

Automated Reconstruction of Neuronal Morphology Based on Local Geometrical and Global Structural Models

Ting Zhao · Jun Xie · Fernando Amat · Nathan Clack · Parvez Ahammad · Hanchuan Peng · Fuhui Long · Eugene Myers

Published online: 6 May 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Digital reconstruction of neurons from microscope images is an important and challenging problem in neuroscience. In this paper, we propose a model-based method to tackle this problem. We first formulate a model structure, then develop an algorithm for computing it by carefully taking into account morphological characteristics of neurons, as well as the image properties under typical imaging protocols. The method has been tested on the data sets used in the DIADEM competition and produced promising results for four out of the five data sets.

Keywords DIADEM · Neuron tracing · Tube models · Tree structure reconstruction · 3D microscopy

Introduction

Progress in microscopy has generated a need for automated computational analysis and research into methods for such analyses has been termed bioimage

informatics, e.g. Peng (2008). In this paper we address a specific problem in this field, that of reconstructing neuronal morphologies from 3D microscopy images. It can be stated as follows:

Given a 3D image that is acquired via a common 3D microscopy protocol, automatically reconstruct the morphology of the neurons present in the image.

This specific problem is important because a good solution will have a significant impact on neuroscience. The ultimate goal of neuroscience is to understand how nervous systems work. This cannot be achieved without obtaining the structure of a real neuronal network, which reveals how neurons are connected. This in turn requires the extraction of morphologies of individual neurons in the system. However, biologists have to rely on time-consuming manual or semi-manual methods to turn the images into morphological models. Given the number of neurons in a nervous system, we can easily foresee the emergence of a bottleneck. Therefore, an accurate fully-automated reconstruction method is critical to the advancement of neuroscience.

The importance of the problem has led to quite a few attempts at developing automated methods for neuron reconstruction (Meijering 2010). To understand these methods, we need to remember two important common properties of the morphology of a neuron:

1. The fibers of a neuron form a tree.
2. At the typical resolution of a light microscope, a fiber of a neuron often has a smooth tubular shape.

T. Zhao (✉)
Qiushi Academy for Advanced Studies, Zhejiang University,
38 ZheDa Road, Hangzhou, 310027, China
e-mail: tingzhao@gmail.com

J. Xie · F. Amat · N. Clack · P. Ahammad · H. Peng ·
F. Long · E. Myers
HHMI Janelia Farm Research Campus, 19700 Helix Dr.,
Ashburn, VA 20147, USA

F. Long
e-mail: longf@janelia.hhmi.org

The most successful automated methods take advantages of one or both of these two properties. They can be further classified into two main categories:

Local Tracing A method in this category usually starts from a seed, which can be located automatically or manually, and traces where a neuronal fiber goes in the given image. The direction of tracing is determined based on the signal around the current location. A representative of this class is the method that was proposed in Al-Kofahi et al. (2002). The authors used template fitting to determine the direction of tracing. The template they used consists of four parallel edge detectors, each of which locates a part of the branch boundary with a greedy search. A local tracing method produces one branch at a time and does not directly trace through branch points where a fiber forks into two or more continuing fibers. So a separate branch point detection method is required to complete the reconstruction (Al-Kofahi et al. 2007). Model-free strategies of local tracing, such as Rayburst Sampling (Rodriguez et al. 2006) and Voxel Scooping (Rodriguez et al. 2009), can trace multiple branches with one seed. However, the quality of the tracing relies heavily on the existence of a preprocessing step that accurately separates foreground and background.

Global Skeletonization In contrast to local tracing methods, the other category of methods extract skeletons of neurons based on the global signal distribution in an image. One straightforward way of doing this is to use a selected segmentation algorithm to turn the image into a binary foreground/background form, and then apply a 3D binary skeletonization algorithm (Dima et al. 2002; Weaver et al. 2004; Evers et al. 2005; Narro et al. 2007). But this has not always been a good choice because of the difficulty of the first step, binarization. A Hessian-based vesselness measurement has been used to enhance line structures in images and improve skeletonization (Abdul-Karim et al. 2005; Yuan et al. 2009; Vasilkoski and Stepanyants 2009). But it requires predefined scales and is computationally expensive when the number of scales is large. In an alternate strategy, Dijkstra's shortest path algorithm has been adopted to extract skeletons directly from gray-scale images (Zhang et al. 2007; Peng et al. 2010b). This algorithm was applied to find the path that gives the shortest geodesic distance between two points. As long as the geodesic distance is reasonably defined, the path found is indeed the one desired. Since Dijkstra's algorithm provides the global optimal solution, the method is robust when the seed points are well located (Meijering et al. 2004; Xie et al. 2010; Peng et al.

2010a). But such a method is not reliable when there are multiple neurons in an image and it requires manual selection of the termini to trace between.

Both types of methods have been shown to be useful for neuron tracing. However, these automated results still require significant human curation and correction in order to be perfect. In this paper our goal is to develop a better method for reconstructing neurons whose morphology adheres to the two common properties mentioned above. For tracing neurons that do not show smooth neurites in the images (e.g. neurons imaged by electron microscopy), we would expect very different methods to work and it is beyond the scope of this paper. So we employ the idea of local tracing, and then refine the result with the use of shortest paths. Especially, we formulate the procedure of model based tracing, in the spirit of Al-Kofahi et al. (2002) but by realizing it with a formal tube model, improve performance significantly. A general framework of tree-like morphology reconstruction from the resulting fiber segments is also proposed. Based on the framework, we have developed an automated pipeline to produce a final reconstruction from a raw or preprocessed image. The method was tested on the DIADDEM data sets (Brown et al. 2011). The main purpose of this paper is to describe the method that brought us to the DIADDEM final, so we do not formally evaluate it against the many algorithmic ideas that have been explored previously.

Methods

Model-based Neurite Fiber Tracing in 3D

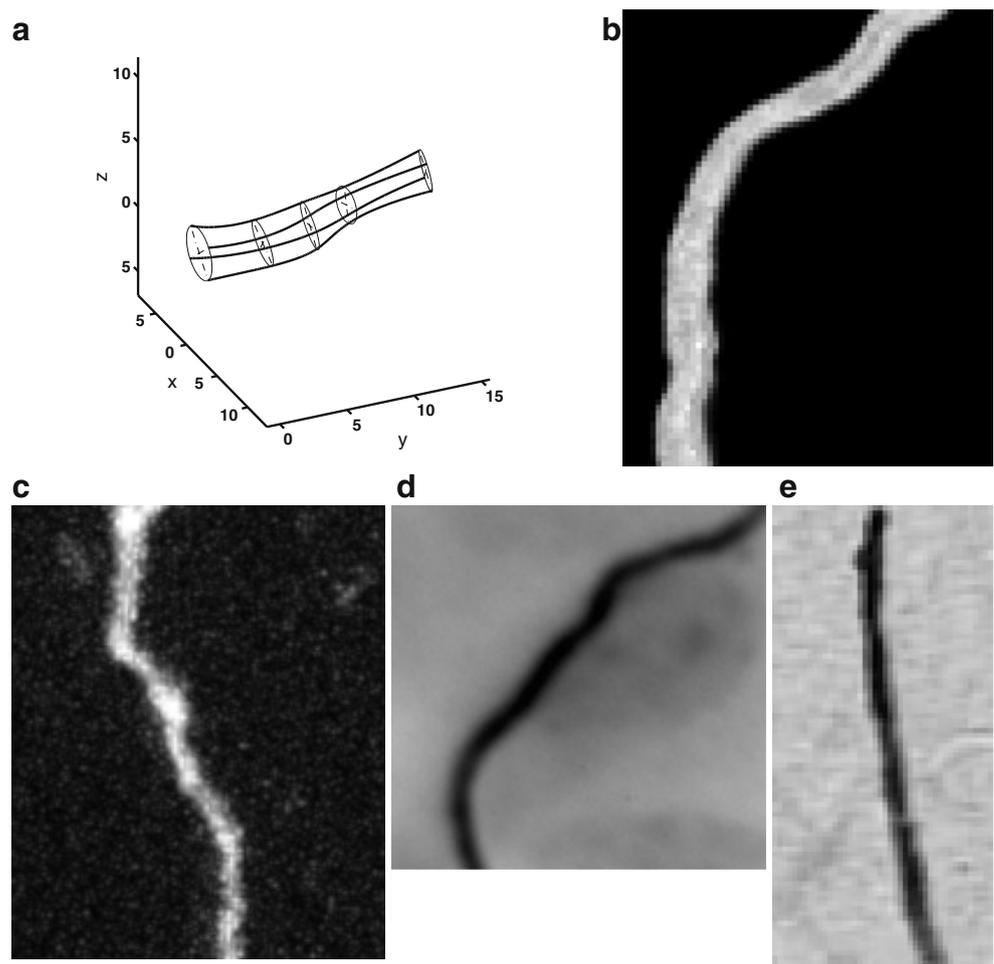
The shape of a neurite fiber can be approximated as a series of circular cross sections along a continuous curve (Fig. 1) that are deformed to an ellipse along the axial direction by the anisotropy of the point spread function (PSF) of the microscope. More formally, its surface $N(t, u)$ for parameters $t, u \in [0, 1]$ can be defined as:

$$N(t, u) = C(t) + O(u, t)\mathbf{R}(t)\mathbf{A} \quad (1)$$

where $C(t) = (x(t), y(t), z(t))$ is a continuous and differentiable curve, $O(u, t) = (r(t) \cos 2\pi u, r(t) \sin 2\pi u, 0)$ is a circle of radius $r(t)$ in the $z = 0$ plane,

$$\mathbf{R}(t) = \begin{pmatrix} \mathbf{r}_y \times \mathbf{r}_z \\ \mathbf{r}_y = \mathbf{k} \times \mathbf{r}_z \\ \mathbf{r}_z = \frac{\Delta C(t)}{\|\Delta C(t)\|} \end{pmatrix}$$

Fig. 1 Neurite model and example. **a** is a cartoon to show that the model is composed of a set of cross sections along a principle axis. **b–e** are neurites from the real images of the 4 DIADDEM datasets. **b, c** are examples of darkfield microscopy and **d, e** are examples of brightfield microscopy. Refer to the “Results” section for more details about the imaging conditions



is a fixed matrix at each value of t that maps the circle from the $z = 0$ plane into a plane perpendicular to the tangent of the curve C ,¹ and

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a(t) \end{pmatrix}$$

stretches or squeezes the z -dimension by the parameter $a(t) > 0$ reflecting the asymmetry of the PSF and the sampling rate.

When a neurite is smooth enough, which is often true in real data, we can approximate a neurite segment as a discrete series of elliptical cylinders. So one way to reliably detect if a point (x, y, z) is on the neurite’s principle axis $C(t)$, is to see if there is a good fit between

the signal and a cylindrical model centered at (x, y, z) of fixed height h reflecting the “mesh size” of the proposed series. Such an elliptic cylinder has four free parameters: its radius r , two Euler angles ϕ and ψ that orient it along a vector in space, and the anisotropy calibration a in the axial direction. This also defines a deformable model of a canonical cylinder, which has unit radius and orientation parallel to the axial or z -axis. Therefore designing a more reasonable filter-like structure for the canonical cylinder will allow us to estimate the parameters more efficiently and accurately.

A 3D Cylinder Filter

To evaluate the fit of a cylinder model to the data, we derived a 3D filter for a cylinder, which we also call a template. The fitting score between the model and the data is defined as the the correlation between the intensity data and the template. Due to its averaging effect, this simple definition can effectively suppress the disturbance of imaging noise or artifacts, even

¹The product for \mathbf{r}_y can give a singular 0-vector when $r_z = k$. In this case, its value is the limit of the expression as the singularity is reached.

if our modeling procedure does not consider imaging noise or artifacts explicitly. Our template is designed for darkfield images, where signal is brighter than background. One can simply invert the template for brightfield images.

In cross-section, a neurite fiber looks like a Gaussian-diffused spot possibly scaled along the axial direction, so we use an elliptical Mexican Hat filter (Laplacian of a Gaussian) to convolve with the signal along the symmetric axis of the cylinder. The choice of using a Mexican Hat filter is similar to the one in Schmitt et al. (2004), except that the shape of our filter can be elliptical. So our canonical unit vector is given by:

$$U(x, y, z) = (1 - (x^2 + y^2)) e^{-(x^2 + y^2)} \quad (2)$$

where $z \in [-h/2, h/2]$. To obtain the space of all possible cylinder templates, one can scale U in x and y by radius r , then rotate it in any manner desired, and final scale it along z (making it elliptical) by factor a .

Of course, the support of the filter must be finite not only for computational purposes, but also because, while the model demands that background surround (most) of the neurite fiber, making this zone too large means that fibers passing very near by will disrupt the fit of the filter. So for a given filter instance, regardless of a and r , we limit the support of the filter to τ pixels beyond the zero-crossing of the Mexican Hat. The second consideration, is that for very thick neurons, their intensity profile often has a flattened top due to saturation of the signal in the center (e.g. Fig. 2). For such a profile a slight deformation of the template induces little or no change in score, so we empirically found that regularizing the template family by multiplying the positive part of the Mexican Hat by $(ar^2)^{1/4}$ solves the problem.

As long as the neurite fiber section under consideration is surrounded by background, the filter essentially

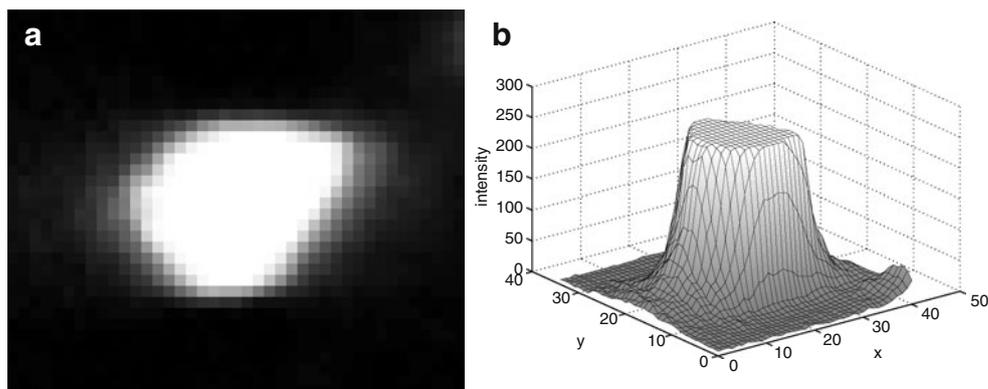
gives a large score for any cross section surrounding the bright region. To rectify this the cross correlation score of the filter convolved with the signal is normalized by dividing by the integral of the absolute value of the filter over its support. In the optimization step to follow, the r and a parameters are varied along with the orientation angles at a specified center point in space to those for which the template yields the largest normalized score when convolved with the signal.

Note that h and τ are the two parameters that need to be chosen prior to running our method for any particular image acquisition conditions and they are primarily a function of pixel size. For example, for most images at 40X $h = 10$ and $\tau = 2$ suffice. In our experience the method is robust to these parameter choices and they remain fixed for any set of images acquired with the same microscope and imaging protocol.

Tracing a Neurite Fiber

One could use a greedy search to optimize the four free parameters (two orientation parameters and two scale parameters) of the template when it is placed at a given point in space. But this is less appropriate for such a large number of parameters because search time increases exponentially in the number of parameters. If the score function is smooth and has a single peak, gradient descent is certainly preferred. Fortunately, our scoring function does have this property in a fairly large zone of the parameter space around the true optimum. Moreover, the symmetry of the function makes it well fit by a quadratic function within this zone, so as long as the template is placed in a reasonable initial position (which could for example be determined by a coarse discrete sampling of the parameter space), one can use the conjugate gradient descent method, which has been shown to be the best for optimizing general

Fig. 2 Example of a thick neuron cross-section where the intensity profile is flattened in the center. **a** is the actual image cross section. **b** is a plot of the intensity profile of the cross section in **a**



quadratic functions. In our implementation, we used the Polak–Ribière conjugate gradient descent method (Wright and Nocedal 2006). Results showed that it is significantly better than steepest gradient descent (e.g. Fig. 3).

There is no closed form expression for calculating the gradients because of the lack of a closed form describing the image. So numerical estimation is necessary, and as per standard practice we use the approximation $(S(u + \Delta u) - S(u - \Delta u))/(2\Delta u)$ for the partial derivative $\partial S/\partial u$ of a variable u .

After fitting a cylinder at a given position, we have only modeled a small section of the target neurite fiber. To cover the whole neurite, we use a strategy similar to the one introduced in Al-Kofahi et al. (2002) to traverse the entire fiber in mesh steps of size $h/2$. Once we have fitted a cylinder c_i , we then place a cylinder c_{i+1} with its center at one end of c_i 's symmetric axis in the same orientation and of the same cross-section shape, and then optimize the orientation and shape of c_{i+1} from this starting position. In this way, given an initial cylinder c_0 , one can walk in both direction to traverse a fiber. The pseudo-code for tracing in one direction is simply as follows.

```

i ← 0
repeat
  ci+1 ← Axial translation of ci by h/2 pixels
  Optimize ci+1
  i ← i + 1
until ci has a score below the threshold or ci hits a
previously traced region

```

In the next subsection, we discuss how to arrive at a set of seed cylinders c_0 for an entire image.

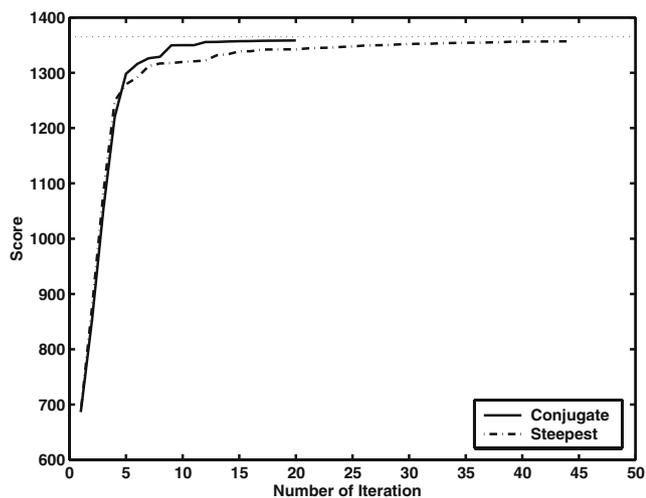


Fig. 3 Conjugate gradient descent reaches the optimal score faster than steepest gradient descent

Tracing All Neurite Fibers

In the previous section we showed how to trace a neurite fiber given an initial filter fit c_0 . To find a set of initial filters, one for each fiber, we developed a seed detection method to locate all locations and approximate orientations of points on the curve of a neurite based on two image features. The first feature is pixel intensity: when a pixel is brighter it is more likely on a neurite. The second feature is local geometry. We prefer a seed to be on a line structure. This can be measured by examining the eigenvalues of a Hessian matrix (Sato et al. 1998) computed at a point, typically at a number of scales. But the calculation of a Hessian matrix is expensive in both time and space, so we only do the calculation once at the smallest scale ($3 \times 3 \times 3$), and assume that thick fibers are bright enough that they will be picked up by the intensity feature. For each of these two features a threshold is computed with the triangle method over the histogram of the local maxima of feature scores at all pixels. Any pixel that has either of its two features above threshold is considered to be an on pixel in a binarization of the image that roughly covers all the neurites or at least a portion of every neurite fiber.

Since we want to place our initial filter fits c_0 at the center of a neurite, we consider as possible *seed* locations those points that are local maxima of the distance map of the binary image. Obviously, many possible seeds can be reported for a given neurite fiber whereas only one is needed. So we optimize the fit of a cylinder template at every possible seed. The optimization at each point is initialized with the best-scoring τ -pixel circular cylinder over a fixed set of orientation sampling the space of all possible orientations. We then sort the seeds in the order of their optimized model scores for c_0 . We start with the best seed and trace its neurite fiber, removing from consideration any seeds that are covered by the tracing. We then pick the next best remaining seed and trace it, and continue greedily in this fashion until all the seeds are exhausted. In this way we build a model of (almost) every neurite fiber in the image. The caveats are that if the signal is sufficiently dim in parts then the fiber may not be spanned leading to the occasional broken fiber, and in some cases the tracing moves directly through a branch point, but this is benign in that it is technically the fusion of two fibers that should ultimately be fused anyway.

Reconstruction of Neuronal Morphology

Once the set of individual neurite fibers or fragmentary trace have been obtained for an image, they can then

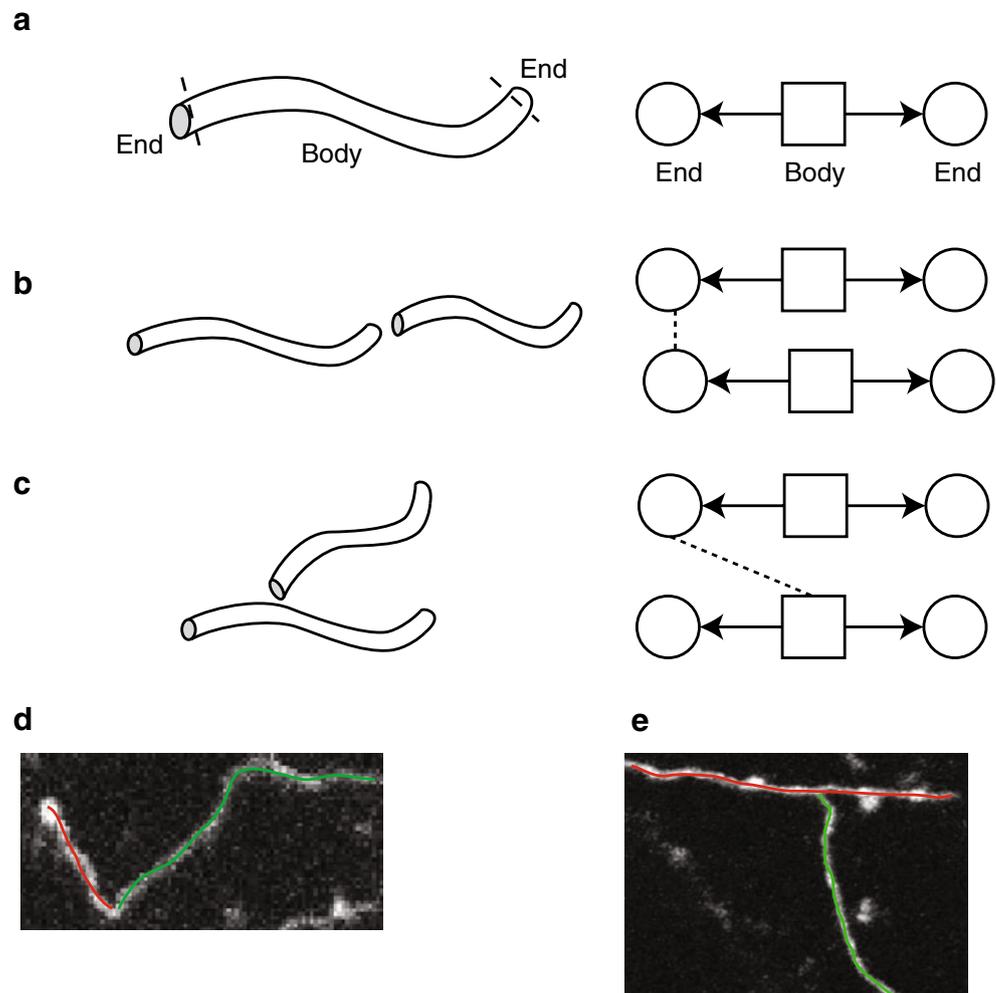
be assembled to form a neuronal tree structure. There are typically many ways to join them together, but by prohibiting the formation of cycles during the assembly process we guarantee that the final structure is a tree. This can be formulated as a graph problem.

Suppose we have a set of neurite fibers, $\{N_k\}_{k=1}^N$. The goal is to determine how they are connected. We first create a *neurite graph*, in which each node is a neurite and each edge indicates a possible connection between two neurites. In contrast to a usual graph, a node in a neurite graph has three parts, two ends and a body. So an edge between two nodes in this undirected graph must also specify at each node whether it connects to an end or the body. The connection pattern of an edge can be either end-to-end or end-to-body (Fig. 4), but never body-to-body. Moreover, only one edge is allowed between two neurites. Our problem, in the framework of this graph, is to find a minimum weight spanning tree in the case one neuron is under view and

all the N_k are true positives. The whole procedure is illustrated in Fig. 5. In our method, we do not consider the connection between a pair of neurites that are too far away from each other. The distance threshold is set to 20 pixels, which is twice as long as the height of the cylinder templates. Any two neurites that have a distance larger than the threshold are supposed to be from different neurons.

So the design issue is how to assign a cost to each possible edge, which will join fibers if selected to be in the result. The cost must be such that the smaller the value the more desirable the edge for inclusion. One of the simplest schemes for the cost of connecting N_i to N_j is the distance between the center of the end of N_i and the center of the end of N_j in the case of an end-to-end edge, and the distance of the center of the end of N_i to the nearest point on the surface of the body of N_j in the case of an end-to-body edge. This definition is quite natural because two fibers tend to be close

Fig. 4 Graph of neurites. A neurite is modeled as a 3-part node **a**, which can form end-to-end connection **b** or end-to-body connection **c** with another node. Real examples of end-to-end connection and end-to-body connection are shown in **d** and **e** respectively. Colors indicate different neurites



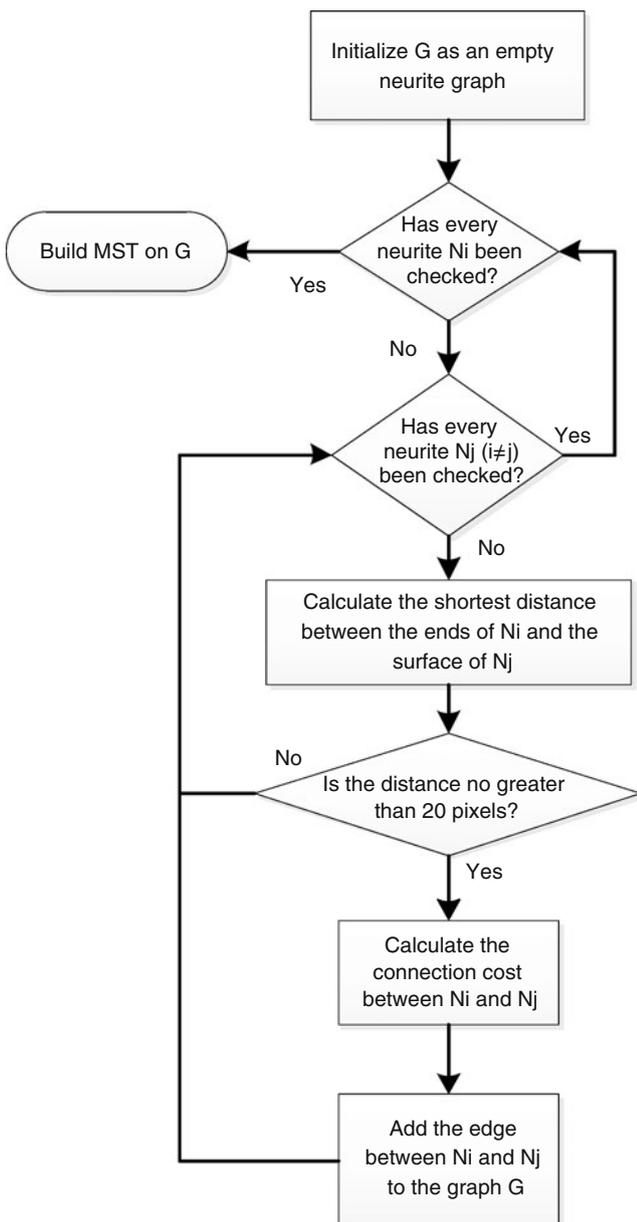


Fig. 5 Flowchart of the tree reconstruction procedure

to each other when they join to form a branch point in the image. However, when such gaps start being as large as the nearest fiber to one of the fibers under consideration, a wrong choice can be made. This does happen with some low frequency. For example, there was one such occurrence for a traced neuron which has about 40 branches. So other cues must be considered.

One such strong cue to tell if two tubes are truly connected is the strength of the image signal between them. If we can find a path from one neurite fiber end or body to the other endpoint along a path of bright pixels, then it is highly likely that the two tubes connect to each

other. Therefore we added another measurement, the geodesic distance, as an option. The geodesic distance between two points x_0 and x_1 is defined as

$$geo_g(x_0, x_1) = \min_{c \in \{\text{all possible paths}\}} \int_c g[I(c(t))] ||dc(t)|| \quad (3)$$

where $c(0) = x_0, c(1) = x_1, ||dc(t)|| = \sqrt{dc_x^2(t) + dc_y^2(t) + dc_z^2(t)}$ is the differential of the arc length of $c(t)$, and $I(x)$ is the image intensity at x . The function g defines how image intensity contributes to the geodesic distance. In our study, we expect the shortest path c , to be on the foreground. This can be expressed by a sigmoid function

$$g(x) = \frac{1}{1 + e^{-\frac{x-\alpha}{\beta}}} \quad (4)$$

where α and β are selected to reflect the separability between the foreground and background and we do not have to set them arbitrarily. For example, we can take the signal inside a tube as foreground and those around the tube as background and calculate their average values, which are denoted as c_f and c_b , respectively. If we want $g(c_f) = 0.99$ and $g(c_b) = 0.01$, it is easy to solve for α and β yielding $\alpha = \frac{c_f + c_b}{2}$ and $\beta = \frac{c_f - c_b}{2 \ln(99)}$.

A neurite graph becomes a typical weighted undirected graph if we ignore the internal structure of its nodes and the connection pattern of its edges. In some sense, the minimal spanning tree of this graph maximizes the likelihood of being the appropriate tree reconstruction if the cost function on edges properly

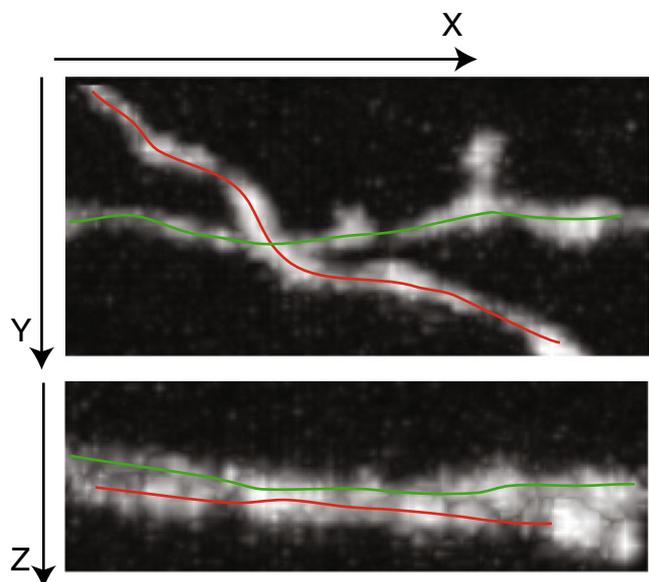
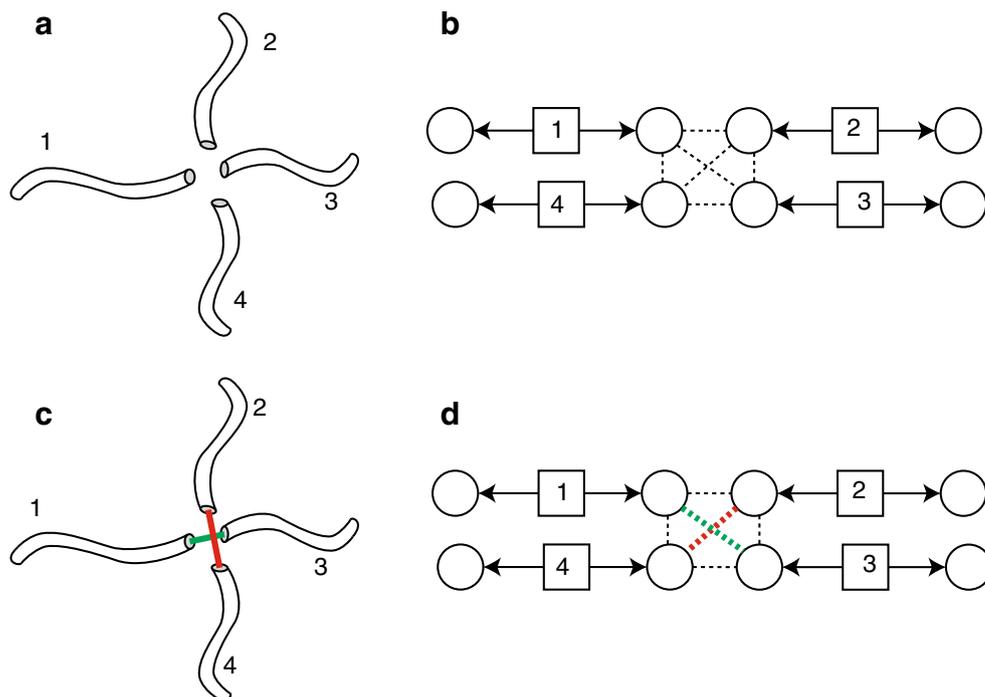


Fig. 6 An example of crossover from real data. The *top* image and the *bottom* image are showing the same two neurites projected onto X-Y plane and X-Z plane respectively. The *color lines* indicate how the two neurites cross

Fig. 7 Crossover **a** in a neurite graph **b**. The solution **c, d** gives the minimal sum of the angles between matched neurites. The *red and green lines* in **c** and **d** show how the neurites are merged after matching



reflects the likelihood that two neurites involved have a direct connection. Unfortunately there is the case that when two neuron fibers cross each other very closely, even the quite reasonable geodesic cost function fails to meet this criterion.

When one fiber passes another very closely, especially in the axial z -dimension where resolution is lower with most microscopes, their signal may overlap to the extent that the geodesic cost function (falsely) indicates that they should be joined (Fig. 6). In other words, the overlap pattern cannot be identified by the MST algorithm. We call such a pattern a *crossover*, and it must be distinguished from the real fusions that need to occur at branch points in a neuron. In the neurite graph, a crossover has one of two special signatures depending on whether the fiber tracing passes through the crossover region or not. If the tracing does not pass, the pattern will be pairwise end-to-end connections among four or more nodes (Fig. 7). Otherwise, the pattern will be the connections from the ends of two or more neurites to the body of another neurite (imagine fibers 1 and 3 are already joined into a single fiber in Fig. 7). For the latter case, the problem is compounded as the local tracing that went through the crossover region may have “jumped tracks” to another fiber. So conservatively, we break any fibers that pass through the connection region. This has the further advantage of reducing the problem to the first case where all possible connections are end-to-end.

Crossover cannot be solved directly with minimum spanning tree as it will tend to just join all the fibers together (in the case that multiple neurons are present), or join fibers depending on the signal strength between them which is not a good indicator of which fiber should be joined with which. A better indicator is the angle between two neurite end vectors: if there is no or little change in direction as one goes from one fiber to the next, then it is likely that the two fibers should be joined. So for the fibers that end at the crossover region we find the best set of pairings based on minimizing the total of the join angles using the Hungarian algorithm. These neurites are then fused. The cross-over process takes place prior to the greedy minimum spanning tree algorithm (MST).

Results

We tested the algorithm on 4 of the 5 data sets² from the Digital Reconstruction of Axonal and Dendritic

²The excluded dataset, Neuromuscular Projectio Fibers, contains neurons that have irregular surfaces and uneven internal stainings. Therefore we used a model-free region growing method to process the dataset instead.

Table 1 Summary of the datasets

Dataset	CF	HC	NL	OP
<i>Species</i>	Rat	Rat	Mouse	Drosophila
<i>Nervous system</i>	Cerebellar cortex	Hippocampus CA3	Peripheral	Olfactory bulb
<i>Fiber type</i>	Axon terminals	Dendrites and axons	Axons	Axons
<i>Microscopy</i>	Transmitted light brightfield	Transmitted light brightfield	2-photon microscopy	Confocal microscopy
<i>Voxel size</i>	$1 \times 1 \times 8.8 \mu\text{m}$	$0.217 \times 0.217 \times 0.33 \mu\text{m}$	$0.294 \times 0.294 \times 1 \mu\text{m}$	$0.33 \times 0.33 \times 1 \mu\text{m}$

Fig. 8 The tracing result (green) of an CF image stack overlaps with a slice of the original stack. The result is translated a little from its original position for better view

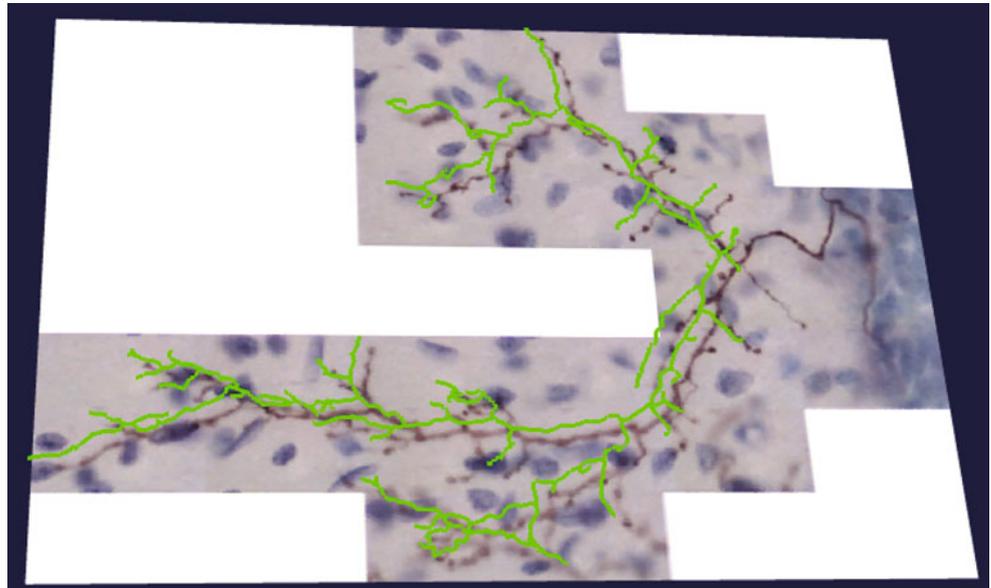


Fig. 9 The tracing result (red) of an HC image stack overlaps with a slice of the original stack. The result is translated a little from its original position for better view

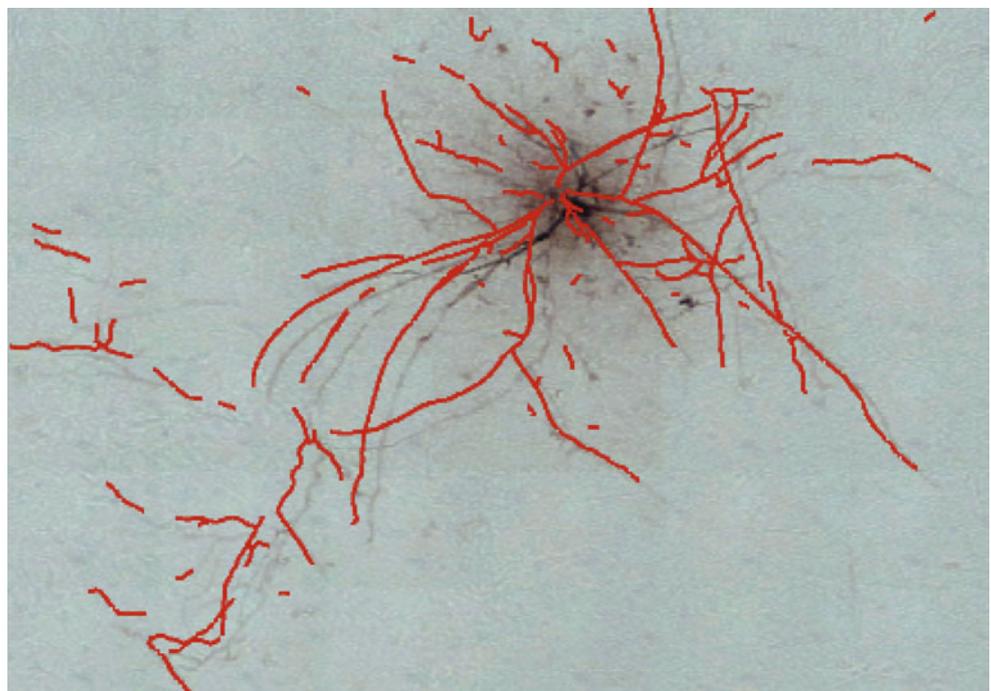
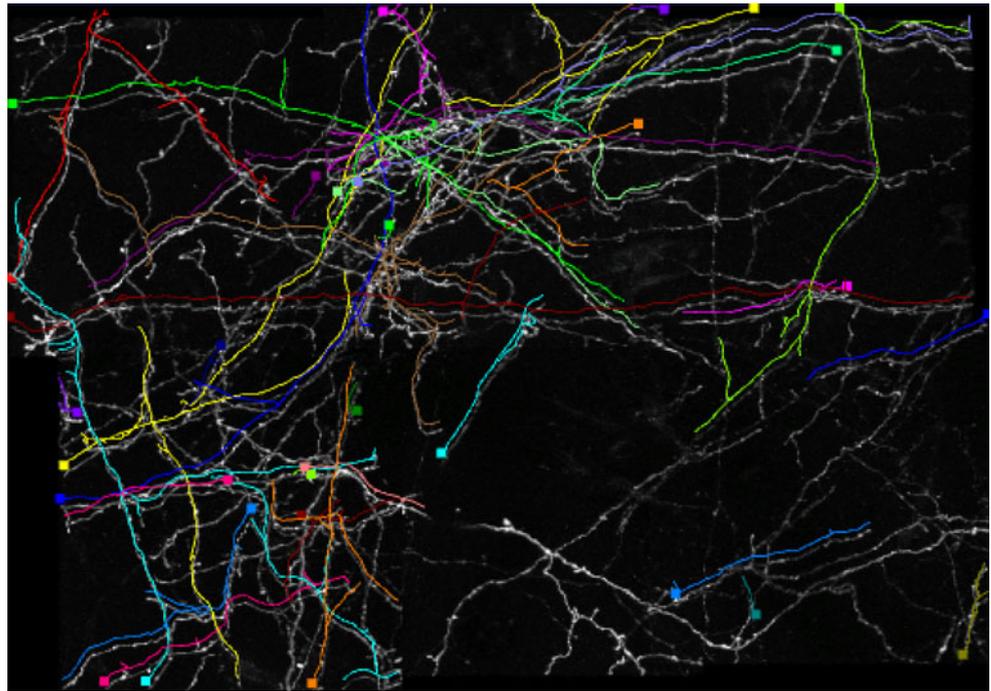


Fig. 10 The tracing result (*colored*) of an NL image stack overlaps with the volume rendering of the original stack. The result is translated a little from its original position for better view. Note that there is plenty of signal that has no corresponding neuronal structure. This is because the structures are excluded from the specified neuron set, despite that the tracing did not miss them



Morphology (DIADEM) competition (<http://www.diademchallenge.org>). Named after the stained neuron types, the 4 datasets are CF (Cerebellar Climbing Fibers) (Sugihara et al. 1999), HC (Hippocampal CA3 Interneuron) (Calixto et al. 2008), NL (Neocortical

Layer6 Axons) (De Paola et al. 2006) and OP (Olfactory Projection Fibers) (Jefferis et al. 2007). The imaging protocol for each dataset is listed briefly in Table 1. The images of the datasets CF and HC are from bright-field microscopy and some special preprocessing steps,

Fig. 11 The tracing result (*red*) of an OP image stack overlaps with the volume rendering of the original stack. The result is translated a little from its original position for better view



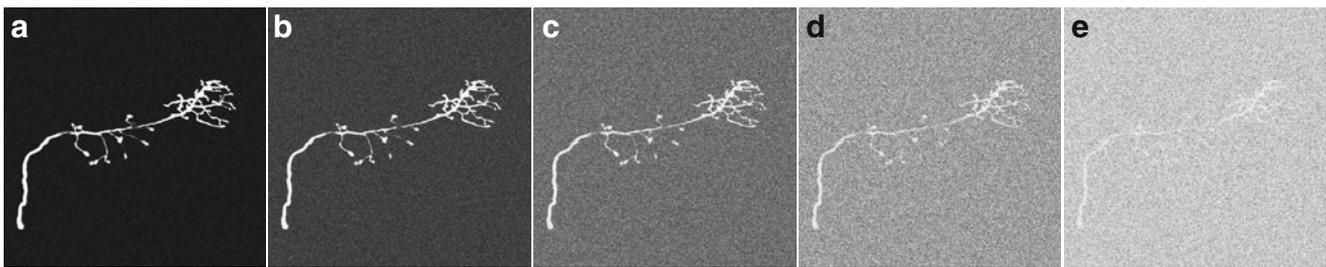


Fig. 12 An OP image stack was contaminated by different levels of Gaussian noise: **a** $\sigma = 20$, **b** $\sigma = 40$, **c** $\sigma = 60$, **d** $\sigma = 80$, **e** $\sigma = 100$

to be described, are necessary before applying model fitting on them. For the NL dataset, an image contains multiple neurons. Given the coordinates of a point on each neuron, we built a single tree first and then cut the edges with largest connection cost to get multiple trees. Also the crossover prior described in the previous section was applied to the NL data only. The final results on all 4 datasets are illustrated in Fig. 8 (CF), Fig. 9 (HC), Fig. 10 (NL), Fig. 11 (OP).

Testing for Robustness to Noise We also tested our method on different noise levels to show its strength. The testing images were created by adding Gaussian noise to the OP image stack (Fig. 11). We used five noise levels, measured by the standard deviation of the noise distribution: $\sigma = 20$, $\sigma = 40$, $\sigma = 60$, $\sigma = 80$, $\sigma = 100$ where values in the image range from 0 to

255. The neuron becomes less visible when σ is larger (Fig. 12), making tracing progressively more difficult. To quantitatively evaluate how robust our method is to noise, we calculated the scores of tracing quality using the DIADEM metric (<http://www.diademchallenge.org/metric.html>). The results were also compared with those obtained from the free software NeuroStudio (Wearne et al. 2005), which was developed based on the Rayburst Sampling tracing algorithm. As shown in Fig. 13, our method produced reasonable result (score = 0.866) even when the noise level is at 100. In contrast, NeuroStudio’s performance degrades rapidly reaching a score of 0 when the noise is at 80. Moreover, our method outperformed NeuroStudio significantly for all the noise levels, including the case where there is no noise.

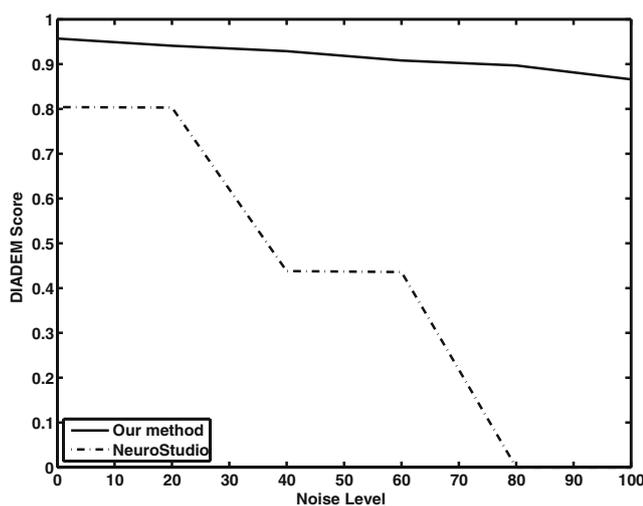


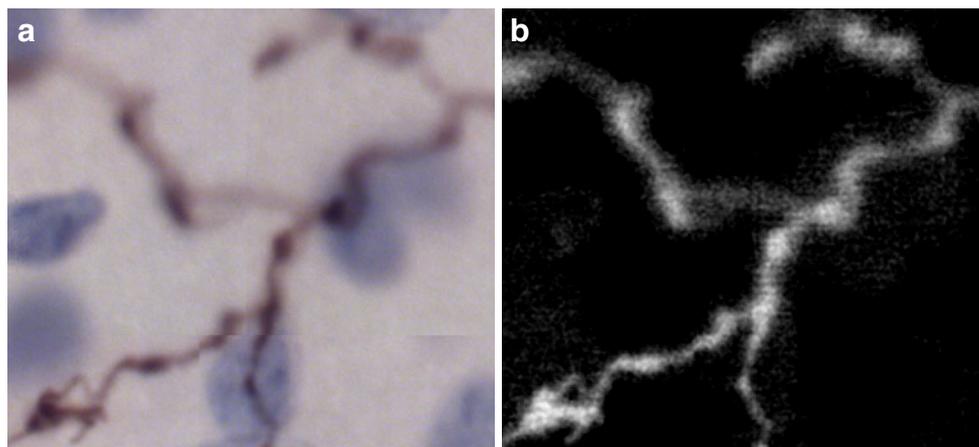
Fig. 13 The tracing results on different noise levels generated by our method (solid line) are compared to the results obtained from NeuroStudio (dash-dot line). It shows that our result is much more robust to the noise

Preprocessing for the CF Dataset In the CF images neurons are brown and nuclei are blue. In some areas they overlap due to the nature of the bright-field point spread function which does not optically section in z . This overlap can not be resolved by extracting a certain color component as in some regions light from both objects is present. We solved the problem based on a color mixing model. In this model, there is an average (bright) background B and there are two different materials, m_1 and m_2 , in the imaging field and each material absorbs light of a certain color, c_1 and c_2 , respectively. At any voxel x , the total observed intensity $I(x)$ is:

$$I(x) = B - [c_1, c_2] \cdot \begin{bmatrix} p_1(x) \\ p_2(x) \end{bmatrix} \tag{5}$$

where $p_i(x)$ denotes the absorption contribution from material m_i . We estimate B as the average intensity of each frame which is a good estimate as the background occupies most of each frame. c_1 and c_2 are calculated reliably by taking the average values of the two

Fig. 14 The color component corresponding to neurons **b** is extracted from an image of the CF dataset **a**



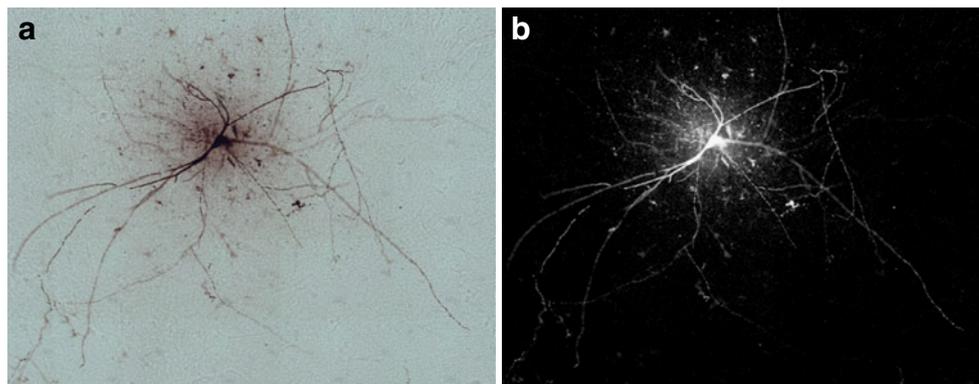
separable peaks in the color histogram of $B - I$. Once we have these three values, we compute the amount of each material at each voxel by solving Eq. 5 by linear regression. The result is shown in Fig. 14.

Preprocessing for the HC Dataset Much of the preprocessing required for this dataset was due to either poor microscopy or poor stitching of the multiple 3D stacks that made up each data set. In particular, the z -planes of a given tile were offset with respect to each other and the luminance of each plane varied significantly. We do not know how these artifacts were introduced, but we perforce had to correct them.

First the images were converted to gray-scale for the neuron intensity using the color model scheme used for CF, save that here matters were particular easy as there was only one material. Next the motion artifacts were corrected by splitting the full volume into a series of subvolumes corresponding to the lateral tiling used to originally acquire the data. Each subvolume

was then processed independently. For each pair of successive z -planes, a lateral translation was estimated as that maximizing the correlation between each edge-enhanced representations of the respective planes. Edge enhanced images were computed by filtering with a difference-of-boxes filter (box sizes 5 and 15 pixels) followed by masking with the original data binarized according to the Ostu threshold. Linear interpolation was used to align images according to the estimated translation. Border regions were filled by clamping to the edges. After motion correction, the volume was median filtered along z (window size $1 \times 1 \times 13$) in order to correct for intensity fluctuations between planes. The window size was chosen to approximate the observed extent of a single neurite image along the z direction. Finally, the vertical blur from strongly labeled objects, such as the cell soma, added an approximate constant signal to all planes above and below the object. To remove this signal, we estimated it as the median plane along z and subtracted that from each plane clamping

Fig. 15 Preprocessing result for the HC dataset: **a** Original image; **b** Processed image



negative values to zero. An example of processing result is shown in Fig. 15.

Discussion

There have been numerous articles on methods for the automated reconstruction of neuronal morphology. However, the quest for better, more powerful reconstruction algorithms remains. Existing methods fall into the two main classes of *local tracing* and *global skeletonization*. Both paradigms have proved to be useful, but we feel their potential has not been fully explored. In this paper, we refined a local tracing approach by designing a more sophisticated template for which we carefully optimize its orientation and positioning as the tracing progresses. Combining it with a novel neurite fiber graph model, we have developed a robust neuron reconstruction method. The method works on various types of images with few modifications. Even though the method was originally designed for darkfield images, it generates reliable results for pre-processed brightfield images as well.

We proposed a geometrical model to formulate the shape of a neurite fiber. From the model, we derived a tracing algorithm based on deformable templates. We also found that the template parameters can be estimated efficiently by conjugate gradient descent. As a result the method can deal with neurites with various sizes and produce accurate measurement of the surface of a neuron.

One main advantage of our model-based approach is its robustness to noise. This is due to the intrinsic averaging effect of the score calculation. So it does not need an additional noise reduction method to get acceptable results. This is especially well-suited for interactive tracing, which allows a user to extract a neurite with one click on raw data.

We have also defined a special graph model to derive algorithms for finding the most likely tree structure given a set of neurite fibers. The graph model is flexible and allowed us to add useful priors. For example, we have shown that adding a prior for crossover patterns can improve the reconstruction results of the NL data set dramatically.

The positive feature of our template model is that given sufficient support h it clearly distinguishes neurite from non-neurite artifacts. That is it has a low false positive rate of identification (an example is shown in Fig. 16). The problem is that it does not detect short features that are at fewer than h pixels in size such as short side branches, or sometimes it will erroneously fit the model over a small break that actually separates two

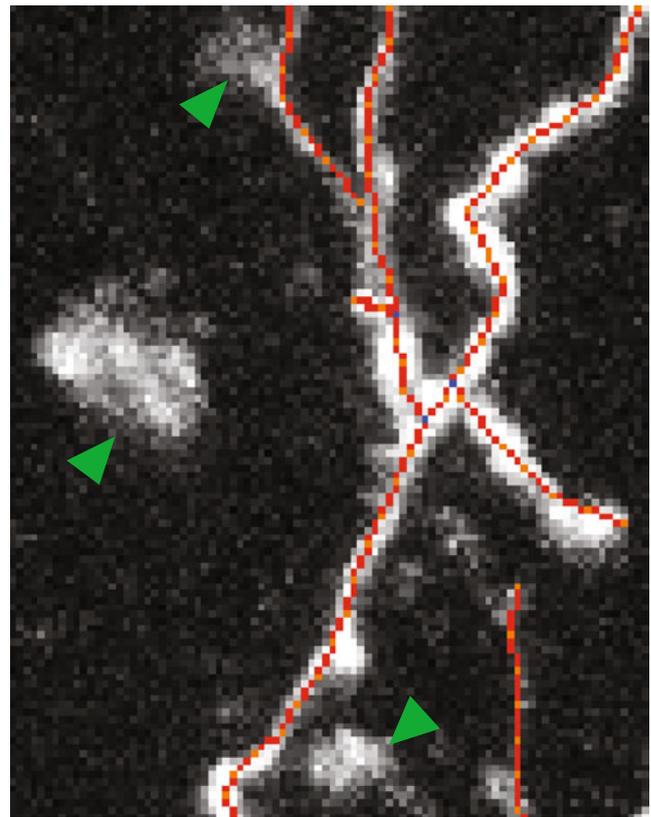


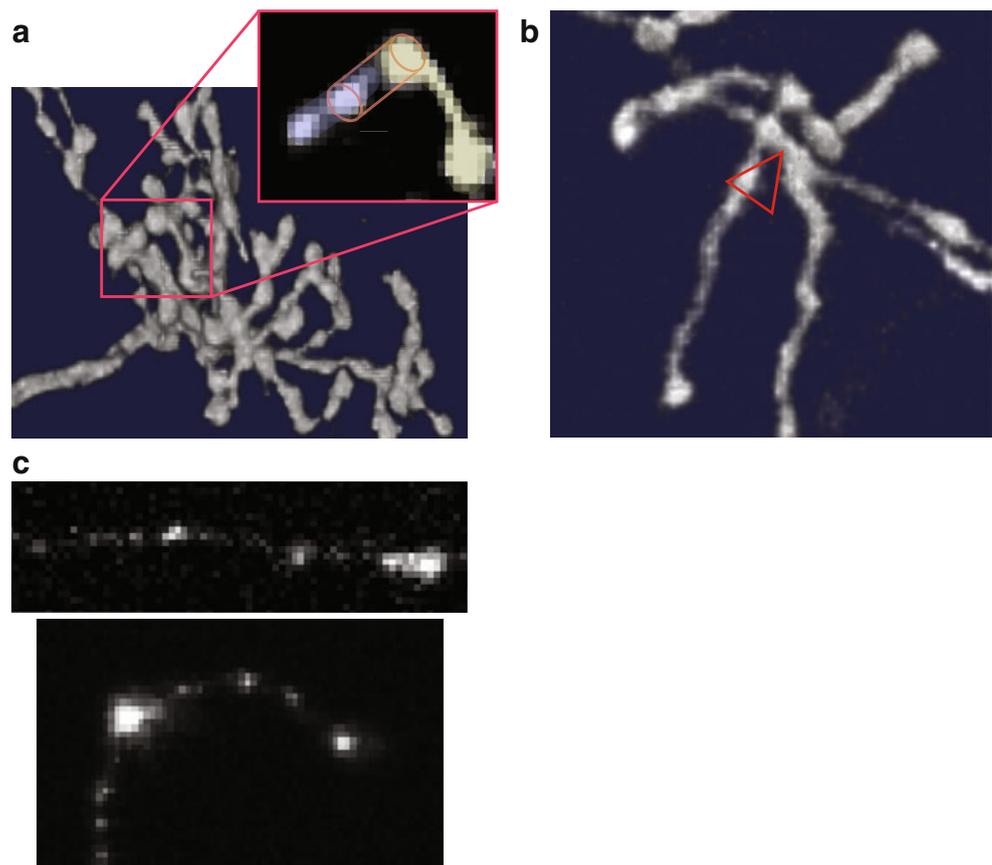
Fig. 16 Our method can avoid tracing blobs (marked by green arrows) that are as bright as some neurites

fibers. To further improve the method, its limitation in these circumstances must be addressed (Fig. 17) and we suggest approaches as follows:

Dense Branches When many branches are present in a small region, such as the end of an axonal projection, the fitting template may jump from one branch tip to another or fail to fit on a short segment between two branch points. Both mistakes will result in topological errors. The first type of mistake is hard to fix because of the loop structures formed in the image due to the limited resolution. We may have to correct them manually or add topological constraints for reconstruction. The second one may be less a problem because it is easier to detect the missing signal than to correct an overfitting. Once we find a bright region not being included in the reconstruction, we can extract it and estimate the orientation and size by principle component analysis if the cylinder model does not fit.

Broken Branch Signal Although a neurite is continuous, we may only see a sequence of isolated bright dots when it is imaged. This can be caused by uneven staining or uneven distribution of GFP particles and is generally indicative of poor sample preparation. Our

Fig. 17 Scenarios causing problems for our tracing method: **a** The two branch tips are so close that the tracing jumps from one to the other; **b** The cylinder model may fail to fit on a short branch between two close branch points; **c** Two examples of broken neurite signal on which the tracing will stop too early at any seeding point



tube model will generally fail on data of this quality because the assumption of continuity is no longer valid. One could treat this as a special case and use a different tracing method, such as the one proposed in Peng et al. (2010a) to attempt to span such dark zones.

Information Sharing Statement

All codes for our neuron tracing pipeline are available in open source and can be found on the DIADEM website (<http://diademchallenge.org>).

Acknowledgements The authors thank Shiv Vitaladevuni, Mark Longair, and Saket Navlakha for helpful discussions. This work was supported by Howard Hughes Medical Institute.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Abdul-Karim, M., Roysam, B., Dowell-Mesfin, N., Jeromin, A., Yuksel, M., & Kalyanaraman, S. (2005). Automatic selection of parameters for vessel/neurite segmentation algorithms. *IEEE Transactions on Image Processing*, 14(9), 1338–1350.
- Al-Kofahi, K., Lasek, S., Szarowski, D., Pace, C., Nagy, G., Turner, J., et al. (2002). Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Transactions on Information Technology in Biomedicine*, 6(2), 171–187.
- Al-Kofahi, Y., Dowell-Mesfin, N., Pace, C., Shain, W., Turner, J., Roysam, B. (2007). Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images. *Cytometry Part A*, 73(1), 36–43.
- Brown, K., Barrionuevo, G., Canty, A., De Paola, V., Hirsch, J., Jefferis, G., et al. (2011). The DIADEM data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*. doi:10.1007/s12021-010-9095-5.
- Calixto, E., Galván, E., Card, J., & Barrionuevo, G. (2008). Coincidence detection of convergent perforant path and mossy fibre inputs by CA3 interneurons. *The Journal of Physiology*, 586(11), 2695–2712.
- De Paola V., Holtmaat A., Knott G., Song S., Wilbrecht L., Caroni P., et al. (2006). Cell type-specific structural plasticity of axonal branches and boutons in the adult neocortex. *Neuron*, 49(6), 861–875.
- Dima, A., Scholz, M., & Obermayer, K. (2002). Automatic segmentation and skeletonization of neurons from confocal mi-

- crosscopy images based on the 3-D wavelet transform. *IEEE Transactions on Image Processing*, 11(7), 790–801.
- Evers, J., Schmitt, S., Sibila, M., & Duch, C. (2005). Progress in functional neuroanatomy: Precise automatic geometric reconstruction of neuronal morphology from confocal image stacks. *Journal of Neurophysiology*, 93(4), 2331.
- Jefferis, G., Potter, C., Chan, A., Marin, E., Rohlfling, T., Maurer, C., et al. (2007). Comprehensive maps of Drosophila higher olfactory centers: Spatially segregated fruit and pheromone representation. *Cell*, 128(6), 1187–1203.
- Meijering, E. (2010). Neuron tracing in perspective. *Cytometry Part A*, 77(7), 693–704.
- Meijering, E., Jacob, M., Sarría, J., Steiner, P., Hirling, H., & Unser, M. (2004). Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A*, 58(2), 167–176.
- Narro, M., Yang, F., Kraft, R., Wenk, C., Efrat, A., & Restifo, L. (2007). NeuronMetrics: Software for semi-automated processing of cultured neuron images. *Brain Research*, 1138, 57–75.
- Peng, H. (2008). Bioimage informatics: A new area of engineering biology. *Bioinformatics*, 24(17), 1827–1836.
- Peng, H., Ruan, Z., Atasoy, D., & Sternson, S. (2010a). Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model. *Bioinformatics*, 26(12):i38–i46.
- Peng, H., Ruan, Z., Long, F., Simpson, J., & Myers, E. (2010b). V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nature Biotechnology*, 28(4), 348–353.
- Rodríguez, A., Ehlenberger, D., Hof, P., & Wearne, S. (2006). Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images. *Nature Protocols*, 1(4), 2152–2161.
- Rodríguez, A., Ehlenberger, D., Hof, P., & Wearne, S. (2009). Three-dimensional neuron tracing by voxel scooping. *Journal of Neuroscience Methods*, 184(1), 169–175.
- Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., et al. (1998). Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2), 143–168.
- Schmitt, S., Evers, J., Duch, C., Scholz, M., & Obermayer, K. (2004). New methods for the computer-assisted 3-D reconstruction of neurons from confocal image stacks. *Neuroimage*, 23(4), 1283–1298.
- Sugihara, I., Wu, H., & Shinoda, Y. (1999). Morphology of single olivocerebellar axons labeled with biotinylated dextran amine in the rat. *The Journal of Comparative Neurology*, 414(2), 131–148.
- Vasilkoski, Z., & Stepanyants, A. (2009). Detection of the optimal neuron traces in confocal microscopy images. *Journal of Neuroscience Methods*, 178(1), 197–204.
- Wearne, S., Rodríguez, A., Ehlenberger, D., Rocher, A., Henderson, S., & Hof, P. (2005). New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience*, 136(3), 661–680.
- Weaver, C., Hof, P., Wearne, S., & Lindquist, W. (2004). Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural Computation*, 16(7), 1353–1383.
- Wright, S., & Nocedal, J. (2006). *Numerical optimization*. New York: Springer.
- Xie, J., Zhao, T., Lee, T., Myers, E., & Peng, H. (2010). Automatic neuron tracing in volumetric microscopy images with anisotropic path searching. In: *Medical image computing and computer-assisted intervention—MICCAI 2010* (pp. 427–429). Springer.
- Yuan, X., Trachtenberg, J., Potter, S., & Roysam, B. (2009). MDL constrained 3-D grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7(4), 213–232.
- Zhang, Y., Zhou, X., Degterev, A., Lipinski, M., Adjeroh, D., Yuan, J., et al. (2007). Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays. *NeuroImage*, 35(4), 1502–1515.