Journal of Neuroscience Methods 184 (2009) 169–175

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Three-dimensional neuron tracing by voxel scooping

Alfredo Rodriguez[a,b,c,*], Douglas B. Ehlenberger[a,b,c], Patrick R. Hof[a,c], Susan L. Wearne[a,b,c]

[a] Department of Neuroscience, Mount Sinai School of Medicine, New York, NY, USA
[b] Laboratory of Biomathematics, Mount Sinai School of Medicine, New York, NY, USA
[c] Computational Neurobiology and Imaging Center, Mount Sinai School of Medicine, New York, NY, USA

ARTICLE INFO

Article history:
Received 16 March 2009
Received in revised form 16 July 2009
Accepted 16 July 2009

Keywords:
Centerline extraction
Neuron tracing
Image analysis
Medial axis

ABSTRACT

Tracing the centerline of the dendritic arbor of neurons is a powerful technique for analyzing neuronal morphology. In the various neuron tracing algorithms in use nowadays, the competing goals of computational efficiency and robustness are generally traded off against each other. We present a novel method for tracing the centerline of a neuron from confocal image stacks, which provides an optimal balance between these objectives. Using only local information, thin cross-sectional layers of voxels ('scoops') are iteratively carved out of the structure, and clustered based on connectivity. Each cluster contributes a node along the centerline, which is created by connecting successive nodes until all object voxels are exhausted. While data segmentation is independent of this algorithm, we illustrate the use of the ISODATA method to achieve dynamic (local) segmentation. Diameter estimation at each node is calculated using the Ray-burst Sampling algorithm, and spurious end nodes caused by surface irregularities are then removed. On standard computing hardware the algorithm can process hundreds of thousands of voxels per second, easily handling the multi-gigabyte datasets resulting from high-resolution confocal microscopy imaging of neurons. This method provides an accurate and efficient means for centerline extraction that is suitable for interactive neuron tracing applications.

1. Introduction

The centerline of a structure, classically used to analyze arbitrary shapes (Blum, 1967), is well-suited to the many complex branching structures that occur in nature. In medical imaging it is successfully applied in the analysis of two-dimensional (2D) and three-dimensional (3D) data obtained through a variety of imaging techniques including ultrasound, X-ray, magnetic resonance imaging (MRI), computed tomography (CT), and fluorescence and brightfield microscopy. While many algorithms have been devised for the general problem of centerline extraction of tubular structures, few methods have been developed to deal with the unique problems that arise in automated neuron tracing. Limited imaging resolution, the existence of secondary surface structures (dendritic spines), and uneven labeling and background illumination, all pose specific challenges that must be addressed by a neuron tracing algorithm. Early methods relied on voxel thinning or the medial axis transform (Lee et al., 1994; Borgefors et al., 1999) for extracting the centerline of neurons (Koh et al., 2002; He et al., 2003; Wearne et al., 2005). Due to the iterative nature of the voxel thinning process, this approach tends to be computationally intensive. Also because it requires binarized images (object and background) it relies heavily on the selection of a global threshold intensity value, which may be difficult to obtain for large datasets with significant variations in signal levels and background illumination.

More recent methods tend to use pattern recognition routines to track the position of each dendrite from an initial set of seed points, without the need for global image operations such as segmentation, distance transform, or image derivatives (Al-Kofahi et al., 2002; Streekstra and van Pelt, 2002; Schmitt et al., 2004; Santamaría-Pang et al., 2007; Myatt et al., 2006). These methods limit the number of voxels processed while extracting the centerline to those in and near the structure, and are often referred to as tracing or exploratory algorithms. For application to tracing sparse structures like neuronal morphology from microscopy images, where object voxels represent only a small fraction of the overall dataset, these algorithms can have a significant speed advantage over more traditional methods, such as voxel thinning and medial axis transforms. Because they act locally, pattern recognition-based algorithms are also able to adjust the tracing parameters dynamically to cope with local variations in morphology and image quality. Al-Kofahi et al. (2002, 2003), for example, used correlation kernels to evaluate the edge response for a number of combinations of radius and orientation at each point along the structure. The length of the kernel is adjusted dynamically depending on the curvature of the structure. Although the basic algorithm does not take junctions into account, an improvement has been proposed for the detection

* Corresponding author at: One Gustave L. Levy Place, Box 1065, New York, NY 10029, USA. Tel.: +1 212 241 8490.
E-mail address: alfredo.rodriguez@mssm.edu (A. Rodriguez).

0165-0270/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2009.07.021

of branch points (Al-Kofahi et al., 2008). Streekstra and van Pelt (2002) make use of Gaussian image derivatives for centerline detection. Each centerline node is detected by convolution of the data just ahead of the previous centerline point with first and second order derivatives of a Gaussian kernel, from which a centerline position and direction are computed. While this method has been shown to be effective at low resolution, its performance on high-resolution data in the presence of dendritic spines has not been demonstrated. The method of Schmitt et al. (2004) is intended for use in computer-assisted systems and fits a generalized cylinder model to the data using an active contour to construct a centerline between two manually defined points along the structure. Medial points are placed along the resulting curve and their location and radius are optimized according to two separate medialness functions. Santamaría-Pang et al. (2007) proposed a method that simulates a propagating wave through the data. By favoring faster propagation through voxels that score well according to a tubular morphological operator and outward flux in image intensity, the centerline can be constructed from points of maximum distance from the origin for a given time interval. Myatt et al. (2006) proposed a method that uses particle filters to reconstruct dendritic segments. At each step, the displacement from the previous position and the current radius of the segment are optimized using differential evolution (Storn and Price, 1997) by maximizing the difference in mean background and foreground intensity.

While these newer, pattern recognition-based algorithms minimize the number of voxels processed, those savings are offset by the computational cost inherent in the complexity of each of the algorithms. Such computational costs are incurred at each position along the structure by either searching for an optimal configuration for the corresponding metric or by relying on a numerically intensive metric that can overcome typical imaging artifacts. For extracting high-resolution, artifact-free morphologies from microscopy data, user interaction for semi-manual error correction will always be necessary, making speed an essential characteristic. The use of existing methods as interactive reconstruction tools for high-resolution data is therefore not optimal. Here, we present a tracing method based on iterative extraction of thin cross-sectional layers of object voxels that minimizes voxel processing, and has low computational complexity. It works equally well in either 2D or 3D data without any modifications, and is very fast, allowing interactive semi-automated extraction of neuronal morphology.

2. Materials and methods

2.1. Data acquisition and preprocessing

A pyramidal neuron from the prefrontal cortex of a macaque monkey was used for the validation studies and development of centerline extraction. An adult male long-tailed macaque monkey (*Macaca fascicularis*) was deeply anesthetized with ketamine hydrochloride (25 mg/kg im) and sodium pentobarbital (20–35 mg/kg iv), intubated, and mechanically ventilated. The animal was perfused transcardially with cold 1% paraformaldehyde in phosphate buffer (pH 7.4) for 1 min followed by cold 4% paraformaldehyde and 0.125% glutaraldehyde for 12 min at about 250 ml per minute. The brain was removed from the skull, cut into 4-mm-thick coronal blocks and postfixed no longer than 2 h in 4% paraformaldehyde at 4 °C. These blocks were placed in phosphate-buffered saline (PBS) with 0.01% sodium azide at 4 °C. These procedures were conducted according to National Institutes of Health (NIH) guidelines for animal research and approved by the Institutional Animal Care and Use Committee (IACUC) at Mount Sinai School of Medicine. The animal whose materials were used for the present study was sacrificed in the context of unrelated experiments (Duan et al., 2002, 2003).

Brain blocks were then coronally sectioned at 200 μm on a Vibratome (Leica, Nussloch, Germany). The sections were incubated in 4,6-diamidino-2-phenylindole (DAPI; Sigma, St. Louis, MO, USA), a fluorescent nucleic acid stain, for 5 min, mounted on nitrocellulose filter paper and immersed in PBS. Using DAPI as a staining guide, individual layer II/III pyramidal neurons of the frontal cortex were loaded with 5% Lucifer Yellow (Molecular Probes, Eugene, OR, USA) in distilled water under a DC current of 3–8 nA for 10 min, or until the dye had filled distal processes and no further loading was observed (Duan et al., 2002, 2003). Tissue slices were then mounted and coverslipped in Permafluor.

Image stacks were collected on a TCS-SP confocal laser scanning microscope (Leica Microsystems, Heidelberg, Germany) using a 100 × 1.4 NA PlanApo oil immersion objective lens at a resolution of 1024 × 1024 pixels. At this magnification, without any optical zooming, the images have a field size of 100 × 100 μm and pixel dimensions of 0.098 μm × 0.098 μm. For through-focus imaging, to approximate cubic voxels, optical sections were collected every 0.081 μm using a scan stage with a step resolution of 40 nm and saved to the hard drive. Before each stack was collected, gain and offset settings were optimized to limit the number of saturated voxels along the dendrite and the number of underexposed voxels in the background. Once a column had been imaged through-focus, the stage was translated in the *XY* plane to the adjacent column allowing for a 10–20% overlap to ensure accurate registration when aligning and stitching adjacent stacks. Then through-focus imaging was repeated on the adjacent column until the entire volume occupied by the neuron was imaged.

Individual stacks where then deconvolved using AutoDeblur deconvolution software (MediaCybernetics, Bethesda, MD, USA) using a blind deconvolution algorithm with a theoretically estimated point spread function and then integrated into a single volume using software developed in our laboratory for volume integration and alignment (VIAS, available at http://www.mssm.edu/cnic/tools.html) as previously described (Rodriguez et al., 2003; Wearne et al., 2005).

2.2. Algorithm overview

For the purpose of simplifying the following presentation we will assume that object voxels are readily distinguishable from the background in the image volume. Data segmentation is discussed in more detail in Section 2.8.

Given a tubular branching structure (Fig. 1a), the algorithm uses a region growing approach to generate clusters of voxels iteratively along the structure based on connectivity and physical location. Starting from a seed voxel inside the structure, each iteration creates a number of clusters from voxels directly connected to clusters of the previous iteration (Fig. 1b). These clusters are in turn used to position the nodes that define the centerline of the structure (Fig. 1c and d).

On the first iteration, which serves as the initialization step, a cluster is created which contains a single user-selected voxel inside the structure. A corresponding centerline node is also created having coordinates equal to the center of this voxel. On each subsequent iteration, i, the algorithm progresses through each jth cluster, $C_{i-1,j}$, of the previous iteration $i-1$ (blue voxels, Fig. 2a). For each cluster $C_{i-1,j}$, the algorithm collects all unvisited object voxels that are in its 26-connected neighborhood. These voxels are then grouped into a number of connected components (red and orange voxels, Fig. 2a). Each connected component, k, forms a new cluster, $C_{i,k}$, and a corresponding node, $N_{i,k}$, is created to form the centerline. Cluster $C_{i-1,j}$ is termed the parent of all $C_{i,k}$ child clusters and all nodes $N_{i,k}$ are connected to their parent node, $N_{i-1,j}$, along the centerline. The algorithm terminates when no unvisited object voxels connected to any cluster $C_{i-1,j}$ exist. Computing the physical locations of each

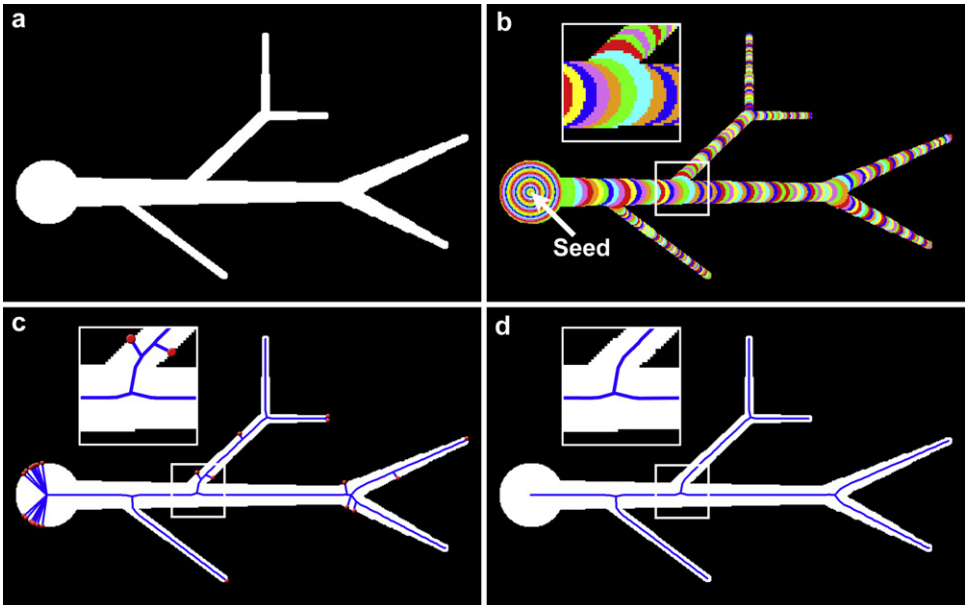

Fig. 1. Cluster and centerline for synthetic branching structure. (a) Initial dataset showing a branching, tubular structure. (b) Clusters are shown in distinct colors from a starting seed location. Inset shows a closeup view of clusters near a branch point. (c) The resulting centerline (in blue) before removal of spurious end nodes (shown as red circles). (d) The centerline after pruning of spurious end nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

new node as well as the addition of voxels into each new cluster is discussed in the next two subsections.

2.3. Node positioning

On each iteration, i, the algorithm creates a number of clusters $C_{i,k}$ from unvisited voxels directly connected to a cluster $C_{i-1,j}$ of the previous iteration (Fig. 2a). In order to position the corresponding new node, $N_{i,k}$, at a location along the centerline of the structure (Fig. 2b), its position, $\vec{\boldsymbol{N}}_{i,k}$, is computed based on the position of the parent node, $\vec{\boldsymbol{N}}_{i-1,j}$, the center of mass of the new cluster, $\vec{\boldsymbol{Cm}}_{i,k}$, and its size, $S_{i,k}$, relative to its parent's size, $S_{i-1,j}$. The size of a cluster is approximated by the length of the diagonal of its axis aligned bounding box (AABB) (Fig. 2a).

The position of each new node is given by the expression:

$$\vec{\boldsymbol{N}}_{i,k} = \begin{cases} \vec{\boldsymbol{N}}_{i-1,j} + 0.5^{(S_{i,k}/S_{i-1,j})} \times (\vec{\boldsymbol{Cm}}_{i,k} - \vec{\boldsymbol{N}}_{i-1,j}) & \text{if} \quad S_{i,k} \le S_{i-1,j} \\ \vec{\boldsymbol{N}}_{i-1,j} + 0.5^{(S_{i-1,j}/S_{i,k})} \times (\vec{\boldsymbol{Cm}}_{i,k} - \vec{\boldsymbol{N}}_{i-1,j}) & \text{if} \quad S_{i,k} > S_{i-1,j} \end{cases} \tag{1}$$

where: $\vec{\boldsymbol{N}}_{i,k}$ is the position of the new node, $\vec{\boldsymbol{N}}_{i-1,j}$ is the position of the node for the parent cluster, $S_{i,k}$ is the length of the diagonal of the current cluster's AABB, $S_{i-1,j}$ is the length of the diagonal of the parent cluster's AABB, and $\vec{\boldsymbol{Cm}}_{i,k}$ is the center of mass of the new cluster.

This expression causes the position of the new node to advance (with respect to the previous node) as a function of the size change (Fig. 2b). When the new connected component, $C_{i,k}$, is of the same size as the parent component, $C_{i-1,j}$, the new node, $N_{i,k}$, is positioned halfway between the center of mass of the connected component and the node of the parent cluster, $N_{i-1,j}$. When the connected component is significantly smaller or bigger than the parent component, the new node position tends to approach the center of mass. For tree-like tubular structures, such as neurons, this calculation places each new node at a location that closely follows the centerline of the object.

2.4. Voxel scooping

Each node position is also used to expand its corresponding cluster by iteratively adding unvisited object voxels in its vicinity. For

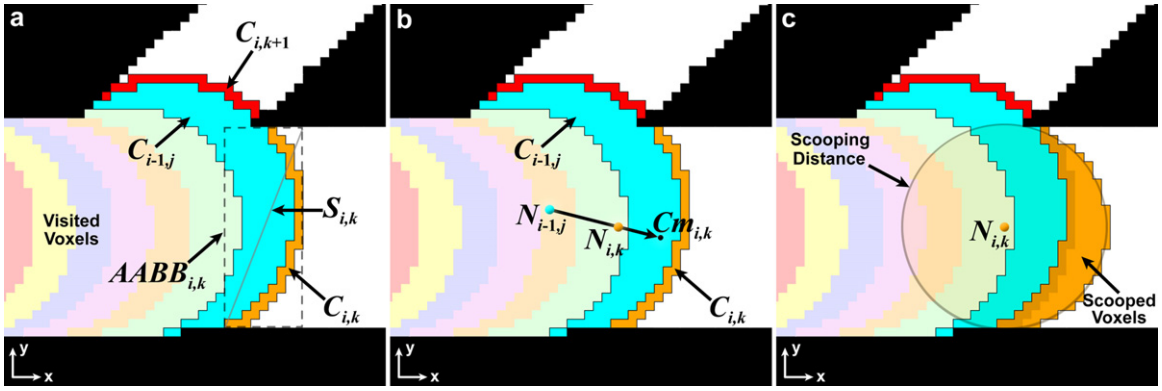

Fig. 2. Two-dimensional schematic of cluster formation and voxel scooping. (a) $C_{i,k}$ (orange) and $C_{i,k+1}$ (red) are formed from connected components of voxels around parent cluster $C_{i-1,j}$ (cyan). (b) The node $N_{i,k}$ (orange dot) for cluster $C_{i,k}$ has been computed using Eq. (1). (c) Unvisited object voxels (bright orange) within the scooping distance (overlayed circle) of $N_{i,k}$ are added to cluster $C_{i,k}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

each cluster $C_{i,k}$, the algorithm computes the maximum distance of any of its voxels to the corresponding node position $\vec{N}_{i,k}$. This distance is termed the *scooping distance* of the cluster. We then proceed to evaluate unvisited object voxels iteratively in the 26-connected neighborhood of voxels already in the cluster. Any of these voxels having its center within the scooping distance from $\vec{N}_{i,k}$ is also added into the cluster (Fig. 2c). This effectively allows each cluster to advance into the structure while adjusting for directional changes.

2.5. Pruning of short branches

Upon completion of the scooping process, the resulting centerline will have a number of short branches. In this context a branch is defined a series of nodes connected in a tree-like structure. Some of these short branches result from spurious end nodes caused by irregularities along the surface of the structure (red dots Fig. 1c). Other short branches result from secondary surface structures, such as dendritic spines, that should not be part of the dendritic centerline. We remove short branches based on their absolute length as well their lengths relative to the scooping distance of their attachment point. The length of a branch is defined as the maximum path length that can be traveled on the branch between its point of attachment and any tip of the branch. While a detailed description of the algorithm used to compute these lengths efficiently is beyond the scope of this paper, it should be noted that the running time added by this computation is negligible in comparison to the duration of the scooping process. In this implementation, a branch is removed if its length divided by the scooping distance of its attachment node is less than a user-specified value or its length falls below a second user-defined parameter. Fig. 1d shows the resulting centerline after removal of short branches.

2.6. Pseudo-code

The following is a pseudo-code implementation of the tracing algorithm. The subscript indicates the iteration and instance of each cluster or node.

```
MARK all voxels as unvisited
SELECT seed voxel

ITERATION 1:
CREATE cluster C_{1,1} from seed voxel
MARK seed voxel as visited
CREATE node N_{1,1} at center of seed voxel

ITERATION i > 1:
WHILE any clusters exist in previous iteration i − 1
  FOR every cluster C_{i−1,j} created in the previous iteration i − 1
    FIND all unvisited object voxels in 26-connected neighborhood of C_{i−1,j}
    MARK all found voxels as visited
    FOR every connected component k of found voxels
      CREATE cluster C_{i,k} from voxels of connected component k
      CREATE node N_{i,k} at location given by Eq. (1)
      CREATE connection from node N_{i,k} to node N_{i−1,j}
      SET scooping distance to the largest distance of any voxel in C_{i,k} to N_{i,k}
      WHILE any connected, unvisited, object voxel within scooping distance
        from N_{i,k}
        ADD voxel to cluster C_{i,k} and MARK voxel as visited
      END WHILE
    END FOR
  END FOR
END WHILE
```

2.7. Diameter estimation

For the purpose of extracting models of neuronal structures from confocal or multiphoton laser-scanning microscopy images, the centerline produced by the algorithm can be complemented with an estimate of diameter at each node. This makes the output compatible with most neuronal modeling and morphometry tools (Hines and Carnevale, 2001; Bower and Beeman, 1998; Scorcioni and Ascoli, 2001), which expect the basic structure of neurons to be represented as a series of connected segments of specified diameters. While a number of methods are available for diameter estimation our particular implementation uses the 2D variant of our Rayburst Sampling algorithm (Rodriguez et al., 2006) to estimate diameters at each node. Rayburst Sampling computes an estimate of diameter by measuring the minimum surface-to-surface span inside a tubular structure (see Rodriguez et al., 2006, for full mathematical details). For structures assumed to have an approximately radially symmetric cross section, 2D Rayburst run in the *XY* plane is insensitive to residual *Z*-axis smear from incomplete deconvolution, yielding a reliable estimate of the structure's diameter irrespective of its orientation within the image stack.

2.8. Dynamic data segmentation

Unsupervised image segmentation is a complex topic and many methods have been devised for it (see Pham et al., 2000 for review). While our tracing algorithm is independent of the method chosen for data segmentation, our current implementation uses an adaptation of the ISODATA algorithm (Ridler and Calvard, 1978) to compute a dynamically adjusting threshold along the structure as described in Rodriguez et al. (2008). Each cluster is segmented based on the ISODATA threshold computed over a cubic section of data centered on its parent's node. For efficiency reasons, we use scattered ramdom sampling inside the cubic section to limit the number of intensity samples to around 1000 voxels. Using ISODATA also provides a simple means to detect low contrast conditions used to create an end point in the centerline.

3. Results

3.1. Accuracy

To validate the accuracy of the centerline detected by the algorithm, we generate a consensus centerline, or gold standard, from a series of manually positioned centerline nodes by three trained human operators.

We start by selecting a section of the test dataset (see Section 2.1) containing a branching dendritic segment oriented mostly parallel to the optical plane (Fig. 3a). We then traced this section using the algorithm to obtain a starting centerline. Each node in this centerline was then perturbed by a random amount within the optical plane and in a direction perpendicular to the dendrite to obtain a *perturbed tracing* (Fig. 3b). Each perturbation ranged from zero to 0.5 μm (approximately 1.5 times the average dendritic radius for the segment) to either side. Perturbing the model this way allows us to preserve the topology of the tracing, essential for comparison between tracings, while removing information about the initial placement of the centerline, which may bias the human operators.

Three human operators were then asked to reposition the nodes of the perturbed tracing manually, to coincide with their own best estimate of the centerline of the dendritic segment. This was done using a graphical user interface that allows drag and drop operations of the nodes overlaid on the data at arbitrary zoom levels. Each operator repeated the process of node repositioning three times for a total of nine *manually positioned tracings*. A consensus tracing, or gold standard, was computed by averaging all the manually adjusted positions of each node.

We measured the deviation between two tracings as the average distance of all nodes in the first tracing to the centerline defined by the second tracing. For this test, the distance was measured in 2D,

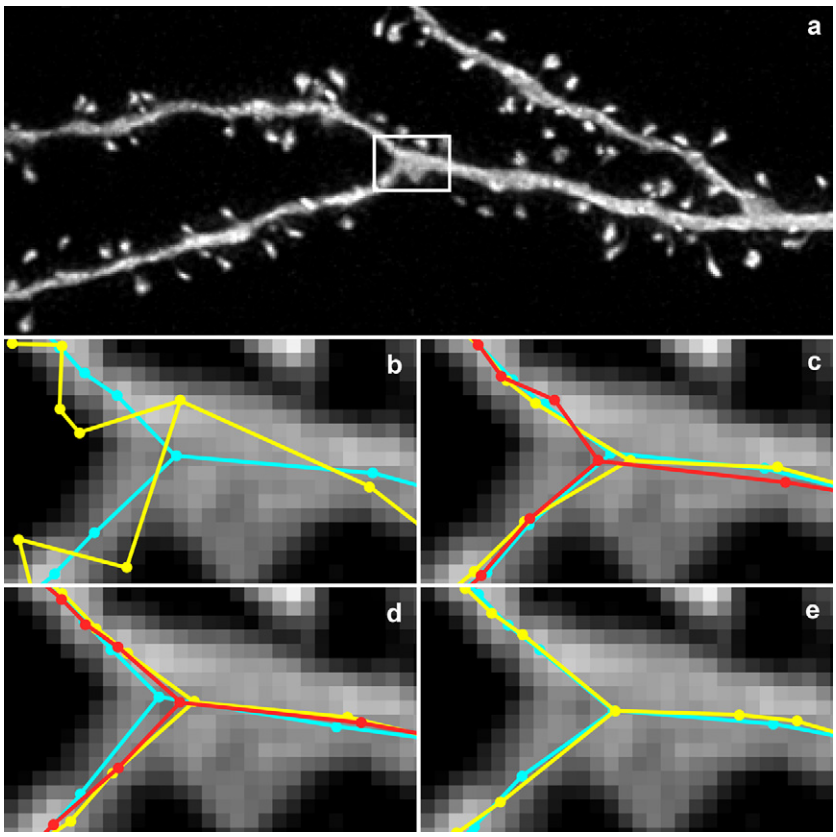

Fig. 3. Validation of the centerline produced by the automated tracing algorithm. (a) Maximal projection of the validation dataset with overlaid rectangle showing the zoomed-in section in panels b, c, d, and e. (b) Detail showing centerline produced by the algorithm (blue) and same centerline after each node was randomly perturbed (yellow). (c) Between-operator variability: comparison of centerlines produced by manual adjustment of perturbed centerline by three different operators (red, yellow and blue traces). (d) Within-operator variability: comparison of three centerlines produced by manual adjustment of perturbed centerline by a single operator. (e) Comparison of typical centerline produced by the algorithm (yellow) and the consensus ('gold standard') centerline of the three human operators (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

and operators where only required to adjust the *x* and *y* position of each node, thus avoiding the more subjective adjustment along the optical axis where resolution is much lower.

The following table shows the average deviation obtained while comparing the tracings of same operator (intra-operator deviation), between operators (inter-operator deviation), automated method to gold standard, and between automated tracings. Note that there exists some variation between different tracings of the automated method due to changes in seed positioning and to the use of scattered random sampling in threshold determination (see Section 2.8 for details).

Comparison type	Number of combinations evaluated	Average deviation (μm)
Within-operator	9	0.026902
Between operators	27	0.029777
Automated to gold standard	3	0.022322
Between automated tracings	3	0.015524

These numbers show that the deviation among tracings by different operators and among tracings by the same operator are both larger than the deviations obtained when comparing the automated tracing to the gold standard or among automated tracings. See also Fig. 3; panels c, d, and e; for visual comparison.

3.2. Performance

As a representative example of performance on multi-gigabyte datasets, we tested the speed of execution using a MS Windows XP Professional x64 Workstation with an Intel Xeon 5160 CPU running at 3.0 GHz and having 16 GB of RAM. The test dataset (Fig. 4) was approximately 10.3 GB in size and was comprised of a number of 3D confocal laser scanning microscopy image stacks tiled together and containing one labeled neuron. The total running time to trace this high-resolution dendritic arbor was approximately 32.1 s, with a total number of processed voxels equal to approximately 8.17 million voxels.

To evaluate performance on moderate size datasets (less than 1 GB), we tested the speed of execution using a MS Windows XP laptop with a 1.2 GHz Intel Pentium M CPU with 1 GB of RAM. This test dataset (not shown) was approximately 760 MB in size and was also comprised of a number of stacks containing one labeled neuron. The total running time to trace the dendritic arbor was 11.8 s, with a total number of processed voxels equal to approximately 1.43 million voxels.

4. Discussion

Structural changes of neurons in the brain in neuropsychiatric illnesses are complex and remain poorly understood. Neurons exhibit significant homeostatic control of essential brain functions including synaptic excitability, gene expression and metabolic regulation. As a direct consequence, structural alterations that neurons undergo during aging and in disease states have direct effects on electrophysiological properties and cognition. During aging it is evident that neurons undergo morphological changes such as a reduction in the complexity of dendrite arborization and dendritic length (Duan et al., 2003; Kabaso et al., 2009). Spine numbers are also decreased, and because spines are the major sites

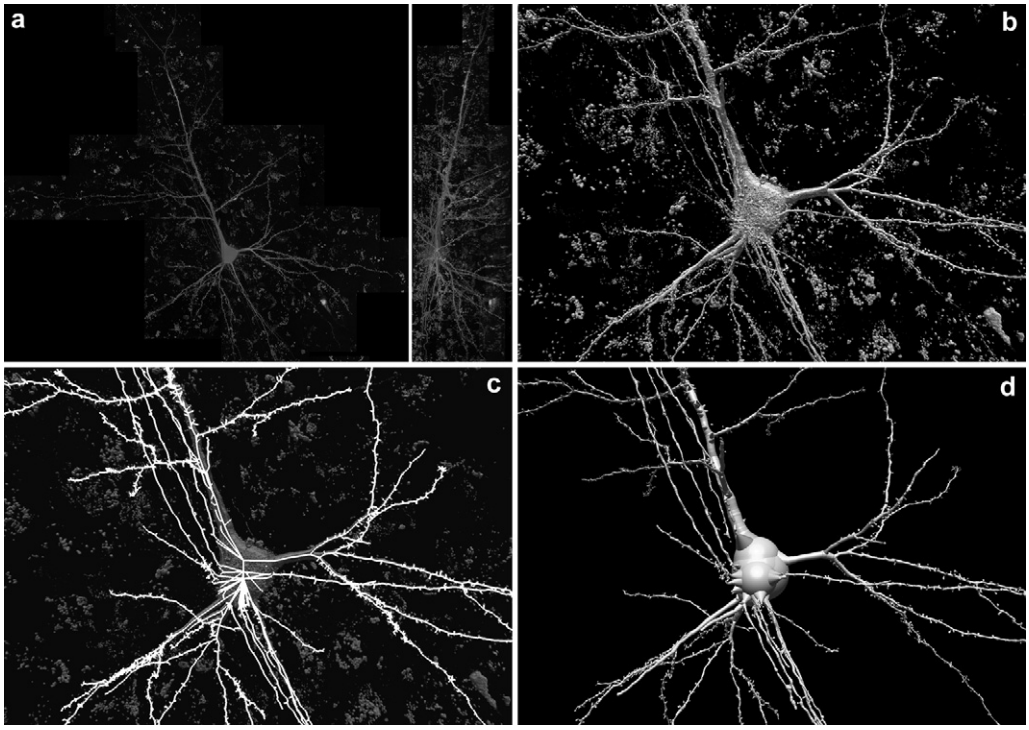

Fig. 4. Sample 3D neuronal dataset comprising a montage of confocal laser-scanning microscopy image stacks of an intracellularly labeled macaque monkey neocortical pyramidal neuron and resulting centerline and model. (a) Maximal projections of the dataset along the optical axis (left), and along the *x*-axis (right). (b) Closeup view of volume-rendered dataset as seen along the optical axis. (c) The final unfiltered centerline is shown as white lines within the volume-rendered dataset. (d) A complete model of the neuron is obtained by assigning diameters at each node along the centerline.

for excitatory synapses, changes in their numbers could reflect a change in synaptic densities, and a decrease in the frequency of spontaneous ionotropic glutamate receptor-mediated excitatory responses as well as a decrease in their levels of expression (Luebke et al., 2004; Dickstein et al., 2007). Altogether, these observations suggest that neuronal dysfunction resulting from either the aging process or from overt pathologies, which in turn underlies decline in cognitive function or behavioral disturbances, likely involves many subtle changes within the cerebral cortex that could include alterations in receptors, changes in the shapes of dendrites and spines, myelin dystrophy as well as alterations in synaptic transmission. Such changes in the cortical structure need to be appreciated in realistic 3D, an essential step for testing mechanistic hypotheses regarding the role of altered morphology in age- and disease-related functional and cognitive decline (Duan et al., 2003; Dickstein et al., 2007; Rocher et al., 2008; Kabaso et al., 2009).

In this context we have developed and used voxel scooping for automated centerline extraction in dendritic sections prior to morphologic spine type analysis (Radley et al., 2008; Rodriguez et al., 2008), as well as for high-resolution 3D reconstruction of entire neurons for compartment modeling (Weaver and Wearne, 2008). The method we present here is fast and easy to implement. It is well suited for extracting models of neuronal morphology from high-resolution datasets at interactive rates, which is crucial for obtaining complete and accurate morphologic models. The method emphasizes speed of execution and accuracy of reconstruction for medium to high-resolution confocal and multiphoton laser-scanning microscopy images of labeled neurons. Because the method operates on binarized data, binarization errors can be a significant issue for low-resolution or noisy data. For example, non-smooth boundaries can lead to the creation of spurious short branches by the tracing algorithm. In addition, adjacent structures may appear connected causing the algorithm to introduce topological errors. While short branches can be removed automatically (see Section 2.5), topological errors must be corrected manually using the editing facilities included in our program. For more accurate results on these types of data, we recommend the use of deconvolution prior to tracing. While a human operator tends to position branchpoint nodes by extrapolating the direction of the branches towards the junction, the algorithm tends to place the nodes a bit more distally within the branch. To accommodate those users who prefer the manual-like position of the junctions, we included an optional post-processing step that realigns the junction nodes to emulate the manual style. In addition, it should be noted that the exact placement of nodes around branchpoints is affected by the direction of the tracing and therefore we recommend that, for consistency, the seed always be placed as upstream as possible along the dendrite or in the soma if it is part of the dataset. The resulting model is amenable to manual editing and is compatible with compartment modeling packages such as NEURON (Hines and Carnevale, 2001) and GENESIS (Bower and Beeman, 1998), and with morphometry analysis software such as L-Measure (Scorcioni and Ascoli, 2001). The method is currently being used as the primary means of automated reconstruction in our freely distributed neuronal tracing application NeuronStudio, which can be obtained from our website at http://www.mssm.edu/cnic/tools.html.

Acknowledgments

The authors wish to thank Drs. Dara L. Dickstein, Farid Hamzei-Sichani, Jennifer I. Luebke and Anne B. Rocher for their valuable help and discussion. This work was supported by National Institutes of Health grants MH071818, DC05669, and MH58911.

References

Al-Kofahi KA, Lasek S, Szarowski DH, Pace CJ, Nagy G, Turner JN, et al. Rapid automated three-dimensional tracing of neurons from confocal image

stacks. IEEE Transactions on Information Technology in Biomedicine 2002;6: 171–87.
Al-Kofahi KA, Can A, Lasek S, Szarowski DH, Dowell-Mesfin N, Shain W, et al. Median-based robust algorithms for tracing neurons from noisy confocal microscope images. IEEE Transactions on Information Technology in Biomedicine 2003;7:302–17.
Al-Kofahi Y, Dowell-Mesfin N, Pace C, Shain W, Turner JN, Roysam B. Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images. Cytometry Part A 2008;73:36–43.
Blum H. A transformation for extracting new descriptors of shape. In: Wathen-Dunn W, editor. Models for the perception of speech and visual forms. Cambridge: MIT Press; 1967. p. 362–80.
Borgefors G, Ingela Nyström I, Sanniti di Baja G. Computing skeletons in three dimensions. Pattern Recognition 1999;32:1225–36.
Bower JM, Beeman D. The book of GENESIS: exploring realistic neural models with the GEneral Neural SImulation System. New York: TELOS/Springer-Verlag; 1998.
Dickstein DL, Kabaso D, Rocher AB, Luebke JI, Wearne SL, Hof PR. Changes in the structural complexity of the aged brain. Aging Cell 2007;6:275–84.
Duan H, Wearne SL, Morrison JH, Hof PR. Quantitative analysis of the dendritic morphology of corticocortical projection neurons in the macaque monkey association cortex. Neuroscience 2002;114:349–59.
Duan H, Wearne SL, Rocher AB, Macedo A, Morrison JH, Hof PR. Age-related morphologic alterations in dendrites and spine densities of corticocortically projecting neurons in macaque monkeys. Cerebral Cortex 2003;13:950–61.
He W, Hamilton TA, Cohen AR, Holmes TJ, Pace C, Szarowski DH, et al. Automated three-dimensional tracing of neurons in confocal and brightfield images. Microscopy and Microanalysis 2003;9:296–310.
Hines ML, Carnevale NT. NEURON: a tool for neuroscientists. Neuroscientist 2001;7:123–35.
Koh IY, Lindquist WB, Zito K, Nimchinsky EA, Svoboda K. An image analysis algorithm for dendritic spines. Neural Computation 2002;14:1283–310.
Kabaso D, Coskren PJ, Henry BI, Hof PR, Wearne SL. The electrotonic structure of pyramidal neurons contributing to prefrontal cortical circuits in macaque monkeys is significantly altered in aging. Cerebral Cortex 2009., doi:10.1093/cercor/bhn242.
Lee T-C, Kashyap RL, Chu C-N. Building skeleton models via 3-D medial surface/axis thinning algorithms. CVGIP: graph. Models Image Processing 1994;56:462–78.
Luebke JI, Chang Y-M, Moore TL, Rosene DL. Normal aging results in decreased synaptic excitation and increased synaptic inhibition of layer 2/3 pyramidal cells in the monkey prefrontal cortex. Neuroscience 2004;125:277–88.
Myatt DR, Nasuto SJ, Maybank SJ. Towards the automatic reconstruction of dendritic trees using particle filters. In: Nonlinear Statistical Signal Processing Workshop 2006; 2006. p. 193–6.
Pham DL, Xu C, Prince JL. Current methods in medical image segmentation. Annual Review of Biomedical Engineering 2000;2:315–37.
Radley JJ, Rocher AB, Rodriguez A, Ehlenberger DB, Damman M, McEwen BS, et al. Repeated stress alters dendritic spine morphology in the rat medial prefrontal cortex. The Journal of Comparative Neurology 2008;507: 1141–50.
Ridler TW, Calvard S. Picture thresholding using an iterative selection method. IEEE Transaction on Systems, Man, and Cybernetics 1978;8:630–2.
Rocher AB, Kinson MS, Luebke JI. Significant structural but not physiological changes in cortical neurons of 12-month-old Tg2576 mice. Neurobiology of Disease 2008;32:309–18.
Rodriguez A, Ehlenberger D, Kelliher KT, Einstein M, Henderson SC, Morrison JH, et al. Automated reconstruction of three-dimensional neuronal morphology from laser scanning microscopy images. Methods 2003;30:94–105.
Rodriguez A, Ehlenberger DB, Hof PR, Wearne SL. Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images. Nature Protocols 2006;1:2152–61.
Rodriguez A, Ehlenberger DB, Dickstein DL, Hof PR, Wearne SL. Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. PLoS ONE 2008;3(4):e1997, doi:10.1371/journal.pone.0001997.
Santamaría-Pang A, Colbert CM, Saggau P, Kakadiaris IA. Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention, vol. 4792; 2007. p. 486–94.
Schmitt S, Evers JF, Duch C, Scholz M, Obermayer K. New methods for the computer-assisted 3-D reconstruction of neurons from confocal image stacks. Neuroimage 2004;23:1283–98.
Scorcioni R, Ascoli GA. Algorithmic extraction of morphological statistics from electronic archives of neuroanatomy. In: Mira J, Prieto A, editors. Lecture notes in computer science. Berlin: Springer-Verlag; 2001. p. 30–7.
Storn R, Price K. Differential evolution—a simple evolution strategy for fast optimization. Dr Dobb's Journal 1997;22:18–24, and 78.
Streekstra GJ, van Pelt J. Analysis of tubular structures in three-dimensional confocal images. Network 2002;13:381–95.
Wearne SL, Rodriguez A, Ehlenberger DB, Rocher AB, Henderson SC, Hof PR. New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. Neuroscience 2005;136:661–80.
Weaver CM, Wearne SL. Neuronal firing sensitivity to morphologic and active membrane parameters. PLoS Computational Biology 2008;4(1):e11, doi:10.1371/journal.pcbi.0040011.