

# Improved Automatic Centerline Tracing for Dendritic and Axonal Structures

David Jiménez · Demetrio Labate ·  
Ioannis A. Kakadiaris · Manos Papadakis

© Springer Science+Business Media New York 2014

**Abstract** Centerline tracing in dendritic structures acquired from confocal images of neurons is an essential tool for the construction of geometrical representations of a neuronal network from its coarse scale up to its fine scale structures. In this paper, we propose an algorithm for centerline extraction that is both highly accurate and computationally efficient. The main novelties of the proposed method are (1) the use of a small set of Multiscale Isotropic Laplacian filters, acting as self-steerable filters, for a quick and efficient binary segmentation of dendritic arbors and axons; (2) an automated centerline seed points detection method based on the application of a simple 3D finite-length filter. The performance of this algorithm, which is validated on data from the DIADEM set appears to be very competitive when compared with other state-of-the-art algorithms.

---

**Electronic supplementary material** The online version of this article (doi:10.1007/s12021-014-9256-z) contains supplementary material, which is available to authorized users.

---

D. Jiménez  
Department of Mathematics, University of Costa Rica,  
San Pedro Montes de Oca, San José,  
Republic of Costa Rica  
e-mail: david.jimenezlopez@ucr.ac.cr

D. Labate · M. Papadakis (✉)  
Department of Mathematics, University of Houston,  
641 Phillip G Hoffman Hall, Houston, TX 772004-3008, USA  
e-mail: mpapadak@math.uh.edu

D. Labate  
e-mail: dlabate@math.uh.edu

I. A. Kakadiaris  
Computational Biomedicine Lab, Department of Computer  
Science, University of Houston, 4800 Calhoun Rd,  
Houston, TX 77204, USA  
e-mail: ioannisk@uh.edu

**Keywords** Image processing · Automated neuron tracing · Neuron image segmentation

## Introduction and Previous Work

Recent advances in high resolution fluorescent microscopy have opened up tremendous opportunities to study neuronal morphology, including the dynamic changes by which neurons respond to external stimuli. It is well known that these morphological changes have fundamental implications in learning and memory formation. To this goal, it is essential to develop accurate and efficient tools to segment neuronal structures and quantify their associated morphological characteristics. Deriving the graph connectivity of the centerline of a dendritic arbor or a bundle of axons and using this graph as the backbone for representing the geometry of that type of tubular structure is essential to building a geometrical representation of a neuron. Clearly, this imaging problem is present in other types of applications such as retinal imaging and the extraction of vasculature in cardiac CT-angiography.

In this paper, we propose an algorithmic suite for the binary segmentation of 3D-tubular structures and the semi-automatic extraction of their centerlines. The proposed methods are competitive in terms of accuracy and computational efficiency. Unlike with most centerline extraction methods, our approach is based on an accurate binary segmentation of the tubular volume of interest from the background. We apply the proposed centerline extraction algorithm to derive the graph connectivity of a neuron. Our methods are fairly general and can be applied for the binary segmentation of tubular structures from the background in images acquired with other modalities, such as X-ray CT or MRI. However, at this point we cannot back this claim with experimental evidence, although the mathematical analysis

of the principles governing our method suggest that this claim is true. The main novelty of the proposed method is the combined use of a small set of Isotropic Laplacian filters acting as self-steerable filters for a quick and efficient binarization of the axonal or of the dendritic structure together with the application of a simple 3D finite-length filter which *automatically* detects the seed points for the centerline extraction. We also contribute a thorough theoretical analysis of why those novelties work to the extent that the readability of this paper by its natural audience is not compromised.

Although the problem of accurately extracting the graph connectivity of axons or dendritic structures still requires a significant amount of human intervention, methods for the computation of the graph connectivity of tubular structures have been extensively studied, including methods for tracing dendritic or axonal centerlines and identifying their branching and terminal points.

A commonly used strategy to determine the centerline in binarized images is *skeletonization* (Bas and Erdogmus 2011; Koh 2001; Zhou and Toga 1999), where the binary volume undergoes an iterative thinning process. However, skeletonization is not particularly robust to noise, and outputs of these algorithms often contain spurious gaps, branches and crossings (Meijering 2010). Although several strategies have been developed to address these weaknesses in a post-process step (Wearne et al. 2005; Yuan et al. 2009). Other variants of this approach include the work of Morrison and Zou (2006), who effectively mitigate errors due to noise. Unfortunately, the computational overhead of these algorithms is rather significant.

Some methods propose to achieve the morphological reconstruction by relying on geometric properties native to the dimensionality of these types of images; e.g., by fitting cylindrical structures to segments of a binarized 3D image as first explored in Al-Kofahi et al. (2002) and more recently (Zhao et al. 2011). As explained by Donohue and Ascoli (2010), most of these methods require the processing of the entire image, thus increasing the computational cost and making the performance sensitive to difference in staining intensity over the image, which may result in producing gaps in the detected structures. This last drawback has been partially addressed in several papers, such as in Xiong et al. (2006), where a global post-process analysis of the image is used to detect and remove these gaps.

Other methods for centerline extraction require the selection of a *point source* (Turetken et al. 2011), also called a *seed point*, that may be automatically detected (Wang et al. 2011), randomly selected (Chothani et al. 2011), or user-provided, (Santamaria-Pang et al. 2007, 2014). The point source initializes a process of wave propagation as in Hassouna and Farag (2005) or of a fast marching algorithm as in Santamaria-Pang et al. (2007, 2014) which

trawls the centerline until it reaches some *terminal points*, and then it re-traces the centerline with back propagation until the source point is reached. While this approach is numerically efficient, its main drawback is that its performance tends to be quite sensitive to the choice of the point source, leading to the risk of potentially missing entire branches of the segmented volume. Other interesting seed-based approaches are the probabilistic extraction of centerlines proposed by Turetken et al. (2011) and by Breitenreicher et al. (2013), which also use Dijkstra's algorithm but in a very different way than the one adopted in this work (see also Turetken et al. (2012, 2013)). Their approach is the construction of a minimum path tree based on the conditional maximum likelihood information derived from the image given certain prior models learned during training. By contrast, we use Dijkstra's algorithm in an exclusively deterministic setting, relying on the accurate segmentation of the volume of the tubular structure from the background.

Another interesting approach was recently proposed by Rodriguez et al. (2009), where a process of *voxel scooping* is iteratively performed starting from the point source. Rodriguez et al. assume that object voxels are readily distinguishable from the background in the image volume. A user-provided point source initializes an iterative algorithm that divides the solid into transversal layers, where a local analysis identifies *seed nodes* of the centerline graph, whose provisional parent node is the closest node generated on the immediately previous iteration. A post-process is subsequently performed to prune short, potentially false, branches.

There are currently several academic (e.g., Hines and Carnevale 2001; 3D-Slicer 2008; Luisi et al. 2011; Peng et al. 2011; Santamaria-Pang et al. 2007, 2014) and other freeware imaging suites (e.g., Scorcioni and Polavaram 2008) delivering morphological reconstructions of neurons including centerline tracing. Their performances vary and depend on the level of training per dataset, the noise that affects the data and the level of post-analysis manual intervention. It is also worth to note that significant work on the tracing and morphological reconstruction of dendritic arbors emerged as a result of the DIADEM Challenge (Brown et al. 2011) among them FAR-SIGHT (Luisi et al. 2011). This algorithmic suite provides centerline extraction and morphological reconstructions of neuronal dendrites and axons.

Although the strategy of our approach bears some similarity to the strategy recently proposed by Xie et al. (2010), (2011), our method is different. The first step of the algorithm by Xie et al. segments the solid by applying a global threshold. This thresholding strategy is commonly used due to speed and simplicity. Nevertheless, this binarization method is not robust (Meijering 2010). In contrast, as

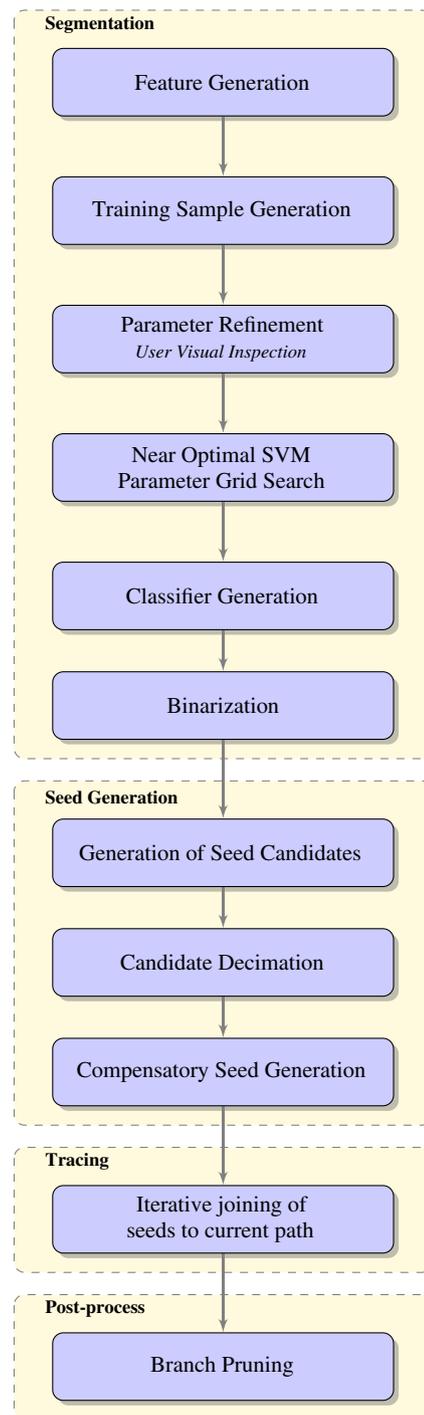
we will show below, our approach provides a more reliable and accurate binarization by means of the use of a special collection of band pass, self-steerable filters and Support Vector Machine classifiers. Moreover, the seeding and tracing algorithms we propose have a rigorous and simple mathematical justification and a more streamlined implementation, which avoids the forward and backward propagation algorithms used in ORION (Kakadiaris and Colbert 2007) and FARSIGHT (Wang et al. 2011), reducing the overall computational overhead.

## Methods

In this paper, we propose a semi-automated centerline extraction algorithm consisting of four main steps: (i) segmentation, the process of classifying the volume's voxels into foreground and background; (ii) seed generation for centerline extraction the binarized tubular structure; (iii) Tracing; (iv) automatic post-process, the search for spurious branches and other artifacts, and their correction. The workflow of our approach is shown in Fig. 1. We refer to the process of seed generation and tracing as centerline extraction.

Our algorithm performs the segmentation of the solid of interest by means of an SVM classifier (Cortes and Vapnik 1995) which uses features generated by a novel set of band pass and self-steerable filters. In theory, these features are exclusively applicable to tubular structures as the study of the mathematical properties of these filters indicates (Jimenez and Labate 2014). This classifier is generated using only a part of the volume. If such a classifier has already been generated, the segmentation process is automatic and follows the last three steps (marked as purple boxes) within the segmentation block of the workflow diagram illustrated in Fig. 1. Nevertheless, a classifier may lead to good segmentations on datasets similar to those used to generate it but perform poorly on sets acquired under significantly different circumstances (e.g. different cell type, different microscope settings). Thus, we have included in the first part of the segmentation process an optional SVM classifier training that does require user intervention and follows the workflow illustrated in the first three steps of the segmentation box of Fig. 1. Below, in section. “[Feature Generation for the Segmentation of Tubular Structures](#)”; we will devote part of our discussion to the feature generation and training of this SVM classifier.

Following the segmentation, to extract the centerline we perform two subprocesses: first, we identifying seed points, i.e., voxels belonging to the centerline of the tubular structure with high probability; next, we trace the centerline by essentially Dijkstra connecting the seed voxels using our own variation of Dijkstra's algorithm (Dijkstra 1959).



**Fig. 1** Workflow of the proposed automatic centerline tracing algorithmic suite

### Feature Generation for the Segmentation of Tubular Structures

The binarization (or binary segmentation) of various neuronal structures of interest, such as somas, dendritic arbors and axons, from the background is carried out with a standard SVM classifier. As illustrated in the workflow of

Fig. 1, the training of the classifier consists of three stages: feature generation, training sample generation and parameter refinement (or optimization). Our main contribution in this segmentation process is feature generation. Features are generated using the following set filters  $F_1, \dots, F_N$  that are designed following to the theoretical requirements of Theorem 1 that we will state below.

1. Low pass filters of the form<sup>1</sup>:

$$\widehat{F}(\xi) = P_n \left( C_{n,\sigma} \|\xi\|^2 \right) e^{-C_{n,\sigma} \|\xi\|^2},$$

with  $\sigma > 0$ , defined to capture low scale shape characteristics (see Fig. 2). Here  $P_n$  is the Taylor polynomial of degree  $n$  associated with the exponential function  $e^x$ , and

$$c_{n,\sigma} = \frac{2n + 1}{2(K\sigma)^2},$$

where  $K$  is a positive integer ( $K = 3$  in our experiments). The filters are called Hermite Distributed Approximating Functionals and were first proposed in Hoffman and Nayar (1991) (see also Hoffman et al. 1996, 2002; Bodmann et al. 2007; Chandler and Gibson 1999). The properties of these filters are discussed after the statement of Theorem 1 below.

2. Isotropic high pass band filters:

$$\widehat{F}(\xi) = P_n \left( c_{n,\sigma_1} \|\xi\|^2 \right) e^{-c_{n,\sigma_1} \|\xi\|^2} - P_n \left( c_{n,\sigma_2} \|\xi\|^2 \right) e^{-c_{n,\sigma_2} \|\xi\|^2},$$

with  $\sigma_1 > \sigma_2$ , defined to capture higher scale shape details.

3. Isotropic Laplacian bandpass filters:

$$\widehat{F}(\xi) = \|\xi\|^2 P_n \left( c_{n,\sigma} \|\xi\|^2 \right) e^{-c_{n,\sigma} \|\xi\|^2},$$

to capture transitions in the image (see Fig 6).

We will discuss the properties of these filters that are relevant to the work presented here while keeping the discussion at a non-technical level in order to make the presentation more accessible. The mathematical properties of these isotropic filters on tubular structures are described in detail by Theorem 2.2 and Proposition 2.3 in Ozcan et al. (2013), where we theoretically predict the outcomes of the interaction of these filters with the tubular structure. The main property of these filters is the ability to act as self-steering directional filters (see item 4 of Theorem 1 below), that is the ability to discover the dominant orientation of a tubular structure.

The tubular structure  $I$  can be modeled or approximated as a finite sum of the form

$$I = \sum_{i=1}^n \sum_{k=1}^K \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} T_{x_i} R_k a_{i,k,j_1,j_2} f_{\sigma_{j_1},\sigma_{j_2}} \quad (1)$$

<sup>1</sup>The roof over the symbol of the filter denotes its Fourier transform.

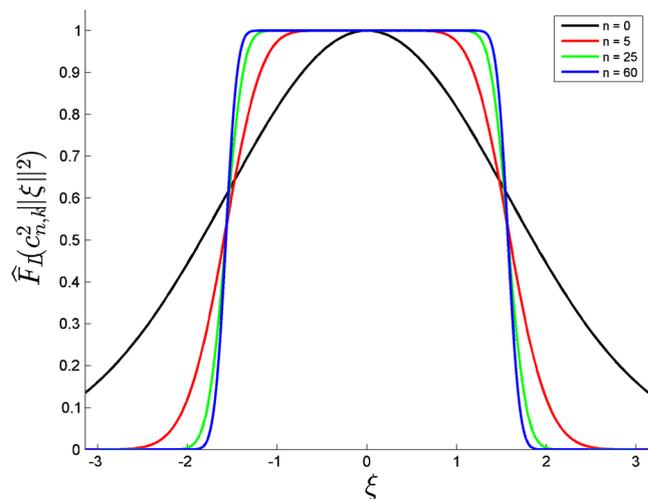


Fig. 2 Values of the Fourier transform of the Isotropic low pass filters  $F$  used for feature extraction. Notice how their transition band changes as  $n$  increases

where  $f_{\sigma_1,\sigma_2}(x, y, z) = e^{-\frac{x^2}{2\sigma_1^2}} e^{-\frac{y^2+z^2}{2\sigma_2^2}}$ ,  $x, y, z, \in \mathbb{R}$ ,  $R_k \in \text{SO}(3)$  are 3-D rotations,  $T_{x_i}g(x) = g(x - x_i)$ , and  $a_{i,k,j_1,j_2} > 0$ . The use of 2D-radial Gaussians to model cross sections of tubular structures has been adopted by several authors, among them (Sato et al. 1998; Frangi et al. 1998; Sato et al. 2000; Krissian et al. 2000), who actually use this model to justify why the eigenvalues of the Hessian matrix at each pixel can be used for the binarization of the tubular structure. In particular, (Krissian et al. 2000) attempt an in-depth mathematical analysis of this problem.

We give below a simpler statement of our theorem. We refer the reader to Jimenez and Labate (2014) for a generalization of this theorem, where we remove the assumption of radial symmetry on the intensity function in a cross-section of the tubular structure. This generalization is useful to deal with other ‘less tubular’ structures such as somas and spines. We use the notation  $\mathbf{v}$  to indicate a point in the physical 3-D space. However, for all practical purposes we can think of it as a voxel, although the model prescribed by Theorem 1 is valid only in the physical 3-D domain.

**Theorem 1** (Ozcan et al. 2013, Theorem 2.2) *Suppose that  $\phi$  is a radial filter, whose partial derivatives up to second order  $D^\alpha \phi$  are continuous and satisfy  $\|\mathbf{x}\|^4 |D^\alpha \phi(\mathbf{x})| < A$ , where  $A > 0$ , for all<sup>2</sup> multi-indices  $|\alpha| \leq 2$  and  $\mathbf{x} \in \mathbb{R}^3$ . Then the following statements hold.*

1. *Filtering the tubular structure  $I$  (1) with the 3D radial filter  $\phi$  amounts to filtering the cross section of the*

<sup>2</sup> These properties imply that the filter  $\phi$  and all of its derivatives up to second order are well-localized in space. This means that, for practical purposes, the spatial support of  $\phi$  and of its derivatives up to second order is small.

tubular structure at each point with a 2D radial filter  $\omega(y, z) := \int_{\mathbb{R}} \phi(x, y, z) dx$ . More precisely we have:

$$I * \phi(\mathbf{v}) \approx g_{\sigma_{j_2}} * \omega(p(\mathbf{v})),$$

where,  $p(\mathbf{v})$  is the projection in the 3-D space of  $\mathbf{v}$  on the plane containing  $\mathbf{v}$  and perpendicular to the centerline of the tubular structure,

$$g_{\sigma}(y, z) = e^{-\frac{y^2+z^2}{2\sigma^2}}$$

and  $j_2$  corresponds to the component of  $I$  which is most proximal to  $\mathbf{v}$ .

2. Filtering the tubular structure with the 3D Isotropic Laplacian filter  $\Delta\phi$  practically amounts to applying the 2-D Laplacian of  $\omega$  on the cross section of the tubular structure  $I$ : Specifically, for every  $\mathbf{v}$  as in the previous item we have

$$I * (\Delta\phi)(\mathbf{v}) \approx g_{\sigma_{j_2}} * (\Delta_{y,z}\omega)(p(\mathbf{v})),$$

$$\text{where, } \Delta_{y,z} = \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

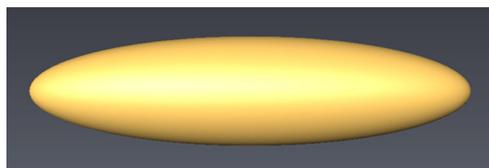
3. The outcome of the filtering process of  $I$  with both  $\phi$  and  $\Delta\phi$  depends only on the relative position of the point  $\mathbf{v}$  with respect to the cross section of  $I$  with the plane containing  $\mathbf{v}$  and on the properties of this cross section and not by the local 3D-orientation of the tubular structure.
4. At every grid point (voxel) inside  $I$  or proximal to it, the filtering of  $I$  with the 3D isotropic filters  $\phi$  and  $\Delta\phi$  is equivalent to the filtering of  $I$  with directional filters, automatically aligning themselves with the axis of the most proximal segment of  $I$  to the voxel. Moreover, the cross sections of these directional filters at their center of symmetry are equal to  $\omega$  and  $\Delta_{y,z}\omega$  respectively.

To extract features for the detection of  $I$  we compute the filter outputs  $F_i * I$ . The use of an SVM classifier requires the use of normalizing constants  $\alpha_i$  satisfying

$$\max_{\mathbf{v} \in I} \alpha_i \{|(F_i * I)(\mathbf{v})|\} = 1.$$

For each voxel  $\mathbf{v}$ , we define the feature vector corresponding to  $\mathbf{v}$  as  $(\alpha_1(F_1 * I)(\mathbf{v}), \dots, \alpha_N(F_N * I)(\mathbf{v}))$ . Note that filters  $F_i$  are radial and satisfy the assumptions of Theorem 1.

*Remark* Items 2, 3 and 4 of Theorem 1 precisely describe how isotropic filters act as self-steerable filters in the presence of a tubular structure. We have observed, though, that self-steerability works for other types of structures. Figures 3 and 4 show the filter  $\phi$ . As item 3 of Theorem 1 suggests, feature extraction does not practically depend on the local orientation and position of the tubular structure in the 3D space. For each voxel, a single feature from the ones we use exclusively depends on the geometric properties of the tubular structure and on the local intrinsic coordinates of the voxel with respect to the structure.



**Fig. 3** Volume rendering (isosurface) of the directional, self-steerable filter derived from  $\phi$  which automatically aligns itself locally with the axis of the tubular structure

Now, let  $\omega(y, z) := \int_{\mathbb{R}} h(x, y, z) dx$ , where  $h$  in the inverse Fourier transform (impulse response) of

$$\hat{h}(\xi) = P_n \left( C_{n,\sigma} \|\xi\|^2 \right) e^{-C_{n,\sigma} \|\xi\|^2}, \quad \xi \in \mathbb{R}^3,$$

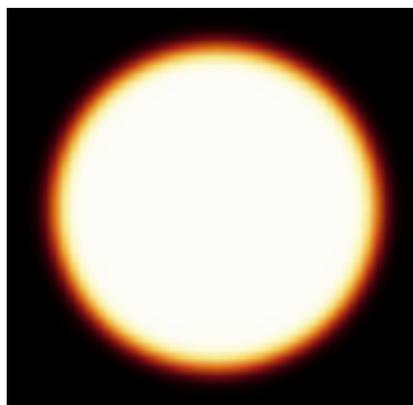
which is the generic form of the low-pass isotropic filters we use for feature extraction. We will now mathematically predict the action of the isotropic Laplacian filters we use.

The previous remark, item 2 of Theorem 1 and the rapid decay in the spatial domain of the Gaussian dominated filter  $h$  (this property is also inherited by  $\omega$ ) imply that if  $I$  is filtered with the Laplacian 3D-isotropic filters  $\hat{F}(\xi) = \|\xi\|^2 \hat{h}(\xi)$ , at the grid point (voxel)  $\mathbf{v}$ , then the output of the filtering is

$$(\Delta_{y,z} g_{\sigma_{j_2}}) * \omega(p(\mathbf{v})) = g_{\sigma_{j_2}} * (\Delta_{y,z}\omega)(p(\mathbf{v})). \tag{2}$$

Implicitly, we assume that  $\mathbf{v}$  is not proximal to any other part of the tubular structure  $I$ , so the effect of the filtering of  $I$  with  $F$  is restricted to the part of the tubular structure closest to  $\mathbf{v}$ , whose cross section is modeled by  $g_{\sigma_{j_2}}$ .

The choice of the constant  $C_{n,\sigma}$  places the inflection point of the radial profile of  $\hat{h}$  firmly at radius  $K\sigma$  from the origin, regardless of the value of  $n$ . As  $n$  increases to  $\infty$  (Bodmann et al. 2007, Remark 3.4) the width of the



**Fig. 4** Cross section at the center of symmetry of the self-steerable, directional filter shown in Fig. 3. This cross section is equal to a specific example of the Fourier transform of  $\omega$ . Here, black indicates zero values while white shows the set where  $\hat{\omega}$  takes the value 1. The white area in the cross-section indicates the cylindrical bandwidth of those tubular structures whose boundaries this filter can adequately detect from the background, as predicted by Theorem 1

radial profile of the transition band of  $\hat{h}$  is proportional to  $\frac{1}{n}$ . This transition band also contains the inflection point of the radial profile of  $\hat{h}$ , for every  $n$ . Moreover, as  $n$  grows, the values of  $\hat{h}$  tend to 1 at every point in the ball centered at the origin with radius  $K\sigma$  (Bodmann et al. 2007, Th. 3.7). *In a nutshell, the low pass filter  $\hat{h}$  asymptotically behaves like a radial ideal filter with pass-band being the ball centered at the origin and radius equal to  $K\sigma$ .* Since  $\sigma$  is a measure of the bandwidth corresponding to a segment of interest of the tubular structure, the bandwidth of the isotropic filter  $h$  applied at this segment for feature extraction has bandwidth  $K\sigma$ .

Using Eq. 2 and following the mathematical analysis of Proposition 2.3 in Ozcan et al. (2013), which states that if  $K$  is selected big enough (in practice  $K \geq 3$ )  $\sigma = \frac{1}{\sigma_{j_2}}$  and  $n$  is also sufficiently big (e.g.,  $n = 60$ ), then the values of  $I * F$  faithfully approximate the values of  $\Delta_{y,z}g_{\sigma_{j_2}}$ , we infer that, up to a certain radius sufficiently away from  $I$ 's centerline and always perpendicularly to it (this is precisely the effect of the self-steerability of the proposed filtering process), the sign of the values of  $I * F$  will be the same as the sign of the values of the cross-sectional Laplacian  $\Delta_{y,z}g_{\sigma_{j_2}}$ . The same analysis shows that further away, but at distances that do not seem to influence the SVM classifier, the sign of the values of  $I * F$  is no longer exclusively positive, as the errors of the approximation of  $\Delta_{y,z}g_{\sigma_{j_2}}$  dominate the outcome. Since the diameters of the tubular structure of interest vary, we use several filters to account for different ranges of values of  $g_{\sigma_{j_2}}$ . Laplacian Filters with a high bandwidth act as singularity detectors and, thus, they are not suitable for the thicker parts of the structure since in the interior of the tubular structure they are unable to sense the difference from the exterior due to the scale difference between the structure and the filter. In other words, filter and structure should live at the same scale in order for the filter to sense that the tubular geometry.

Using now the formula of the 2D Laplacian in polar coordinates we obtain

$$\Delta_{y,z}g_{\sigma_{j_2}}(r) = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial}{\partial r} \left( e^{-\frac{r^2}{2\sigma_{j_2}^2}} \right) \right) = \frac{e^{-\frac{r^2}{2\sigma_{j_2}^2}}}{\sigma_{j_2}^2} \left( \frac{r^2}{\sigma_{j_2}^2} - 1 \right)$$

This calculation implies that the “sign-change point” of  $g_{\sigma_{j_2}}$  in the radial direction is located at radius  $\sigma_{j_2}$  from the origin. This radius coincides with the distance from the origin of the circle of all points of the cross-sectional plane at which the sign of  $\Delta_{y,z}g_{\sigma_{j_2}}$  changes to positive. On the other hand, in the annulus  $[\sigma_{j_2}, 2\sigma_{j_2}]$ , the intensity values of  $g_{\sigma_{j_2}}$  decrease smoothly and become small, thus corresponding to a region which can be considered to be the “skin” of the dendritic

branch or of the axon. This model suggests that indeed the set of positive values of  $\Delta_{y,z}g_{\sigma_{j_2}}$  in a cross-sectional plane of  $I$  belongs to the exterior of the tubular structure. This sign change is precisely the property that our feature extraction with the self-steerable filtering exploits in order to detect the skin of the tubular structure and segment them via an SVM-classifier from the background. Our analysis above aims to justify this observation and to determine the proper filter parameter selection for the optimal performance of feature extraction.

Note that salt and pepper noise present in the data can potentially corrupt the outputs of self-steerable filters. Since noise has mostly high frequency content, the use of the low pass filter  $h$  mitigates its effects. Nevertheless, voxels where the Laplacian filtering assigns relatively small values are more likely to be associated with noise. Therefore, voxels assigned positive or negative values with magnitude exceeding a certain threshold have a higher likelihood of being associated with the background or to belong to the tubular structure, respectively. This threshold is determined in the training process by the expert user and it may change on each data set.

This SVM training procedure was inspired in part by ORION (Santamaria-Pang et al. 2007), with the main difference being that here we use a very different class of filters and propose a novel method of determining the training ROIs inside and outside of the tubular structure. Namely, a human operator chooses the training ROIs  $R_i$  and the parameters  $p_i$  and  $n_i$  corresponding to the percentages of the total number of positive and negative samples, respectively, that will be used as training examples. The sample voxels for the training are randomly selected among voxels from the training ROIs at rates  $p_i$  and  $n_i$ . Typically  $p_i$  and  $n_i$  range between 0.5% and 2%. The operator can visually inspect the sample sets and approve them or adjust the parameters  $p_i$  and  $n_i$  accordingly; the operator can also fine-tune the training ROIs by plain intensity thresholding in order to generate sample sets as representative and accurate as possible. Once the training samples and their respective feature vectors are obtained, a grid search for the SVM parameter determination is performed via a four-fold validation process. Upon completion of the search, the classifier is generated.

## Binarization

Often, in a neuronal volume, it is expected that the number of object voxels will be considerably smaller than the number of the background voxels. Moreover, background voxels typically have lower intensity values than voxels belonging to the tubular structure. Thus, if we compute  $\mu$ , the average intensity of  $I$ , the voxels with intensity values lower than  $\mu$  can safely be considered to belong to the

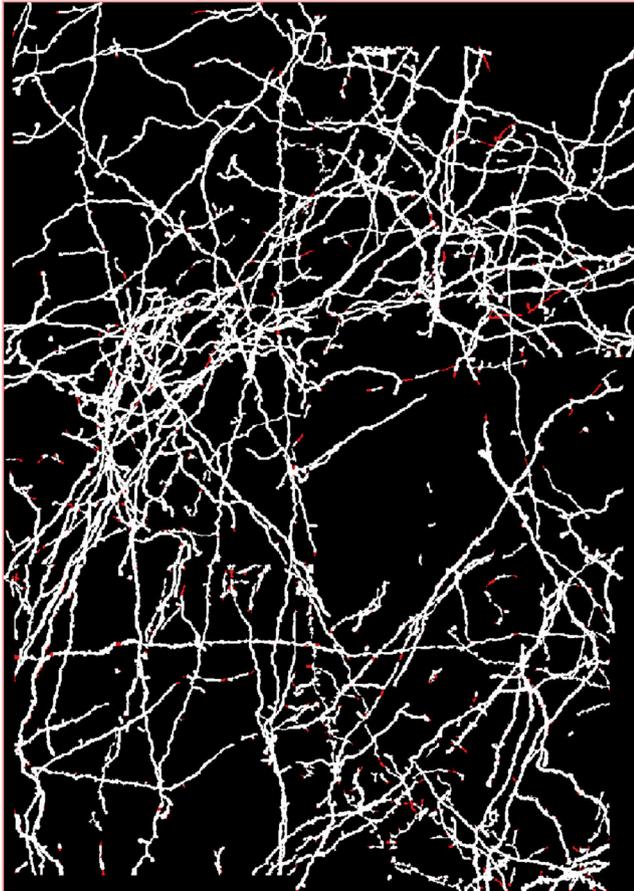
background. This preliminary crude segmentation step reduces the computational overhead of the binarization process.

To segment the remaining voxels, we use the previously generated SVM classifier. This classifier has been stored in memory together with the specifications of the filters  $F_1, \dots, F_N$  and normalizing constants  $\alpha_1, \dots, \alpha_N$  used to generate it. The use of these constants is typical for SVMs since all features need to have same range of values to ensure that the training process is not biased.

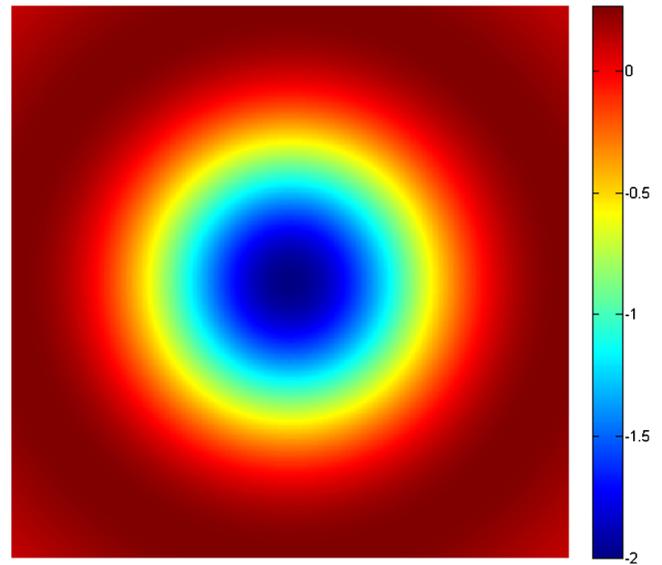
For each test voxel  $\mathbf{v}$ , we generate the feature vector

$$(\alpha_1(F_1 * I)(\mathbf{v}), \dots, \alpha_N(F_N * I)(\mathbf{v}))$$

which is used as an input in the classifier derived with training. The output of this process is the binary 3-D segmentation of the original input image stack. In Figs. 5 and 15 we show MIPs of some segmented volumes from the DIADEM datasets (Fig. 6).



**Fig. 5** Maximum intensity projections of the segmented datasets and superimposed reconstruction for the Neocortical Layer 1 Axon DIADEM set: White indicates the volume ORION (Santamaria-Pang et al. 2007) segmented in this data set. Red marks the volume which the proposed method segmented in addition to the part segmented by ORION



**Fig. 6** Color representation of the Laplacian of a 2D Gaussian bell. Notice that the values with higher magnitude are negative, so the territory of red indicates values close to zero

#### Selection of Seed Points

The selection of seed points from the binary volume is performed in three steps: filter seeding, where points with a high probability of belonging to the centerline are extracted; seed decimation, where the seed candidates are evaluated with respect to a set of criteria, and a large portion of them are eliminated; and compensatory seeding, where certain regions of the solid are re-examined due to inadequate seed selection. The approximation of the segmented volume with voxels (binarization) allows geometric inaccuracies to creep in. More precisely, both the centerline and the tubular structure are defined not on the continuous 3D space but on a discrete grid (more precisely on a sampling grid) and it is, thus, not possible in general to obtain cross sections of the tubular structures precisely perpendicular to the centerline. This limitation motivates us to find a necessary condition to identify candidate centerline voxels via an averaging process which we next describe.

The first step of this process is called *filter seeding*. Let  $B$  be the set of voxels in the segmented solid after binarization. More precisely, we define  $B$  to be the set of all voxels in the segmented 3-D image whose value is equal to 1. Each voxel  $\mathbf{v}$  is identified by the integer coordinates  $(x, y, z)$  of its center. We call

$$d(\mathbf{v}) = \min\{\|\mathbf{v} - \mathbf{p}\| : \mathbf{p} \notin B\},$$

where  $\|\cdot\|$  is the Euclidean norm, the distance of  $\mathbf{v}$ . Throughout the analysis that follows, we assume that  $\mathbf{v}$  and the orientation and diameter of the tubular structure remain

constant in a small neighborhood of  $\mathbf{v}$ . The fact that filter seeding is a process that is taking place in the interior of a tubular structure leads us to the simplifying modeling assumption that *any cross section of the tubular structure is symmetric with respect to the centerline voxel belonging to it*. Moreover, *the voxel  $\mathbf{v}$  in the interior of the tubular structure belongs to the centerline if and only if  $\mathbf{v}$  maximizes  $d$  in a small neighborhood of  $\mathbf{v}$* .

Now, suppose that  $\mathbf{v}$  belongs to the centerline of the tubular structure and that it shares no vertex with any voxel outside of the tubular structure. This assumption disallows, for example, cross-shaped sections of the tubular structure perpendicular to its centerline. In fact, it implies that  $\mathbf{v}$  and its 27-connected neighborhood  $N(\mathbf{v})$  are contained in the interior of the tubular structure. Then, under this and the previous hypotheses for the geometry of the tubular structure, we have

$$d(\mathbf{v})^2 \geq d(\mathbf{w})^2 + 1,$$

where  $\mathbf{w}$  is in  $N(\mathbf{v})$  and is not a centerline voxel. This implies that

$$d(\mathbf{v}) - d(\mathbf{w}) \geq \frac{1}{d(\mathbf{v}) + d(\mathbf{w})}$$

and, thus,

$$d(\mathbf{v}) - d(\mathbf{w}) \geq \frac{1}{2d(\mathbf{v})} \tag{3}$$

Notice that  $N(\mathbf{v})$  contains exactly two other centerline voxels besides  $\mathbf{v}$ . We exploit these elementary observations to identify candidate centerline voxels of the tubular structure. First, we compute the distance map  $d(I)$  of  $I$ . Subsequently, we convolve  $d(I)$  with  $A$ , where  $A$  is a  $3 \times 3 \times 3$  filter with  $A_{0,0,0} = 1$ , and  $A_{i,j,k} = -1/26$  for all  $(i, j, k) \neq (0, 0, 0)$ . Now, let  $N_0(\mathbf{v})$  consist of all voxels of  $N(\mathbf{v})$  but  $\mathbf{v}$ . The quantity that we obtain

$$(A * d(I))(\mathbf{v}) = \frac{26}{26}d(\mathbf{v}) - \frac{1}{26} \sum_{\mathbf{w} \in N_0(\mathbf{v})} d(\mathbf{w})$$

is the difference between the distance of  $\mathbf{v}$  from the boundary of  $I$  and the average distance of all its neighboring voxels again from the boundary of  $I$ . Finally, we normalize the values of  $A * d(I)$  by dividing it with the maximum of its values.

Next, in order to justify the previously described averaging method for the identification of voxels in the centerline (seeds), recall the assumption that the orientation and the diameter of the tubular structure remain locally constant. Let  $\mathbf{v}$  be a centerline point which shares no common vertex with any exterior voxel. Since exactly two other voxels in  $N(\mathbf{v})$  are on the centerline and all of the geometric properties of the tubular structure remain constant in the vicinity of  $N(\mathbf{v})$ ,

we conclude that  $d(\mathbf{v})$  is the value of the distance function for the other two centerline voxels in  $N(\mathbf{v})$ . So,

$$(A * d(I))(\mathbf{v}) = \frac{24}{26}d(\mathbf{v}) - \frac{1}{26} \sum_{\mathbf{w} \in N_0(\mathbf{v})} d(\mathbf{w}),$$

where the factor 24/26 accounts for the all three centerline voxels in  $N(\mathbf{v})$ . Using Eq. 3, we conclude

$$\frac{24}{26}d(\mathbf{v}) - \sum_{\mathbf{w} \in N_0(\mathbf{v})} d(\mathbf{w}) \geq \frac{12}{26d(\mathbf{v})}.$$

Now assume that  $\mathbf{v}$  is not on the centerline. Then, if  $\mathbf{w}_1$  and  $\mathbf{w}_2$  are immediate neighbours of  $\mathbf{v}$ , diametrically opposite to each other with respect to  $\mathbf{v}$ , they should satisfy  $d(\mathbf{w}_1) + d(\mathbf{w}_2) \approx 2d(\mathbf{v})$ , because one of  $\mathbf{w}_1$  and  $\mathbf{w}_2$  is closer to the boundary of the tubular structure than the other. Consequently,  $(A * d(I))(\mathbf{v}) \approx 0$ .

The above discussion implies that, after normalization with the maximum value of  $d$  on  $N(\mathbf{v})$ , the set of the sought seed points is contained in

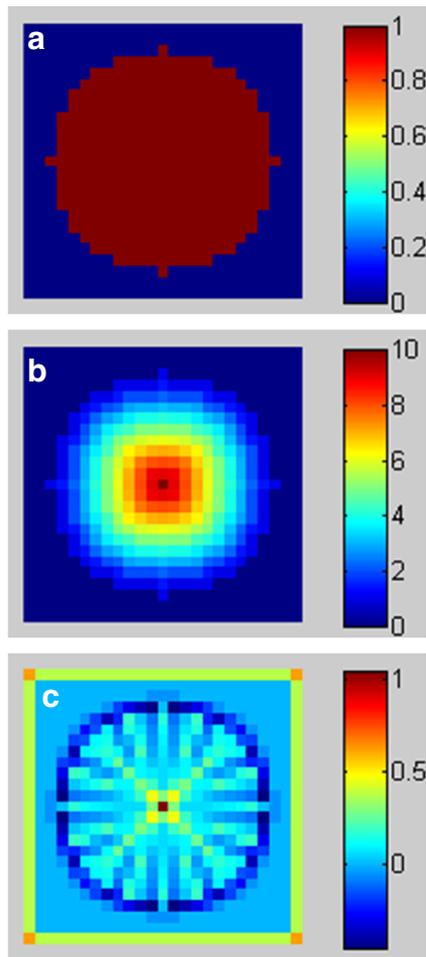
$$\left\{ \mathbf{v} : \frac{(A * d(I))(\mathbf{v})}{\max_{\mathbf{w} \in N(\mathbf{v})} d(\mathbf{w})} \geq \tau \right\},$$

where  $\tau = \frac{6}{13}$ . A graphical illustration of this filtering procedure is shown in Fig. 7.

As mentioned above, this analysis is accurate when the center of  $N(\mathbf{v})$  is on the centerline and all of  $N(\mathbf{v})$  is contained in the tubular structure. If this is not true, then it is not hard to find examples where  $(A * d(I))(\mathbf{v})$  is less than 0.5. One such example would be a cube with one of its sides parallel to the  $x$ -axis and cross-shaped cross section with the center of the “cross” belonging to the centerline. In this case  $(A * d(I))(\mathbf{v}) = 0.42$ . Therefore, for all practical purposes, we can set  $\tau = 0.5$  to obtain a first set of centerline candidates. We have obtained this threshold using the guidance offered by the previous analysis and painstaking experimentation. As one can see from Fig. 8, although  $\tau$  is marginally lower than the theoretically predicted  $\tau = \frac{6}{13}$ , it is still low enough to yield a large number of false positives. The lower  $\tau$  is, the higher rate of false seed candidates detections it becomes. But as  $\tau$  increases, so does the likelihood that no seed candidate is generated in certain regions of the segmented solid, with the consequence that these regions may be missed by the tracing routine, especially if their diameters are small. To mitigate these two problems we proceed as follows.

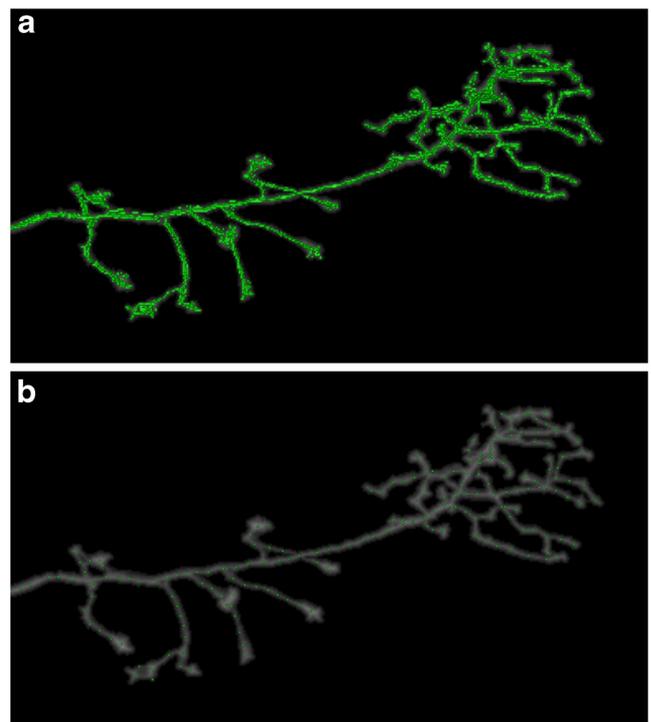
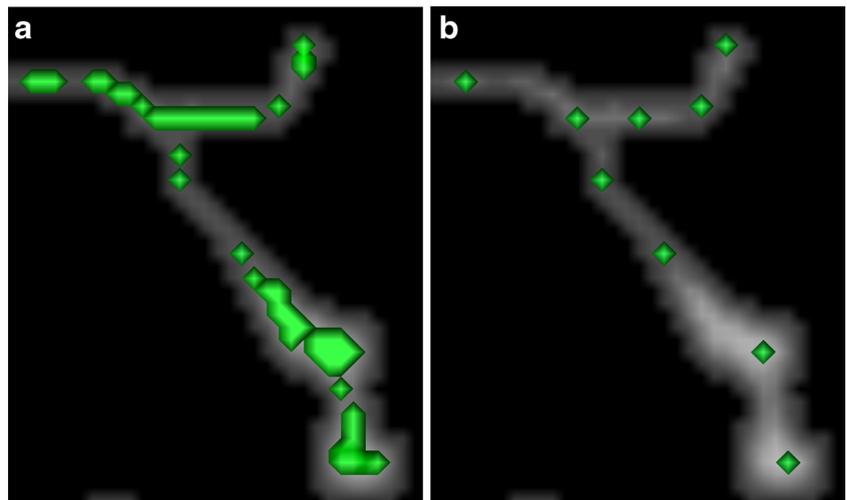
The next step in the centerline seed point selection is a *decimation processes*, aimed at eliminating false seed point detections (see Figs. 8 and 9). This pruning process is performed by Algorithm 1 under the guiding principle that, if  $\mathbf{v}$  is a centerline point then  $d(\mathbf{v})$  should be the maximum value of  $d$  on the 27-connected neighborhood  $N(\mathbf{v})$  of  $\mathbf{v}$ .

Suppose that  $\mathbf{v}$  is a seed candidate that is on the centerline and let  $\mathbf{w}_M \in N(\mathbf{v})$  maximizing  $d$  constrained on  $N(\mathbf{v})$ .



**Fig. 7** Detection of seed points. Panel **a**: Cross section of a representative segmented tubular structure. Panel **b**: The color-coded plot shows the values of the distance  $d$  computed at each point of the cross-section in panel (**a**). Panel **c**: The figure shows the result of the convolution of  $d$  with  $A$ , where  $d$  is given in panel (**a**). As expected,  $A * d$  attains its maximum at the center point of the cross-section

**Fig. 8** Process of seed generation. Panel **a**: Depiction of seed candidates obtained from the filtering process, prior to decimation. The green surface is an isosurface of the volume containing the detected seed candidates after the filtering  $A * d$  and thresholding with  $\tau$ . Not all voxels in the interior of the isosurface are used as seed candidates. Panel **b**: Seeds remaining after decimation



**Fig. 9** Seed Decimation. Panel **a**: The green surface is an isosurface of the volume containing the detected seed candidates in the Olfactory Projection fibers, image stack OP 1, from the DIADEM data set. Panel **b**: After our seed decimation process, only a small number of seeds remains to be used for the centerline tracing

Then, either  $w_M = v$  or both voxels are on the centerline. Recall, that if  $v$  is on the centerline, then two other voxels in  $N(v)$  must be on the centerline as well, but all three must return the same distance value which should be equal to the maximum value of  $d|_{N(v)}$  provided that the geometric properties of the tubular structure remain locally constant.

On the other hand, if  $d(v) < d(w_M)$  and  $N(v)$  contains an exterior voxel, then clearly  $v$  cannot be on the centerline

since one of its immediate neighbors belongs to the exterior of the structure and another voxel in  $N(\mathbf{v})$  maximizes  $d$ . If  $N(\mathbf{v})$  contains no exterior points and

$$\sqrt{d(\mathbf{v})^2 + 1} < d(\mathbf{w}_M) = \max\{d(\mathbf{w}) | \mathbf{w} \in N(\mathbf{v})\},$$

then  $\mathbf{v}$  is not a centerline voxel, for it were then, according to the previous discussion on the derivation of  $\tau$ , we must have that  $d$  constrained in  $N(\mathbf{v})$  attains its maximum in  $\mathbf{v}$  and  $d(\mathbf{v}) < d(\mathbf{w}_M)$  clearly contradicts that. This argument justifies the criterion of Algorithm 1 used for pruning the false centerline seed candidates. Does this mean that after running Algorithm 1 all remaining seed points are on the centerline? Theoretically, the answer to this question is still negative, but detections of seed points are now very safely on or close to the centerline. It is not easy to argue about accuracy because the tolerance level is reduced to one co-planar voxel as the previous argument indicates. The lack of cylindrical symmetry due to the physical shape of the structure and the binarization due to the digital acquisition may render the pursue of any higher accuracy rather futile. After all, the accuracy of the centerline detection can only be measured after the tracing step is completed. The sought seed points only serve as guiding landmarks for the tracing algorithm. In fact, we have observed that our centerline extraction method does not significantly suffer from false negatives obtained after Algorithm 1 has been applied, as these would, in most cases, be automatically corrected during tracing as we have already previously noted. Nevertheless, false positives that are significantly off the centerline are more likely to be left after Algorithm 1 has been applied only in thick tubular structures and those may indeed negatively affect the performance centerline extraction, as they may produce, in such cases, spurious branches or lead to the tracing of short, incorrect sections of the centerline. Our experiments indicate that this phenomenon is rather infrequent (see Fig. 10).

---

#### Algorithm 1 Seed Decimation

---

**Require:**

Voxel  $\mathbf{v}$  and set  $N(\mathbf{v})$ .

**Ensure:**

Decision  $S$ :  $\mathbf{v}$  is not a seed

- 1: Set  $d_M(\mathbf{v}) = \max\{d(\mathbf{w}) | \mathbf{w} \in N(\mathbf{v})\}$
  - 2: Set  $d_m(\mathbf{v}) = \min\{d(\mathbf{w}) | \mathbf{w} \in N(\mathbf{v})\}$
  - 3: **if**  $d_m(\mathbf{v}) = 0$  and  $d(\mathbf{v}) < d_M(\mathbf{v})$  **then**
  - 4:    $S = \mathbf{true}$
  - 5: **else if**  $\sqrt{d(\mathbf{v})^2 + 1} < d_M(\mathbf{v})$  **then**
  - 6:    $S = \mathbf{true}$
  - 7: **else**
  - 8:    $S = \mathbf{false}$
  - 9: **end if**
- 

It is also possible that no seeds were generated in some regions of the solid, creating the potential to miss part of the solid or even entire branches during tracing. To mitigate this effect, we identify these portions as the complement in



**Fig. 10** Centerline tracing in a thick tubular structure. The image shows the MIP projection of a segmented volume from the Olfactory Projection Fiber, Image Stack 3 (see the MIP of the entire segmented volume in Fig. 15, panel (d)). In this detail, we see an example of small spurious branches being traced in the “bowtie” shaped mid part of the volume. Note: the red line appears thicker because it represents the projection of the 3-D centerline on the MIP-plane which is 2-D

the solid of the set  $\{\mathbf{v} | \exists \mathbf{s} \in \mathcal{S}, \|\mathbf{v} - \mathbf{s}\| \leq C \cdot d(\mathbf{s})\}$  where  $C$  is twice the  $z$ -smear factor<sup>3</sup> of the input and  $\mathcal{S}$  is the current set of seeds. The new seeds are computed by iteratively computing the maximum value  $d(\mathbf{v})$  of the voxels in such regions, marking the voxel  $\mathbf{v}$  in question as a seed, and removing from such regions the voxels  $\mathbf{w}$  such that  $\|\mathbf{v} - \mathbf{w}\| \leq C \cdot d(\mathbf{v})$ . This process is significantly slower than the first seed selection step and, therefore, it is not used to generate seeds on the entire solid but only for a few small portions of the solid.

#### Tracing and Extraction of the Graph Structure of the Dendritic Arbor

Once the seeds are generated, the centerline is traced by iteratively identifying and connecting seeds to the current path, according to Algorithm 2.

Let  $S$  be the set of centerline seed points identified by Algorithm 1. The tracing algorithm (Algorithm 2) uses an inductive step described below to extract the centerline  $P$  of the tubular structure. First, we enumerate the seed points in  $S$ . The enumeration order does not matter. Suppose that the centerline path,  $P_{k-1}$ , connecting  $k - 1$  seed points has been extracted and that the next seed point to be connected to  $P_{k-1}$  is  $\mathbf{s}_k$ . Let  $R_k$  be a portion of the binarized structure containing  $\mathbf{s}_k$  in its interior starting from the closest point of the centerline that has been traced in the previous  $k - 1$  steps. Implicitly, we assume that  $R_k$  is in the same connected component with some part of the volume that has already been traced and it contains at least one point of  $P_{k-1}$ . To compute the new centerline subpath in  $R_k$  from this seed to  $P_{k-1}$ , we propose Algorithm 2, for the identification of

<sup>3</sup>This quantity represents that ratio of the length in the  $z$ -direction of a voxel relative to its length in the  $x, y$  directions. Therefore, it characterizes the sampling grid and thus it shows the anisotropy of the point-spread function in the  $z$ -direction.

the minimum length path between  $s_k$  and the most proximal voxel to  $s_k$  in  $P_{k-1} \cap R_k$ . This algorithm is a simple variation of Dijkstra’s algorithm (Dijkstra 1959). As we see next, from the description of the algorithm the sought path may not be the straight line connecting  $s_k$  to the already traced centerline, because such a path may not be inside  $P_{k-1} \cap R_k$ .

Our implementation differs from the standard Dijkstra’s algorithm, in that Algorithm 2 uses an ordered pair of weights  $W(e) = (W_1(e), W_2(e))$  for the directed edge  $e = (\mathbf{v}, \mathbf{w})$  between neighbouring voxels  $\mathbf{v}$  and  $\mathbf{w}$ , instead of a single weight used in the traditional Dijkstra’s algorithm.

The first of these two weights is defined by

$$W_1(e) = (d(\mathbf{v}))^{-1} + (d(\mathbf{w}))^{-1},$$

and it is introduced to maximize the distance from the path elements to the background. The progression step of the algorithm searches to identify the next voxel in the centerline starting from a given voxel  $\mathbf{v}$  in the following way: Between two choices  $\mathbf{u}$  and  $\mathbf{w}$  satisfying  $d(\mathbf{w}) > d(\mathbf{u})$ , then, by definition,  $\mathbf{w}$  is more likely to be a centerline voxel than  $\mathbf{u}$  is. In this case, it also follows that  $W_1(\mathbf{v}, \mathbf{w}) < W_1(\mathbf{v}, \mathbf{u})$ , and thus, the algorithm prefers  $\mathbf{w}$  over  $\mathbf{u}$ . Nevertheless, the effect of  $W_1$  is compromised if the cross section of the tubular structure is locally significantly anisotropic (this includes the case where the entire image has a significant  $z$ -smear). This necessitates the introduction of the second weight defined by

$$W_2(e) = 1 - \max \left\{ \frac{\langle \mathbf{v} - \mathbf{w}, \mathbf{w} - \mathbf{p} \rangle}{\|\mathbf{v} - \mathbf{w}\| \cdot \|\mathbf{w} - \mathbf{p}\|} \mid \mathbf{p} \in P_{k-1} \cap R_k \right\},$$

that is designed to ameliorate the effects of the anisotropy of the sampling grid by placing higher relevance to the

path with lower directional variation. Weight  $W_2$  maximizes the cosine of the angle between  $\mathbf{v} - \mathbf{w}$  and  $\mathbf{w} - \mathbf{p}$ , minimizing, therefore, the angle between the edges connecting the intermediate point  $\mathbf{w}$  and the starting point  $\mathbf{v}$ , on one hand, and  $\mathbf{w}$  with the candidate point  $\mathbf{p} \in P_{k-1} \cap R_k$  on the other. We apply the two weights conditionally with the second of them used only if we have cases where  $W_1(\mathbf{v}, \mathbf{w}) = W_1(\mathbf{v}, \mathbf{u})$ .

This process generates the minimum length subpath  $P_k$  connecting a voxel in  $P_{k-1} \cap R_k$  with  $s_k$ . The newly identified subpath  $P_k$  is added to the main path  $P$ , along with the information of the parent of each node. The seed  $s_k$ , and any other seed contained in the new subpath  $P_k$ , are now labeled as visited.

---

**Algorithm 2** Tracing

---

**Require:**

Seed set  $S$  and solid voxels  $B$ .

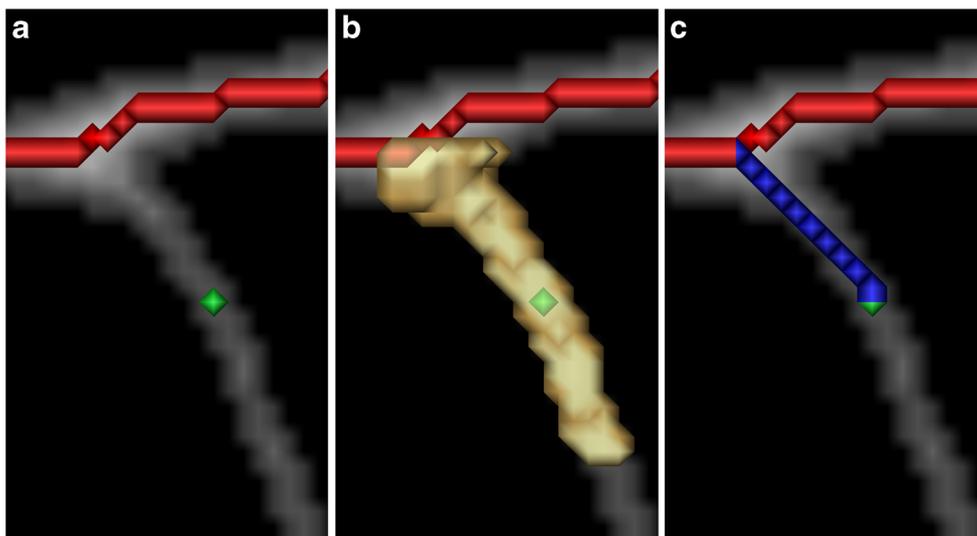
**Ensure:**

Centerline path voxel set  $P$ .

- 1: Select seed  $s_1$  so that  $d(s_1) = \max\{d(s) \mid s \in S\}$ .
  - 2:  $P \leftarrow \{s_1\}$ .
  - 3:  $k \leftarrow 1$ .
  - 4: **while**  $S \setminus P \neq \emptyset$  **do**
  - 5:      $k \leftarrow k + 1$ .
  - 6:     Select seed  $s_k$  so that  $d(s_k) = \max\{d(s) \mid s \in S \setminus P\}$ .
  - 7:     Set  $R_k = \{s_k\}$ .
  - 8:     **while**  $R_k \cap P = \emptyset$  **do**
  - 9:          $R_k \leftarrow (R_k \cup N(R_k)) \cap B$ .
  - 10:     **end while**
  - 11:      $P_k \leftarrow \text{DijkstraVariation}(s_k, R_k, P \cup R_k)$ .
  - 12:      $P \leftarrow P \cup P_k$ .
  - 13: **end while**
- 

Occasionally, some seeds that are not on the centerline remain after the decimation process, because they may have been associated with false branches. These falsely

**Fig. 11** Depiction of the tracing process. Following an iteration of the Algorithm 2, panel **a** shows in red a path  $P_{k-1}$  and the next seed  $s_k$  in green. In panel **b**, the region  $R_k$ , where the path search will be performed, is marked with yellow. This is the region inside which the subpath from the new seed to the current path will be identified. Panel **c** depicts in blue the subpath of  $P_k$  generated between  $P_{k-1}$  and  $s_k$



detected seeds are close to the real centerline, thus it is reasonable to expect that erroneously detected branches associated with these seeds are fairly short. To address this potential source of errors, we identify from the graph structure of the centerline, the terminal and branching points. We compute the length of the branch from the terminal point  $\mathbf{u}$  to its closest branching point  $\mathbf{v}$ . If this length is smaller than  $d(\mathbf{v})$ , we eliminate all the voxels in this branch but  $\mathbf{v}$ .

Algorithm 2 gives the centerline output in the form of a graph by encoding information about the coordinates of centerline voxels and of their connectivity. Each node in the graph of the centerline corresponds to a branching, initial, or terminal voxel and is annotated accordingly. An edge in this graph describes a segment of the centerline which connects a node to its immediate parent node (the last neighboring voxel to be added to the path before the voxel

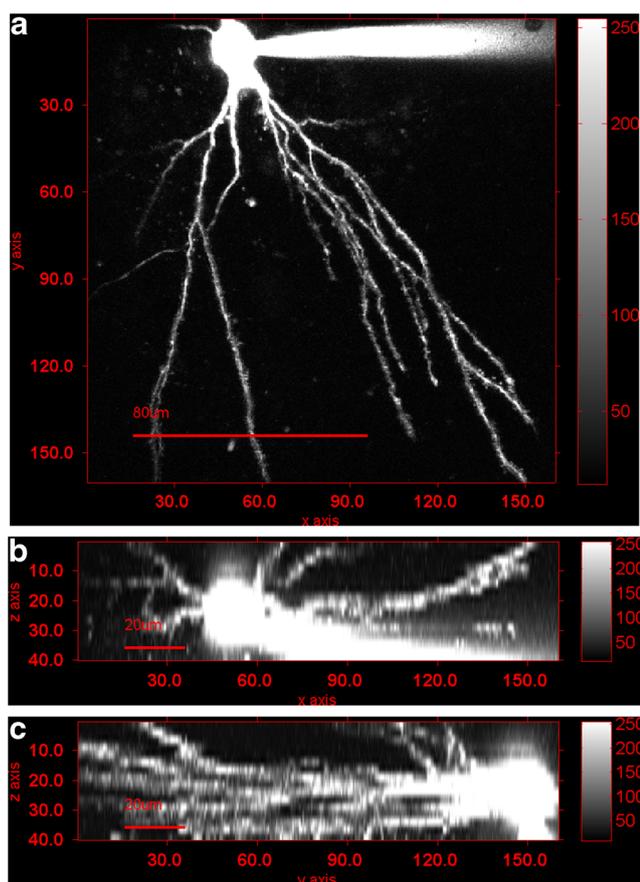
in question was added). The centerline tracing process is illustrated in Figs. 11 and 12.

## Experiments and Results

To evaluate the accuracy of the proposed method, we used the six image stacks of the Olfactory Projection Fibers (these are axons) datasets (only six sets were available at the time of paper submission), the first set of Neocortical Layer 1 Axons (see Fig. 5) and the first set of Neuromuscular Projection Fibers datasets from the DIADEM challenge (Brown et al. 2011). To compare our results, we relied on the Gold Standard Reconstructions provided by DIADEM. For each of these data sets, a single classifier was trained in one solid only. Our results on these sets are reported in Tables 1, 2 and Figs. 13, 14. Some visual examples are given in Fig. 15. A preview of a small number of these results appeared in Jiménez et al. (2013) that also included a brief description of the herein proposed methods which have been discussed in detail in Section “Methods”.

We also used a set of fluorescent images of pyramidal neuron cells that we indicate as Cells A, B and C. Cells A and B were acquired using a multiphoton microscope (the raw image of Cell A is shown in Fig. 12). The image quality of Cells A and B were ranked as good and fair, respectively. The image of Cell C was acquired with a confocal microscope and its quality was ranked as poor. Our results on this set are reported in Table 3.

Some of the Gold Standard Reconstructions provided for the Neuromuscular Projection Fibers by the DIADEM project do not have gold standard reconstructions for the entire volume appearing visually in the raw data. Hence, an estimate of the percentage of the solid traced was computed



**Fig. 12** MIPs of fluorescent images of a pyramidal cell. This image is used in our experiments whose results are shown in Tables 3 and 4 and labeled as Cell A. Panel **a**: MIP on the XY-plane; panel **b**: MIP on the XZ-plane; panel **c**: MIP on the YZ-plane (Courtesy of Professor P. Saggau of Baylor College of Medicine)

**Table 1** Performance metrics results for Olfactory Projection fiber stacks

Stack	Precision	Recall	MES	ADE ( $\sigma$ )	VRS
OP 1	1.00	1.00	1.00	0.71 (0.66)	0.97
OP 2	1.00	0.98	0.98	0.89 (1.02)	1.00
OP 3	0.85	0.88	0.82	1.31 (0.89)	0.85
OP 4	0.99	1.00	0.99	0.95 (1.07)	0.99
OP 5	0.79	0.94	0.81	1.37 (0.97)	0.88
OP 6	0.95	1.00	0.96	0.76 (1.08)	0.98
Average	0.93	0.97	0.93	1.0 (0.95)	0.87

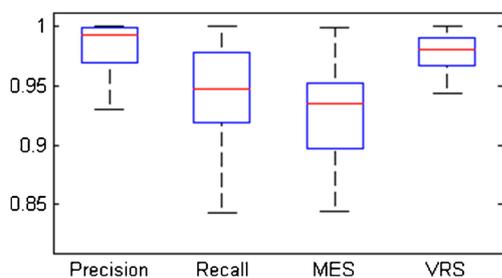
<sup>1</sup>In the ADE column we give sample averages and sample standard deviations of ADE for each stack measured in voxels

**Table 2** Performance metrics results for Neocortical Layer 1 Axon stacks (see Fig. 5)

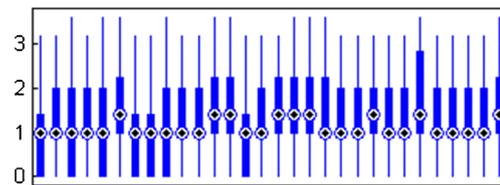
Stack	Precision	Recall	MES	ADE ( $\sigma$ )	VRS
1	0.90	0.81	0.78	0.79 (0.74)	0.89
2	0.92	0.60	0.61	1.14 (1.03)	0.88
3	0.74	0.60	0.56	1.28 (1.26)	0.76
4	0.62	0.75	0.52	1.00 (0.90)	0.86
5	0.82	0.92	0.77	0.96 (0.98)	0.93
6	0.79	0.86	0.71	0.91 (0.86)	0.92
Average	0.80	0.76	0.66	1.00 (0.97)	0.87

for each of the 152 stacks in this dataset by linearly interpolating first the expertly identified centerline points and by constructing a synthetic solid from the resulting piecewise linear approximations of the centerline. To construct these synthetic solids we used the radii of the tubular structure prescribed by the expert annotator at each of the manually identified centerline points. To avoid an inappropriate duplicate use of our segmentation algorithm at the stage where we identified the suitable stacks for benchmarking our centerline tracing algorithm, we compared the overlap of the crude synthetic solids with the solids segmented from the raw data by means of an ad hoc, user-selected intensity threshold. This simple test revealed that for only thirty solids out of the 152 stacks the manually identified centerline via straight line segments connecting manually traced centerline points overlap with at least 99% of the binary solid. Consequently, we limited our analysis to these stacks only.

For our binarization step, we considered nine features: three Laplacian filters, with  $\sigma \in \{0.25, 0.5, 0.75\}$ , three low pass band filters, with  $\sigma \in \{0.3, 0.5, 0.7\}$  and three high pass filters with  $(\sigma_1, \sigma_2) \in \{(0.4, 0.1), (0.6, 0.3), (0.8, 0.5)\}$ . We set  $n = 60$ , where  $n$  is the degree of the Taylor polynomial  $P_n$ .



**Fig. 13** Boxplot of the values of performance metrics for the thirty Neuromuscular Projection Fibers



**Fig. 14** Displacement-Error boxplots for the processed thirty Neuro-muscular Projection Fibers. Dots indicate Average DE, while the ends of the thick blue bar show the first and third quartiles of DE. The entire range of DE for each of the thirty stacks is marked by thin blue bars overlaid on the thick blue bars

To validate the performance of our algorithm, we used the metrics employed in Wang et al. (2011) and Xie et al. (2010). These include *Precision*, *Recall*, and the *Miss-Extra-Score* (MES) defined as

$$\text{Precision} = \frac{S_C}{S_T},$$

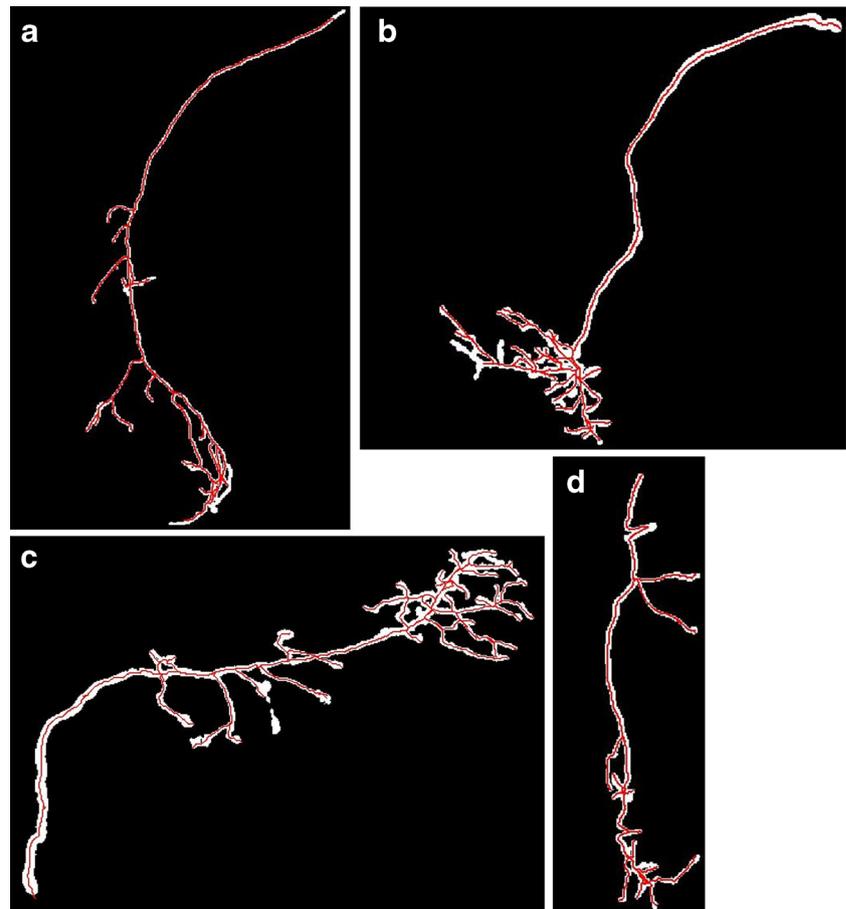
$$\text{Recall} = \frac{S_C}{S_C + S_{missed}},$$

$$\text{MES} = \frac{S_G - S_{miss}}{S_G + S_{extra}},$$

where  $S_C$  is the total length of the *correctly* traced segments,  $S_T$  is the total length of the traced centerline,  $S_G$  is the total length of all segments in the ground-truth trace,  $S_{miss}$  and  $S_{extra}$  are the total lengths of missing and extra segments in the computed trace, respectively. The second metric is the *Average-Displacement-Error* (ADE), which is defined as the average displacement of the matched components. The third and last metric to be computed is *Volume-Reached-Score* (VRS), defined as the ratio of the cardinalities of the set  $\{\mathbf{v} \in B | \exists \mathbf{c} \in C \text{ s.t. } \|\mathbf{v} - \mathbf{c}\| \leq d(\mathbf{c})\}$ , and of  $B$ , where  $B$  is the set of voxels in the binarized solid, and  $C$  is the set of voxels in the extracted centerline.

It is important to note that, for the reasons we previously explained, the traced and the gold standard centerlines may not match each other precisely at significant segments of the volume. The two centerlines may run parallel or be interwoven with one another. Nevertheless, they physically differ by a distance of a few voxels only. This error is reasonable because the traced paths are marked manually and automatically in a discretized volume and the manually produced centerline is a polygonal approximation of the true one. The degree to which this systematic error appears varies with acquisition resolution and the physical thickness of the dendrite. In fact, it is easier to manually mark with high accuracy the centerline of a thin branch rather than that of a thicker branch. To deal with this systematic error we

**Fig. 15** Centerline tracing superimposed on MIPs of binary segmentations of representative volumes from the DIADEM set processed with the proposed algorithm. Panel **a**: Olfactory Projection Fibers (axons), Image stack 2; panel **b**: Olfactory Projection Fibers (axons), Image stack 4; panel **c**: Olfactory Projection Fibers (axons), Image stack 1; panel **d**: Olfactory Projection Fibers (axons), Image stack 3



consider a voxel  $\mathbf{v} \in P_T$  to have been correctly traced if there is a voxel  $\mathbf{w} \in P_G$  such that  $\|\mathbf{v} - \mathbf{w}\| \leq L$ , where  $P_G$  the set of voxels in the centerline path in the gold standard annotations and  $P_T$  to be the set of voxels on the traced centerline and  $L$  is an operator determined tolerance constant. For the Olfactory Projection Fiber, the Neocortical Layer 1 Axons and the Pyramidal cells datasets, that include solids with mostly thin dendritic and axonal structures, we use  $L = 3$ . For the data sets of the Neuromuscular Projection Fibers where most branches were significantly thicker than those in the former datasets we use  $L = 7$ .

**Table 3** Performance metrics results for Pyramidal Cells

Cell	Precision	Recall	MES	ADE ( $\sigma$ )	VRS
A	0.92	0.85	0.92	1.96 (1.31)	0.99
B	0.91	0.85	0.89	2.94 (1.87)	0.97
C	0.75	0.89	0.61	2.88 (1.91)	0.95

Our results compare very favourably with the published literature. Specifically, (Xie et al. 2011) reported an average MES of 0.86 for a subset of five out of the six solids in Table 1. Our algorithm's average MES is 0.93 in the entire collection, and no less than 0.91 on any subcollection of five of these solids. Compared with the FARSIGHT algorithm (Wang et al. 2011), on a subset of 26 solids from the Neuromuscular Projection Fibers datasets the paper reports an average *Precision* of 0.98 and *Recall* of 0.95, after manual editing. From the subset analyzed in this paper, we obtain an

**Table 4** Performance metrics results for Pyramidal Cells, from ORION reconstructions

Cell	Precision	Recall	MES	ADE ( $\sigma$ )
A	0.88	0.85	0.78	2.29 (1.07)
B	0.87	0.77	0.71	3.40 (1.89)
C	0.77	0.58	0.57	2.75 (1.67)

**Table 5** Comparative Performance results of APP2, Neuronstudio, ORION and our Algorithm on the OP Dataset

Method	Precision	Recall	MES
Neuronstudio	0.97	0.81	0.79
APP2	0.73	0.88	0.71
ORION	0.93	0.92	0.86
Our algorithm	0.93	0.97	0.93

All values are sample averages of each algorithm's performance on eight of the nine OP volumes of the OP dataset. One of these volumes was not considered at all since ground truth was obtained for only one of the two dendrites present in this particular volume

average Precision of 0.98 and Recall of 0.94, with unedited traces and without additional training (Table 4).

From the comparison of ORION's performance with the performance of Neuronstudio by NeuroLucida, (NeuroLucida 2013) and APP2 (Xiao and Peng 2013) on the Olfactory datasets we can indirectly make a baseline comparison of the herein proposed algorithm with APP2, Neuronstudio and ORION. This comparison is reported in Table 5.

Processing times are reported in Table 6. All experiments were performed using a PC with a 2.79 GHz i5 quad-core processor and 16 GB of RAM memory, using MATLAB R2011b. We used the LIBSVM library as our Support Vector Machine engine.

## Discussion

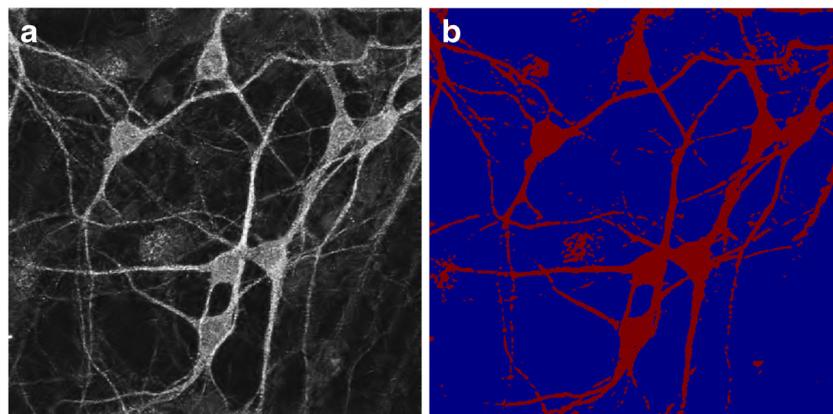
We have introduced a new algorithmic suite for the binary segmentation and centerline extraction of tubular structures (such as dendrites and axons of neurons). The proposed suite is competitive in terms of accuracy and computational efficiency. This excellent performance is achieved thanks to the integration of an effective filtering

**Table 6** Execution times in seconds for Binary Segmentation of volume of interest (B), Tracing (R) and Total (T)

Stack	B	R	T
OP1	54	17	71
OP2	77	26	103
OP3	61	9	70
OP4	60	18	78
OP5	76	8	84
OP6	88	13	101
Neocortical	314	3825	4139

stage to provide binarization and a simple and computationally efficient filter for point source detection. Although all of the applications of the proposed methods have been tested using laser microscopy images only, our methods are fairly generic and can be applied to images of tubular structures acquired with other modalities such as x-ray CT, MRI or Electron Microscopy both for segmentation and centerline tracing. Despite the fact that this claim cannot be supported with hard experimental evidence, the mathematical analysis we present in the methods section adequately justifies this claim, because the novel self-steerable filters require an image of a tubular structure, regardless of the acquisition modality, and the seed identification and centerline tracing are applied on the segmented volume of interest.

As shown in Tables 4 and 5, our algorithm consistently performs either better or equally well with its competitors, APP2, Neuronstudio and ORION, with respect to all three metrics, i.e., precision, recall and MES on the Olfactory Fiber Projection data sets. One of the advantages of the proposed centerline tracing algorithm is the automatic initialization from multiple seed points, each selected for just one connected component. The proposed algorithm extracts centerlines in multiple connected components of the same binarized volume, which frequently have more than one connected component. In contrast to other approaches assuming only a single connected component present in an image, our method *automatically* identifies the seed points in the entire volume, regardless of the number of the connected components in the entire data volume. The proposed method is easily implementable and does not require, yet it does not preclude, parallel processing. Last but not least, we wish to highlight the fact that the first part of our algorithm, which segments the dendritic arbor or axons from the background, works with high accuracy, as the VRS metric indicates. In this often difficult task, this kind of performance is essential for the success of our centerline detection method, hence we consider it to be one of the main contributions of this work. We want to emphasize that this part our algorithm has broad applicability. Although, the mathematical proofs indicate that the self-steerable filters are applicable to tubular structures, we have verified in practice that the Laplacian filters retain their self-steerability properties even when they are applied on images with more isotropic structures, like somas. Fig. 16 shows an example of application of our segmentation algorithm to a confocal image of a neuronal culture. Although we have not systematically studied this behavior of our filters beyond tubular structures, the accuracy of our segmentation results indicates that there the Laplacian filters are effective to detect structure that are not necessarily vessel-like. This is an indirect indication that they retain their self-steerability



**Fig. 16** Panel **a**: Maximum projection of a confocal image of primary hippocampal neurons labeled with anti-microtubule associated protein 2 (MAP2) antibody, visualized with an Alexa 647-conjugated secondary antibody. The image was acquired using a Zeiss LSM-510 Meta confocal microscope with an excitation lines at 633 nm and an emission filter low-pass 650 (courtesy of Courtesy of Prof. F. Laezza

of the University of Texas, Medical Branch in Galveston). The denoising algorithm, (Easley et al. 2008; Negi and Labate 2012), uses the shearlet transform in 2-D as a preprocessing step. Panel **b**: The segmented volume of interest using the segmentation method with feature extraction based on self-steerable Laplacian and shearlet filters, both adopted for 2-D

property. It is this property that makes the Laplacian filters act as singularity detectors at the direction of the image intensity gradient, hence to act as omnidirectional boundary detectors. We attribute to this property the fact that Laplacian filters are able to successfully segment both somas and dendritic arbors in images of neuronal cultures. Note that these images are essentially 2-dimensional because of the small extension of the z-axis (typically only about 20 pixels are available along the z coordinate). We found that, in this situation, to compensate for this limitation it can be useful to include a set 2D-shearlet filters, a class of directional multiscale filters (Easley et al. 2008), to derive additional features.

We would like to close with a comment with regards to the need of automatic or semi automatic image analysis tools for neuroscience imaging. Everyone who has worked with fluorescent or electron microscopy images of neurons has observed that often manual annotations lack accuracy and consistency. One such example is the neuromuscular fibers of the DIADEM competition data sets. A good number of the image stacks of this data set are under-traced. From this and many other examples is apparent that there is a need of improved tools for image segmentation and image analysis that are capable to handle this type of data. We hope that our work will be a humble contribution to this challenging field of investigation.

### Information Sharing Statement

Matlab source code of all binary segmentation, seeding and tracing routines are publicly available for download at <http://www.math.uh.edu/~mpapadak/centerline>.

A manual is included in the same folder together with the source code. The code works for 3-D data sets, but it can be modified for use with 2-D sets. Please cite the above web address if you use our code for any publication or commercial purpose. All data sets used in our experiments are publicly available at the DIADEM competition (RRID:nif-0000-23194) website.

**Acknowledgments** This work was supported in part by NSF-DMS 1320910, 0915242, 1008900, 1005799 and by NHARP-003652-0136-2009. I.A. Kakadiaris was also supported by the Hugh Roy and Lillie Cranz Cullen Endowment Fund. We thank our student P. Hernandez-Herrera for allowing us to use the performance test results he obtained with Neuronstudio, ORION and APP2 on numerous data sets. Last, but most certainly not least, we would like to thank the three reviewers for their encouraging comments and their valuable input which helped us improve the quality of the paper.

### References

- 3D-Slicer (2008). <http://www.slicer.org/>.
- Al-Kofahi, K., Lasek, S., Szarowski, D., Pace, C., Nagy, G. (2002). Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Trans Information Technology in Biomedicine*, 6(2), 171–187.
- Bas, E., & Erdogmus, D. (2011). Principal curves as skeletons of tubular objects. *Neuroinformatics*, 9(2-3), 181–191.
- Bodmann, B., Hoffman, D., Kouri, D., Papadakis, M. (2007). Hermite distributed approximating functionals as almost-ideal low-pass filters. *Sampling Theory in Image and Signal Processing*, 7, 15–38.
- Breitenreicher, D., Sofka, M., Britzen, S., Zhou, S. (2013). Hierarchical discriminative framework for detecting tubular structures in 3d images. In J. Gee (Ed.) *Proceedings of the Information Processing in Medical Imaging. Lecture Notes in computer Science* (Vol. 7917, pp. 328–339). Berlin Heidelberg: Springer-Verlag.

- Brown, K., Barrionuevo, G., Canty, A., Paola, V., Hirsch, J., Jefferis, G., Lu, J., Snippe, M., Sugihara, I., Ascoli, G. (2011). The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9(2-3), 143–157.
- Chandler, C., & Gibson, A.G. (1999). Uniform approximation of functions with discrete approximation functionals. *J Approx Th*, 100, 233–250.
- Chothani, P., Mehta, V., Stepanyantas, A. (2011). Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, 9(2-3), 263–278.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.
- Dijkstra, E. (1959). A note on two problems in connection with graphs. *Numerische Mathematic*, 1, 269–271.
- Donohue, D., & Ascoli, G. (2010). Automated reconstruction of neuronal morphology: an Overview. *Brain Research Reviews*.
- Easley, G., Labate, D., Lim, W.Q. (2008). Sparse directional image representations using the discrete shearlet transform. *Applied and Computational Harmonic Analysis*, 25(1), 25–46.
- Frangi, A., Niessen, W., Vincken, K., Viergever, M. (1998). Multiscale vessel enhancement filtering. In *Proceedings Medical Image Computing and Computer Assisted Intervention* (Vol. 1496 pp. 130–137). MA: Cambridge.
- Hassouna, M., & Farag, A. (2005). Robust centerline extraction framework using level sets. In *IEEE Proceedings Conference on Computer Vision and Pattern Recognition* (Vol. 1 pp. 458–465).
- Hines, M., & Carnevale, N. (2001). NEURON: a tool for neuroscientists. *The Neuroscientist*, 7, 123–135.
- Hoffman, D.K., & Nayar, N. (1991). Analytic banded approximation for the discretized free propagator. *Journal of Physical Chemistry*, 95, 8299–8305.
- Hoffman, D.K., Frishman, A.M., Kouri, D.J. (1996). Distributed approximating functional approach to fitting multi-dimensional surfaces. *Chemical Physics Letters*, 262, 393–399.
- Hoffman, D.K., Kouri, D.J., Pollak, E. (2002). Reducing gaussian noise using distributed approximating functionals. *Computer Physics Communications*, 147, 759–769.
- Jiménez, D., Papadakis, M., Labate, D., Kakadiaris, I.A. (2013). Improved automatic centerline tracing for dendritic structures. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* (pp. 1050–1053): IEEE.
- Jimenez, D., & Labate, D. (2014). Papadakis M A directional representation for 3D tubular structures resulting from isotropic well-localized atoms under review.
- Kakadiaris, I., & Colbert, C. (2007). Orion: Automated reconstruction of neuronal morphologies from image stacks. In *Proceedings 24<sup>th</sup> Annual Houston Conference on Biomedical Engineering Research, Houston* (p. 275).
- Koh, Y. (2001). Automated recognition algorithms for neural studies. PhD thesis, State University of New York at Stony Brook.
- Krissian, K., Malandain, G., Ayache, N., Vaillant, R., Troussset, Y. (2000). Model based detection of tubular structures in 3d images. *Computer Vision and Image Understanding*, 80(2), 130–171.
- Luisi, J., Narayanaswamy, A., Galbreath, Z., Roysam, B. (2011). The FARSIGHT trace editor: an open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions. *Neuroinformatics*, 9(2-3), 305–315.
- Meijering, E. (2010). Neuron tracing in perspective. *Cytometry Part A*, 77A(7), 693–704. doi:10.1002/cyto.a.20895.
- Morrison, P., & Zou, J. (2006). Skeletonization based on error reduction. *Pattern Recognition*, 39(6), 1099–1109.
- Negi, P., & Labate, D. (2012). 3D discrete shearlet transform and video processing. *IEEE Transactions Image Process*, 21(6), 2944–2954. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6129426](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6129426).
- NeuroLucida (2013). MBF Bioscience: stereology and neuron morphology quantitative analysis. URL <http://www.mbfbioscience.com>.
- Ozcan, B., Labate, D., Jiménez, D., Papadakis, M. (2013). Proceedings Wavelets and Sparsity XV. In D Van De Ville MP V Goyal (Ed.), Vol. 8858. San Diego: SPIE.
- Peng, H., Long, F., Myers, G. (2011). Automatic 3D neuron tracing using all-path pruning. *Bioinformatics*, 27(13), i239.
- Rodriguez, A., Ehlenberger, D., Hof, P., Wearne, S. (2009). Three-dimensional neuron tracing by voxel scooping. *Journal of Neuroscience Methods*, 184(1), 169–175. doi:10.1016/j.jneumeth.2009.07.021.
- Santamaria-Pang, A., Colbert, C., Losavio, B., Saggau, P., Kakadiaris, I. (2007). Automatic morphological reconstruction of neurons from optical images. In *Proceedings International Workshop in Microscopic Image Analysis and Applications in Biology*. NJ: Piscataway.
- Santamaria-Pang, A., Hernandez-Herrera, P., M P P S, Kakadiaris, I.A. (2014). Automatic morphological reconstruction of neurons from multiphoton and confocal microscopy images. *Neuroinformatics*.
- Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., Kikinis, R. (1998). 3-D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2), 143–168.
- Sato, Y., Westin, C., Bhalerao, A., Nakajima, S., Shiraga, N., Tamura, S., Kikinis, R. (2000). Tissue classification based on 3d local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2), 160–180.
- Scorcioni, R.A.G., & Polavaram, S. (2008). L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature Protocols*, 3(5), 866–876. doi:10.1038/nprot.2008.51.
- Turetken, E., Gonzalez, G., Blum, C., Fua, P. (2011). Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9(2-3), 279–302.
- Turetken, E., Benmansour, F., Fua, P. (2012). Automated reconstruction of tree structures using path classifiers and mixed integer programming. In *IEEE Conference on Proceedings Computer Vision and Pattern Recognition (CVPR)* (pp. 566–573). Rhode Island: IEEE.
- Turetken, E., Benmansour, F., Andres, B., H P, Fua, P. (2013). Reconstructing loopy curvilinear structures using integer programming. In *Proceedings CVPR* (pp. 1822–1829). Portland: IEEE.
- Wang, Y., Narayanaswamy, A., Tsai, C.L., Roysam, B. (2011). A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, 9(2-3), 193–217.
- Wearne, S., Rodriguez, 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. doi:10.1016/j.neuroscience.2005.05.053.
- Xiao, H., & Peng, H. (2013). APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11), 1448–1454.
- Xie, J., Zhao, T., Lee, T., Myers, E., Peng, H. (2010). Automatic neuron tracing in volumetric microscopy images with anisotropic path searching. In *Proceedings Medical Image Computing and Computer-Assisted Intervention* (pp. 472–479). Beijing: Springer.
- Xie, J., Zhao, T., Lee, T., Myers, E., Peng, H. (2011). Anisotropic path searching for automatic neuron reconstruction. *Medical image analysis*, 15(5), 680–689.
- Xiong, G., Zhou, X., Degterev, A., Ji, L., Wong, S. (2006). Automated neurite labeling and analysis in fluorescence microscopy images. *Journal of the International Society for Analytical Cytology*, 69A, 494–505. doi:10.1002/cyto.a.20296/pdf, <http://onlinelibrary.wiley.com.ezproxy.lib.uh.edu>.

- 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.
- Zhao, T., Xie, J., Amat, F., Clack, F., Ahammad, P., Peng, H., Long, F., Myers, E. (2011). Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, 9(2-3), 247–261.
- Zhou, Y., & Toga, A. (1999). Efficient skeletonization of volumetric objects. *IEEE Transactions on Visual and Computer Graphics*, 5(3), 196–209.