

Neuron Crawler: An Automatic Tracing Algorithm for Very Large Neuron Images

Zhi Zhou, Staci A. Sorensen, and Hanchuan Peng*

Allen Institute for Brain Science, Seattle, WA 98103. * Corresponding author.

ABSTRACT

Automatic 3D neuron reconstruction for very large 3D light microscopy images remains to be a challenge in neuroscience. Few existing neuron tracing algorithms can be used with commonly available computers (laptops, desktops, or workstations) to efficiently and accurately reconstruct a neuron in image stacks that are tens of gigabytes or greater. We introduce a new automatic tracing algorithm called Neuron Crawler, which works by first tracing a region of interest (e.g., around the soma), and then iteratively tracing in adjacent image tiles to grow the neuron structure in 3D to its termination point within the image. Our experimental results show that Neuron Crawler can achieve reconstruction accuracy that is comparable to several state-of-the-art algorithms, but with much less computational cost.

Index Terms— 3D neuron reconstruction, large-scale image, Neuron Crawler, all-path-pruning

1. INTRODUCTION

Automatic 3D reconstruction of single neurons is very important for neuron morphology quantification and cell type classification. In order to trace the neuron accurately and efficiently, researchers have developed a number of neuron tracing algorithms [1-9].

To produce more detailed structures of single neurons, high-resolution 3D images of neurons are being collected in many ongoing neuroscience research efforts. For mammalian neurons, often a very large image stack will be generated. The size of such a dataset usually has 10s-100s of gigabytes. In this situation, many of the previously proposed algorithms for automated neuron reconstruction are not readily useful, as they normally would not be able to handle images of this size (volume). Most existing methods often need to read all image voxels into memory before the tracing algorithm can be used. Such approaches cannot easily scale with the amount of image data due to the limited amount of memory a computer (or even a cluster machine) has.

A potential solution for large-scale image tracing would be to divide such a large dataset into many smaller image tiles, reconstruct each individual image tile first, and then assemble these individual tracing results into one single reconstruction. However, the reconstruction of all such tiles is not necessarily effective since not all of them contain useful signals. It may also be difficult to assemble independently produced reconstructions.

To address these issues, we have developed an automatic 3D neuron tracing method called Neuron Crawler (Figure 1). The key advancement of Neuron Crawler is to trace a small image tile first, and continue to trace only the adjacent image tiles where the signal is continuous. Our new method avoids the two problems mentioned above and makes it possible now to reconstruct very large 3D images.

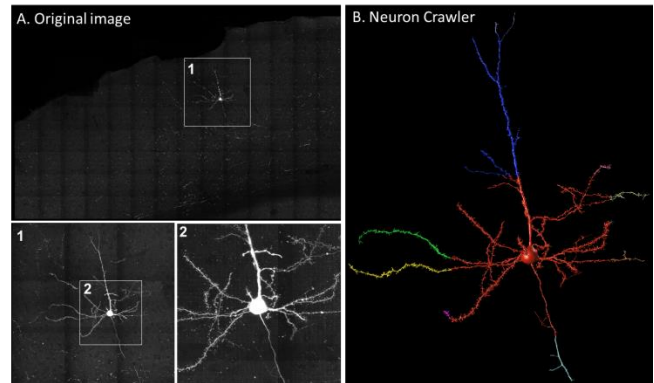


Figure 1. Images of a single neuron and the respective automatic reconstruction result generated by Neuron Crawler. A) Confocal image of a fluorescently-labeled mouse neuron (29.9 GB) shown at different resolutions. Original image is stitched from 126 3D image tiles. The image voxel size is 143 nm x 143 nm x 280 nm. B) Reconstruction generated by first tracing the tile with the soma (cell-body), followed by iterative tracing of all remaining image tiles. Different colors indicate different tracing orders.

2. METHOD

Neuron Crawler utilizes a basic tracing method to reconstruct neuronal processes within any of the small image tiles extracted from a large image stack. We note that many algorithms [1-9] may be used as the basic tracing module, given that such possible candidates are sufficiently fast and accurate. Here we describe the Neuron Crawler using All-Path-Pruning 2 (APP2) as the basic tracing module, as APP2 has been shown to be relatively fast and accurate based on pruning a dense initial reconstruction of a neuron to generate a compact representation of the neuron [1, 2].

2.1. Tracing the first image tile

To avoid the big cost of computer resources to directly load an entire large 3D image into computer memory, we use a much smaller 3D image tile as our starting location for the reconstruction. In this example, we use the soma location of the neuron as the root, and select the local 3D region of a large image for the first round of tracing (Figure 2). A user may specify the size of such a local region, but it can also be standardized easily (see Results).

Figure 2 shows the reconstruction result from the first image tile. We then detect all boundary terminal tips (green boxes in Figure 2). Of note, in microscopy images usually there is strong anisotropy in imaging, the z-dimension of the dataset is often much smaller than x- and y-dimensions. Thus for illustration purpose

here we consider only the boundary tips in the x-y planes. Clearly this scheme can be extended to all x-, y-, and z-directions without difficulty.

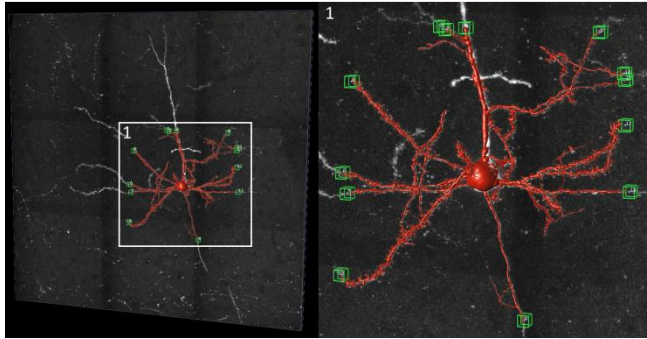


Figure 2. The reconstruction of the first image tile along with its boundary terminal tips. These selected tips are marked using green boxes.

2.2. Neuron Crawler

After detecting all terminal tips in the first round of reconstruction, we continue to search for potential continuous signals in the adjacent image tiles from any tip that is close to the border of the current tile to grow the neuron structure. This process is iterated until the neuron structure cannot grow any more. In this way, instead of tracing all small tiles extracted from the original image, our method works like a crawler, only crawling through the area with continuous signals and thus achieving good efficiency.

When we use a boundary terminal tip as the new starting location to trace an adjacent image tile, we require the adjacent tile has 10% overlap with the current tile, in order to avoid false continuation and increase the robustness of the tracing.

However, the overlapping area between adjacent tiles could cause over-tracing or some topological errors (Figure 3A) when the respective reconstructions from these tiles are assembled. We design a simple fusion method to reduce the chance to run into such errors. As typically a neuron reconstruction is represented by a series of reconstruction compartments that have different coordinates and radii, we check whether or not the overlap region of two reconstruction compartments from two adjacent tiles is greater than 50%. If yes, then we merge these two compartments by keeping only the sub-structure in the first reconstructed tile (Figure 3B). Otherwise we keep them as two separate compartments.

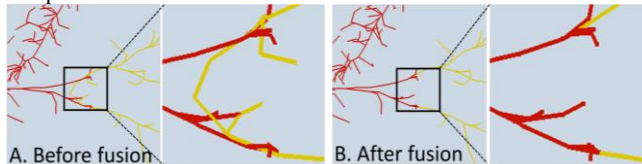


Figure 3. One examples of correcting the topological error by our fusion method. Note: these reconstructions are shown in skeleton mode for better visualization.

Figure 4 shows one example of tracing three adjacent image tiles. The overall Neuron Crawler algorithm can be found in Box 1.

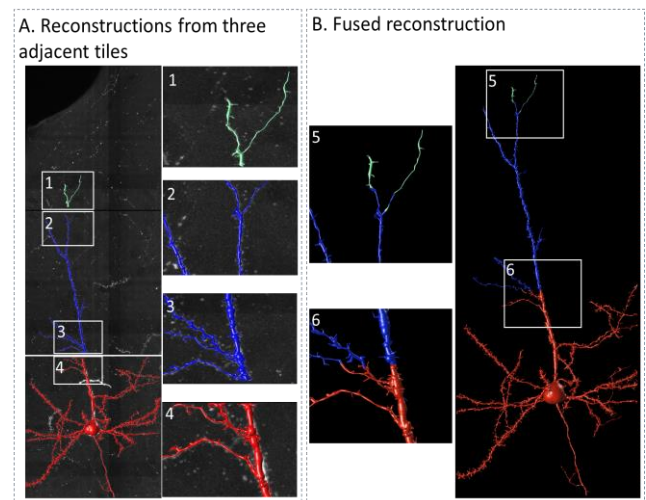


Figure 4. One example of tracing three adjacent image tiles using Neuron Crawler. A) Three reconstructions from three adjacent tiles, which have 10% overlap. B) Corrected, fused reconstruction of these three tiles. The zoomed area B5 is the fused result for A1 and A2, and B6 is the fused result for A3 and A4.

Box 1. Neuron Crawler algorithm

Input: soma location, an entire set of 3D image tiles, and the coordinates information

Output: the reconstruction result

- 1: Generate the first 3D image tile using the soma location as the center. Note: the soma location is selected by the user.
- 2: Use APP2 to reconstruct the image tile.
- 3: Check whether or not a terminal tip from the reconstruction is near the boundary. If yes, save it as the new starting location.
- 4: Use the starting location found in step 3 to reconstruct the adjacent tile with 10% overlap.
- 5: Fuse the reconstructions from adjacent tiles to avoid over-tracing and topological errors.
- 6: Iterate process steps 3 to 5, and keep tracing the remaining areas using the boundary tips until no more boundary tip can be found.

2.3. Selection of tile size

To quantify the robustness of Neuron Crawler related to different image-tile sizes, we compare the consistency among reconstructions that are generated using varying tile-sizes using a distance score that measures the average spatial distance for all nearest compartments of two neurons. It is used to measure the consistency of reconstructions of the same neuron generated by multiple methods [10].

Figure 5 shows the reconstructions using three different tile sizes, the corresponding tracing time, and the respective distance scores for all possible pairings of such comparison. As tile size decreases, the tracing time increases slightly since more image tiles are traced using a smaller tile size. It can be seen that there is a better consistency between the reconstructions of tile sizes 1024 and 512. For instance, only 2.88 voxels distance in this pair of reconstructions, compared to a much greater 10.25 voxels distance between that of tile sizes 1024 and 256. This indicates that for better robustness we should use bigger tile size as long as there is enough memory in the computer.

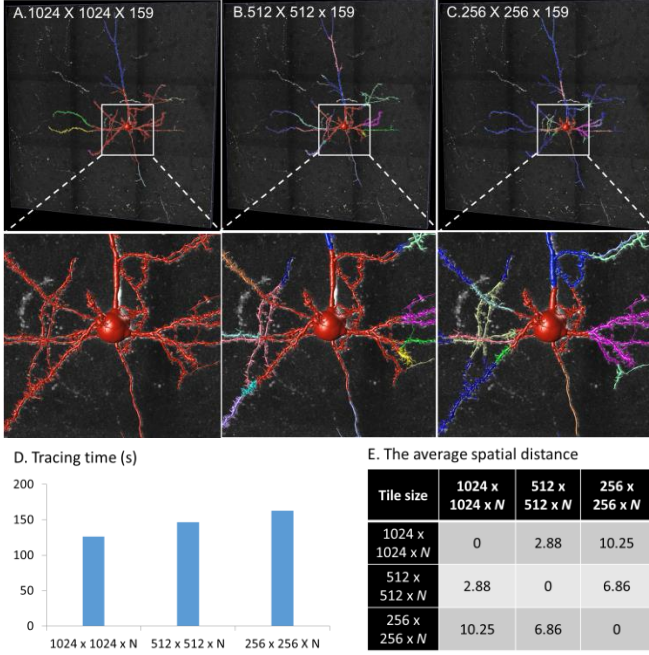


Figure 5. Three reconstructions and the distance scores based on different tile sizes. From A to C, the smaller tile size has more colored reconstruction, which means Neuron crawler traced more tiles with a smaller size. D shows the tracing time using these three tile sizes (N is the number of z-slices). E shows the distance score for each tile size pair.

We note that a larger tile will contain more background voxels, thus potentially slowing down the tracing process. Fortunately, since our basic tracing module is APP2, which is based on the well-known, fast-marching scheme to generate the initial reconstruction followed by pruning, the speed of this method is largely independent of the number of background voxels. Thus we can basically use the same tile size $1024 \times 1024 \times N$ (N is the number of z-slices) for a variety of large image datasets.

3. EXPERIMENTAL RESULTS

In order to evaluate the performance of the Neuron Crawler algorithm, we first compared the reconstruction results of Neuron Crawler and two other reconstruction approaches NeuTube [6] and Micro-Optical Sectioning Tomography (MOST) ray-shooting [9] (called “MOST ray-shooting” in the following) using the block based stitching approach. We used three neuron image stacks for testing. The image sizes of these three images are 1.2 GB (Image 1), 3.3 GB (Image 2), and 6.4 GB (Image 3) respectively.

Figure 6 shows the comparison reconstruction results generated by Neuron Crawler, NeuTube, and MOST ray-shooting. From the results we can see that the reconstructions generated by NeuTube and MOST ray-shooting using the block based stitching approach have lots of fragments and are not a single neuron tree. But Neuron Crawler can reconstruct a completed, single neuron tree.

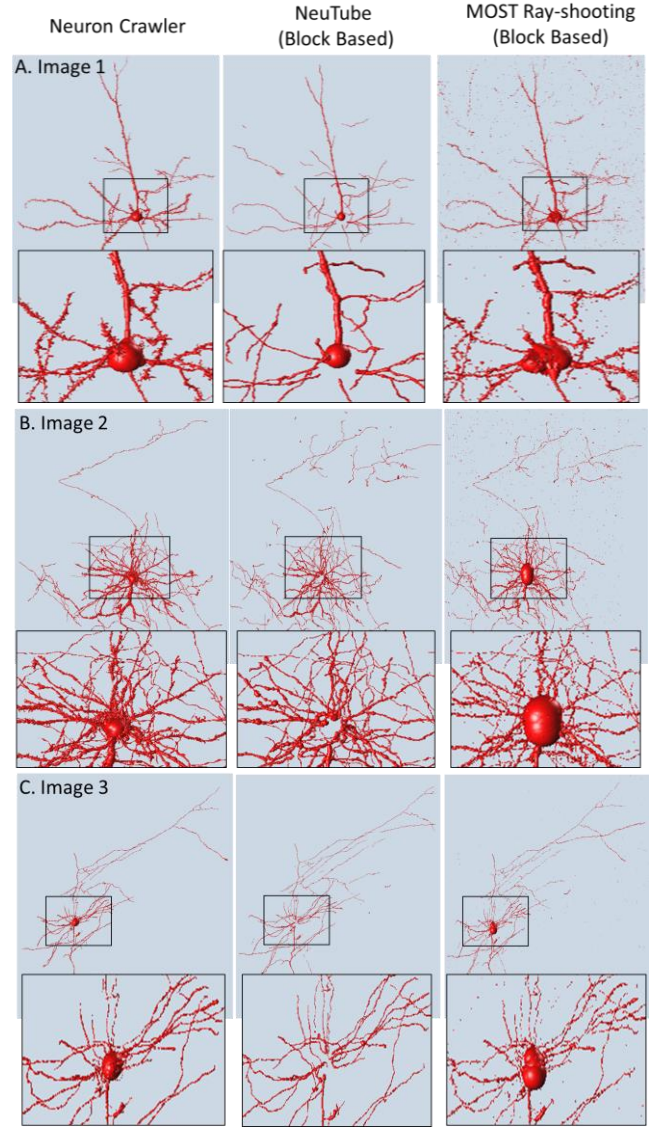


Figure 6. Comparison of neuron reconstruction results using three different reconstruction approaches. The reconstructions in the first column are generated by Neuron Crawler; the reconstructions in the second column are generated by NeuTube using the block based stitching, and the reconstructions in the third column are generated by MOST ray-shooting using the block based stitching.

We also compared reconstruction results of Neuron Crawler with those produced using APP2 directly (called “direct-APP2” in the following) to evaluate the accuracy of Neuron Crawler.

Figure 7 shows the reconstruction results generated by Neuron Crawler and direct-APP2 for three testing images. To compare these reconstructions, we calculated the distance score discussed in Section 2.3 for all three images. It shows that average spatial distances between Neuron Crawler and direct-APP2 reconstructions are merely 0.119, 1.407, and 2.230 voxels for Images 1, 2, and 3, respectively. The very small differences of the two sets of reconstruction indicate Neuron Crawler is accurate.

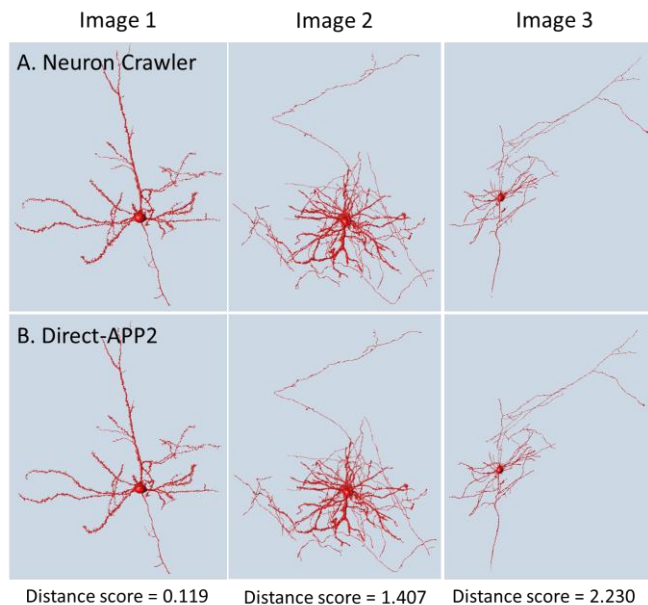


Figure 7. Comparison of neuron reconstruction results. In A, these three reconstructions are generated by Neuron Crawler using $1024 \times 1024 \times$ original z dimension tiles. In B, these three reconstructions are generated by direct-APP2.

Table 1 shows the maximum memory and tracing time when using these two automatic reconstruction algorithms. In our experiments, we used a Linux machine with 8 Intel(R) Xeon(R) CPU E5-1620 0 @ 3.60GHz, 128 GB memory, and C++ programming language. In the maximum memory comparison, Neuron Crawler needs much less memory than direct-APP2, and importantly, this is true even when the image size becomes larger. However, when the image size increases, the maximum memory used by direct-APP2 grows dramatically, ranging from 6 to almost 20 times that of Neuron Crawler.

For the tracing time, Neuron Crawler took slightly longer than APP2 for the three images. However when the total computational cost, i.e. the product of the maximum memory and the tracing time, is considered, Neuron Crawler has a much less computational cost than direct-APP2. For very-large scale images, the total cost of Neuron Crawler will be even less.

Table 1. Comparison of computational cost of Neuron Crawler and direct-APP2.

ID	Image size (GB)	Algorithm	Maximum Memory (GB)	Tracing Time(s)	Total Cost (Maximum Memory x Tracing Time)
Image 1	1.2	Neuron Crawler	3.97	109.72	435.59
		Direct-APP2	19.45	83.06	1615.52
Image 2	3.3	Neuron Crawler	4.86	370.51	1800.68
		Direct-APP2	51.46	193.97	9981.70
Image 3	6.4	Neuron Crawler	4.99	1047.90	5229.02
		Direct-APP2	98.05	693.83	68030.03

The biggest advantage of Neuron Crawler is that it enables reconstruction of very large 3D neuron images, for which other existing methods are not applicable. To evaluate the performance of our proposed method for very large images, Table 2 shows

results for images over 100 GB. Since these two images cannot be traced using direct-APP2 or any other methods accessible to us, we only show the computational cost using Neuron Crawler. It can be seen that the maximum memory used in these two >100 GB image is less than 13 GB. When using direct-APP2 or other existing tracing algorithms, it would have to use much more than 100 GB of memory. This best demonstrates the power of Neuron Crawler.

Table 2. The computational cost in Neuron Crawler for very-large scale neuron images.

ID	Image size (GB)	Maximum Memory (GB)	Tracing Time(s)	Total Cost (Maximum Memory x Tracing Time)
Large-scale Image 1	101.1	12.50	7628.60	57856.25
Large-scale Image 2	110.5	10.50	3458.34	36312.57

4. CONCLUSIONS AND AVAILABILITY

We propose Neuron Crawler as a solution to trace very large 3D neuron images. We have found Neuron Crawler has comparable reconstruction accuracy, but a much lower computational cost in terms of memory requirement and tracing time than existing methods. We have successfully used Neuron Crawler to reconstruct very large neuron images at the single-neuron resolution.

Neuron Crawler is available in Open Source at the Vaa3D source code repository (check <http://vaa3d.org> for the downloading address).

5. ACKNOWLEDGEMENTS

This work is supported by Allen Institute for Brain Science. We thank the large-scale image datasets contributed by Allen Institute.

REFERENCES

- [1] H. Xiao, and H. Peng, "APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree," *Bioinformatics*, vol. 29, no. 11, pp. 1448-1454, 2013.
- [2] H. Peng, F. Long, and G. Myers, "Automatic 3D neuron tracing using all-path pruning," *Bioinformatics*, vol. 27, no. 13, pp. i239-i247, 2011.
- [3] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam, "A broadly applicable 3-D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, no. 2-3, pp. 193-217, 2011.
- [4] A. Narayanaswamy, Y. Wang, and B. Roysam, "3-D image pre-processing algorithms for improved automated tracing of neuronal arbors," *Neuroinformatics*, vol. 9, no. 2-3, pp. 219-231, 2011.
- [5] P. Chothani, V. Mehta, and A. Stepanyants, "Automated tracing of neurites from light microscopy stacks of images," *Neuroinformatics*, vol. 9, no. 2-3, pp. 263-278, 2011.
- [6] T. Zhao, J. Xie, F. Amat, N. Clack, P. Ahammad, H. Peng, F. Long, and E. Myers, "Automated reconstruction of neuronal morphology based on local geometrical and global structural

- models,” *Neuroinformatics*, vol. 9, no. 2-3, pp. 247-261, 2011.
- [7] E. Türetken, G. González, C. Blum, and P. Fua, “Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors,” *Neuroinformatics*, vol. 9, no. 2-3, pp. 279-302, 2011.
 - [8] E. Bas, and D. Erdogmus, “Principal curves as skeletons of tubular objects,” *Neuroinformatics*, vol. 9, no. 2-3, pp. 181-191, 2011.
 - [9] J. Wu, Y. He, Z. Yang, C. Guo, Q. Luo, W. Zhou, S. Chen, A. Li, B. Xiong, and T. Jiang, “3D BrainCV: Simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution,” *NeuroImage*, vol. 87, pp. 199-208, 2014.
 - [10] 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*, vol. 28, no. 4, pp. 348-353, 2010.