If some of the 2D tips are not successfully detected in one certain slice, that will not affect the 3D tip detection results, because the algorithm will find the 2D tips in the nearby xy-locations in the adjacent slices. So the proposed method is robust to false negative detection, and we will reject the positive detections by the adjacent slice verification procedure.

C. 3D Tip Detection Results

Then the experiment on entire stacks is done, with some typical results shown in Figures 7-8, where the detected 3D tips are superimposed on the original neuron stacks. Also, in order to demonstrate the usefulness of the detected 3D tips, the neuron tracing results by V3D-Neuron based on the detected 3D tips are also shown. V3D-Neuron uses the 'shortest path' method to connect those tip points and then trace the neurons. It treats individual image voxels as graph vertexes [11] and uses Dijkstra's algorithm [3] to find the shortest paths from every non-root tip to the root. It further refines the path using a deformable curve model [10].

Figure 7(a) is the detection result in the OP1 stack shown in Figures 1 and 2. With those detected 3D tips (by color dots), the V3D-Neuron can trace the whole neuron structure successfully, as shown in (b), where the colored segments are the automatically reconstructed neurite structures. Figure 8 shows the detection result of a very crowded case in OP4 stack, where some of the tips are very close to each other, and our method can effectively detect most of the tips.

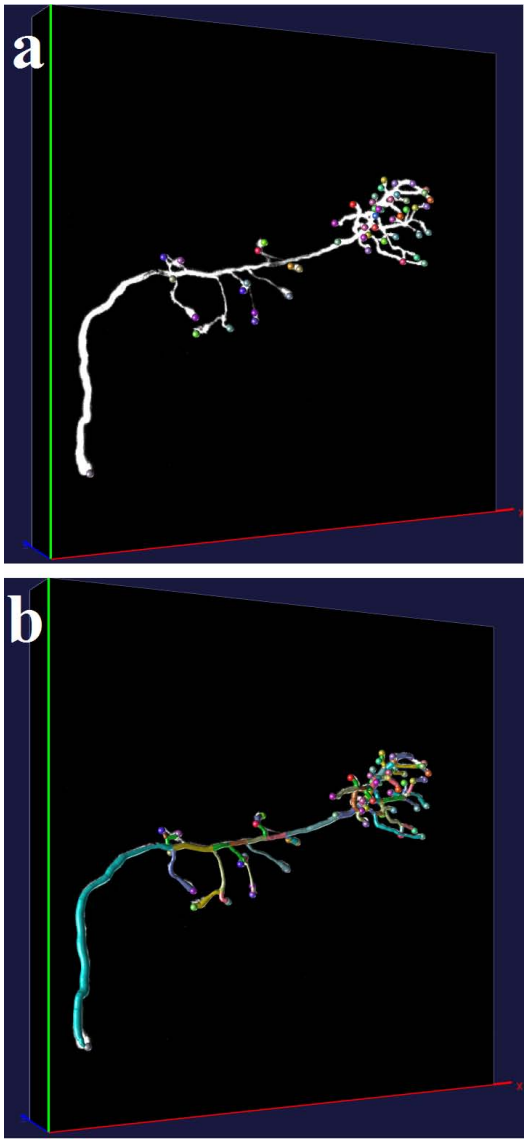

Figure 7. (a) The 3D tip detection results (color dots) for the OP1 stack shown in Figure 1. Some of the color dots seem to be not in the correct tip location due to the limitation of 2D representation. (b) The neuron tracing results by v3D-Neuron, based on the 3D tips detected in (a). The colored segments are the automatically reconstructed neurite structures.

D. Quantitative Results

In order to demonstrate how accurate the detection results are, we show the statistical results from 6 neuron stacks in Table 1. There is no similar work to compare, so we evaluate our result by comparing it to the 'ground truth' from the manual detection.

In Table 1, names in the 1st column denote different neuron stacks from DIADEM Competition 2010. The 2nd

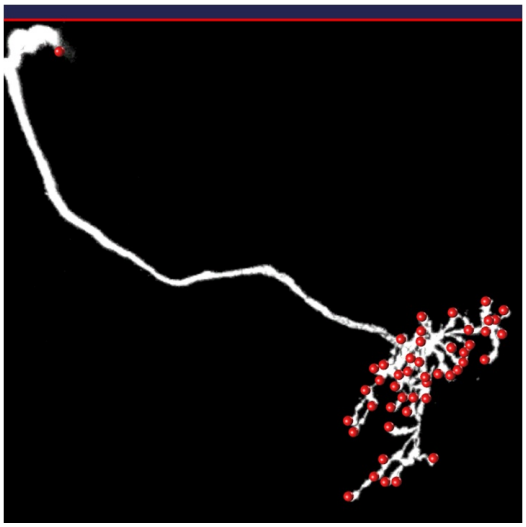

Figure 8. The 3D tips detected in the OP4 stack.

Table I
DETECTION ACCURACY.

Dataset	Tip #	False Positive #	False Negative #
OP1	48	2/48	1/48
OP3	18	1/18	0/18
OP4	46	4/46	2/46
OP5	17	1/17	0/17
OP6	23	2/23	1/23
NL1	165	10/165	7/165

column is the manfully produced 'ground truth' of how many 3D tips in each stack. The 3rd column is the number of 'tip' points false positively detected, while the 4th column is the number of 'tip' points false negatively detected. From this table we can see that the detection accuracy is from 91.3% to 95.8%. We can also see that the false negative detection ratio is lower than the false positive detection ratio, which is preferable in the seeding based tracing methods.

IV. CONCLUSION

Tracing of 3D neurite structures, which is one of the essential yet bottleneck steps in understanding brain circuit structure and function. Most of the existing tracing methods require a good choice of the seeding points, such as the starting points and ending points of the neuron branches. We propose an automated detection method to find the 3D tips in volumetric microscopy image stacks. We tested this method in the V3D-Neuron platform, which can reconstruct a neuron based on automated searching of the optimal 'paths' connecting the detected 3D tips. Experiments demonstrate the effectiveness of the proposed method in finding the 3D tips in neuron image stacks.

REFERENCES

[1] Y. Al-Kofahi, N. Dowell-Mesfin, C. Pace, W. Shain, J. N. Turner, and B. Roysam. Improved detection of branching points in algorithms for automated neuron tracing from 3d confocal images. *Cytometry A*, 73(1):36–43, 2008.

[2] K. Brown, G. Barrionuevo, A. Canty, V. De Paola, J. Hirsch, G. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G. Ascoli. The diadem data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9:143–157, 2011.

[3] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[4] X. C. He and N. H. C. Yung. Corner detector based on global and local curvature properties. *Optical Engineering*, 47(5):057008, 2008.

[5] G. S. Jefferis, C. J. Potter, A. M. Chan, E. C. Marin, T. Rohlfing, C. R. Maurer, Jr., and L. Luo. Comprehensive maps of drosophila higher olfactory centers: Spatially segregated fruit and pheromone representation. *Cell*, 128(6):1187 – 1203, 2007.

[6] A. Li, H. Gong, B. Zhang, Q. Wang, C. Yan, J. Wu, Q. Liu, S. Zeng, and Q. Luo. Micro-Optical Sectioning Tomography to Obtain a High-Resolution Atlas of the Mouse Brain. *Science*, 2010.

[7] E. Meijering. Neuron tracing in perspective. *Cytometry A*, 77(7):693–704, 2010.

[8] V. D. Paola, A. Holtmaat, G. Knott, S. Song, L. Wilbrecht, P. Caroni, and K. Svoboda. Cell type-specific structural plasticity of axonal branches and boutons in the adult neocortex. *Neuron*, 49(6):861 – 875, 2006.

[9] H. Peng, F. Long, and G. Myers. Automatic 3d neuron tracing using all-path pruning. *Bioinformatics*, 27(13):i239–i247, 2011.

[10] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson. Automatic reconstruction of 3d neuron structures using a graph-augmented deformable model. *Bioinformatics*, 26(12):i38 –i46, 2010.

[11] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature Biotechnology*, 28:348–353, 2010.

[12] K. Sohn, J. H. Kim, and W. Alexander. A mean field annealing approach to robust corner detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(1):82 –90, 1998.

[13] O. M. Tataw, M. Liu, A. Roy-Chowdhury, R. K. Yadav, and G. V. Reddy. Pattern analysis of stem cell growth dynamics in the shoot apex of arabidopsis. In *IEEE International Conference on Image Processing*, 2010.

[14] D. M. Tsai, H. T. Hou, and H. J. Su. Boundary-based corner detection using eigenvalues of covariance matrices. *Pattern Recognition Letters*, 20(1):31 – 40, 1999.

[15] J. W. Wallis and T. R. Miller. Three-dimensional display in nuclear medicine and radiology. *J. Nucl. Med.*, 32(3):534–546, 1991.

[16] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam. A broadly applicable 3-d neuron tracing method based on open-curve snake. *Neuroinformatics*, 9:193–217, 2011.

[17] J. Xie, T. Zhao, T. Lee, E. Myers, and H. Peng. Automatic neuron tracing in volumetric microscopy images with anisotropic path searching. In *Proceedings of the 13th international conference on Medical image computing and computer-assisted intervention*, pages 472–479, 2010.