

# Tubularity Flow Field—A Technique for Automatic Neuron Segmentation

Suvadip Mukherjee, *Student Member, IEEE*, Barry Condron, and Scott T. Acton, *Fellow, IEEE*

**Abstract**—A segmentation framework is proposed to trace neurons from confocal microscopy images. With an increasing demand for high throughput neuronal image analysis, we propose an automated scheme to perform segmentation in a variational framework. Our segmentation technique, called tubularity flow field (TuFF) performs directional regional growing guided by the direction of tubularity of the neurites. We further address the problem of sporadic signal variation in confocal microscopy by designing a local attraction force field, which is able to bridge the gaps between local neurite fragments, even in the case of complete signal loss. Segmentation is performed in an integrated fashion by incorporating the directional region growing and the attraction force-based motion in a single framework using level sets. This segmentation is accomplished without manual seed point selection; it is automated. The performance of TuFF is demonstrated over a set of 2D and 3D confocal microscopy images where we report an improvement of >75% in terms of mean absolute error over three extensively used neuron segmentation algorithms. Two novel features of the variational solution, the evolution force and the attraction force, hold promise as contributions that can be employed in a number of image analysis applications.

**Index Terms**—Confocal microscopy, neuron tracing, level set, vector field convolution.

## I. INTRODUCTION

**S**HAPe based neuron morphology analysis provides important cues in deciphering several functional behaviors of the brain of an individual [1]. Neuronal morphology has been studied to develop a functional model [2] for that neuron category, to analyze the branch patterns of serotonergic neurons [3], [4], or to correlate the structural aberrations in the dendritic arbors of an organism due to genetic factors or degenerative diseases like Alzheimer's [5].

An extensive shape based study of neuron morphology for an organism requires a comprehensive collection of digitally reconstructed neurons [6], which in turn demands intelligent processing tools to reconstruct neurons from the raw microscopy data. Recent advances in microscopy has enabled

Manuscript received June 6, 2014; revised September 22, 2014; accepted November 17, 2014. This work was supported by the National Science Foundation under Grant 1062433. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Olivier Bernard.

S. Mukherjee and S. T. Acton are with the C. L. Brown Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904 USA (e-mail: suvadip21@gmail.com; acton@virginia.edu).

B. Condron is with the Department of Biology, University of Virginia, Charlottesville, VA 22904 USA (e-mail: condron@virginia.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2378052

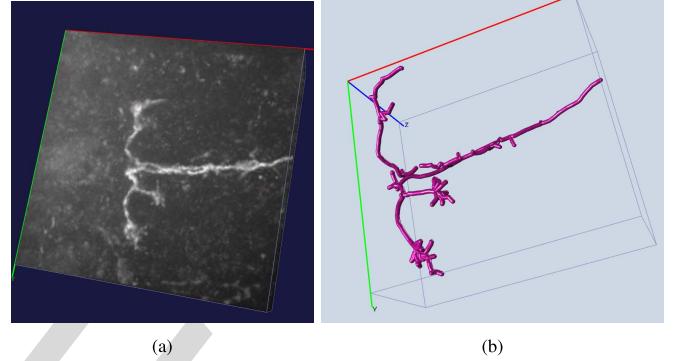


Fig. 1. (a) A *Drosophila* neuron imaged by confocal microscope. The background clutter is due to illuminated non neuronal filaments. (b) The corresponding reconstruction is shown.

imaging scientists to acquire substantial quantity of images. With more than 20,000 neurons in the brain of the fruit fly *Drosophila* and even more for other species such as mice and humans, the task of automated, high throughput neuro-image analysis is both critical and daunting.

Given the complexity of the problem, it is not surprising that automated neuron segmentation still remains a critical open problem in the field. State of the art neuron segmentation methods rely heavily on manual interaction to generate the morphological reconstruction. Complicated branching patterns of the neurons pose challenge to automated tracing. Moreover, the confocal microscopy images are, in general, degraded by low signal to noise ratio and non uniform illumination of the neurites which leads to fragmented appearance of the object. Fig. 1(a) shows a 3D neuron image of *Drosophila* imaged using a laser scanning confocal microscope. Topologically, a neuron resembles a tree, with multiple filamentous branches emerging from a single cell body. This is shown in Fig. 1(b), which is a digital reconstruction of (a), obtained using our algorithm. In this paper, we present an automated neuron segmentation method, based on an energy minimization framework. Segmentation results on GFP-labeled *Drosophila* neurons, imaged using confocal microscope are studied to demonstrate the efficacy of our technique.

### A. Background

In this section we briefly review some relevant research in neuron segmentation. In this paper, we are interested in segmenting neurons from confocal microscopy images only. Therefore, techniques which use other imaging modalities

(such as electron microscopy) are excluded from this discussion.

We can broadly categorize the neuron segmentation schemes in two basic approaches. The first set of methods use user defined (or automatically detected) initial seed points to perform tracing. The second category of algorithms avoid seed initialization and perform segmentation globally.

Manual seed selection has the advantage that the segmentation region is identified a priori by an expert. This introduces locality in processing, which results in higher processing speed. Typically such algorithms generate the neuronal tree from semi-automatically initialized seed points on the neurite centerlines. Al-Kofahi *et al.* [7] used the medial response of multiple directional templates to determine the direction to generate successive seed points along the neuron medial axis. This local tracing method shows good performance in high-contrast images, but requires continuity in the neuron branches for reliable segmentation.

Segmentation performance can be considerably improved if the seed points are selected manually. These seeds are then treated as nodes in a graph, and segmentation is performed using graph theoretic algorithms. When seed selection is done automatically, a pruning step is generally used to eliminate the non-neuronal points. With this optimal set of seeds, the methods in [8]–[10] establish connectivity between the nodes using a shortest path algorithm [11], by suitably selecting the weights on the graph edges. Fast and accurate segmentation is possible using the above mentioned approaches if the neuron structure is morphologically simple and the image noise level is low. Gonzalez *et al.* [12] introduced a graph theoretic technique to delineate the optimal neuronal tree from an initial set of seeds by computing a K-Minimum Spanning Tree. An approximate solution to this NP-hard problem was realized by minimizing a global energy function in a linear integer programming framework. However, due to its greedy nature, the algorithm may converge to undesired local minima.

We hypothesize that seed based techniques are useful if the imaged neurons are not too complicated structurally. In such scenarios, where manual seed selection is easy, reliable segmentation can be obtained. However, since automatically choosing the correct set of seed points is still an open problem, it is difficult to use the above mentioned techniques for high throughput, no intervention analysis. Also, since proper selection of seeds points is instrumental in these methods, the segmentation accuracy is sometimes compromised if a sub-optimal set of points is chosen. Furthermore, the connectivity analysis between the seeds assume uniform signal intensity, and noise and low contrast in the images may degrade the segmentation quality.

In contrast to the seed based local techniques, traditional segmentation approaches are more global, typically requiring an initial pre-processing of the image followed by a specialized segmentation step. Although a global approach may suffer from expensive computation, they are more suitable for neurite junction and end point detection. Typically, such methods rely on a four stage processing pipeline – enhancement, segmentation, centerline detection and post processing. The voxel scooping algorithm proposed in [13] assumes

tubular structure of the neurite filaments and iteratively searches for voxel clusters in a manner similar to region growing. A pruning step is then deployed to eliminate spurious end nodes. A similar region growing method is implemented in the popular automatic neuron tracing tool Neuronstudio [14]. The segmentation step is generally followed by a centerline detection [2], [15] stage to detect the medial axis of the segmented structure. In many cases further smoothing of the medial axis is performed by spline fitting [16]. Since such methods do not rely on human intervention, it is evident that the segmentation quality would depend heavily on the initial segmentation, which may be affected by the noise and clutter in the images.

Tree2Tree [16] and its variants [17], [18] propose to solve the neuron segmentation problem in a graph theoretic framework. However, unlike traditional seed selection approaches, where manually initialized points are treated as the nodes of the graph, an initial segmentation algorithm is devised to produce disjoint connected components. Connectivity between the components is analyzed based on their separating distance and orientation, which determines the weights of the graph edges to perform segmentation using a minimum spanning tree approach.

Although the primary contribution of Tree2Tree is to connect the fragmented neurite segments automatically, this connectivity analysis relies on heavily on the initialization. Noise and clutter in the images create undesired artifacts in the global segmentation, resulting in loss of structural information. Moreover, linking the components based on their relative geometric orientation requires computation of the leaf-tangents from the object centerlines, which is sensitive to the irregularities of the neurite surface. Furthermore, elimination of false nodes from the neuronal tree is difficult, and ultimately requires further manual parameter tuning.

Segmentation based on active contours [19] have also been proposed [20], [21] to directly obtain the neuron centerline, without performing a global thresholding. The algorithm proposed by Wang *et al.* [20] involves evolution of an open ended snake guided by a force field that encourages the neuron trace to lie along the filament centerline. A pre-processing step based on tensor voting [22] was introduced to enhance the vascular structure of the neurites. Combined with a post-processing step to eliminate false filaments, this method is efficient in segmenting neuronal structures from low SNR confocal stacks. However, due to the inability of parametric active contours to naturally handle topological changes such as object merging, neurite branch point detection depends requires a non-trivial post processing to determine snake merging at the junctions. Santamaría-Pang *et al.* [23] use a multistage procedure for detection of tubular structures in multi-photon imagery, which includes a pre-filtering stage to identify the filaments based on supervised learning. This requires offline learning of the model parameters and prior knowledge about the vessel appearance information, which necessitates a set of accurate training examples and demands extensive human involvement to generate the ground truth. Zhou *et al.* [24] propose a variational framework based on geodesic active contours to identify neurite branches from

two photon microscopy. This strategy is effective when the edge information is reliable, and hence depends on efficient pre-processing to eliminate image irregularities. However, both these methods do not deploy additional schemes to identify and analyze the broken neurite fragments in their model, and hence it demands a specialized post-processing step.

The medical imaging community has performed substantial research in developing algorithms to detect and segment filamentous shapes in non-microscopy medical images [25]. The CURVES algorithm by Lorigo *et al.* [26] evolves a 1D curve along a 3D vessel centerline guided by the curvature of a 1D curve.

Gooya *et al.* [27] developed an elegant and generalizable regularization methodology to enhance the performance of the popular geometric curve evolution methods. The method allows for anisotropic curve propagation which minimizes contour leakage when vessel edge information is weak. The only apparent downside of this technique is that the ultimate solution somewhat depends on the shape of the initialized contour. Another recent work by Gooya *et al.* [28] generalizes the flux maximizing flow [29] on Riemann manifolds and uses a vessel enhancing tensor, which improves segmentation when edge information is noisy.

Shang *et al.* [30] propose a vessel tracing method where wider vessels are first segmented using a region based criteria. Then the eigenvectors of the hessian matrix are utilized to derive a geometric flow equation to segment the thinner vessels. The mathematical formulation of the problem involves only a single eigenvector (the one along axial direction of vessel) for curve evolution, and hence is unsuitable for detecting thicker vessels. As we will show later, our formulation presents an unified framework to segment vessels of heterogeneous thickness by utilizing information from all three principal vessel directions (axial and orthogonal). Also, since the above mentioned methods are tailored for applications such as MRA, CT etc, they require further modifications to satisfy the demands of confocal microscopy where noise and clutter is present in a significantly higher proportion.

### B. Our Contribution

We focus on reconstructing single neuron from a confocal microscope image. A robust neuron segmentation scheme needs to address two primary issues. First, the technique should be suited to identify neuron structures from the noisy confocal images. Second, it should be adept at handling the local structure discontinuities (see Fig. 2) resulting from imaging artifacts. We propose a solution to this segmentation problem using a variational framework driven by level sets. The level set evolution is guided by minimizing an application specific energy functional. A tubularity flow field (TuFF) is computed by utilizing the local tubularity of the neurites which guides the segmentation procedure by encouraging curve evolution along the length (axis) and the thickness of the tubular neurites. A specialized local attraction force is also designed to accommodate the intensity variations in the images of neurite structures, thus presenting an unified framework to naturally link the fragmented structures. Our method does

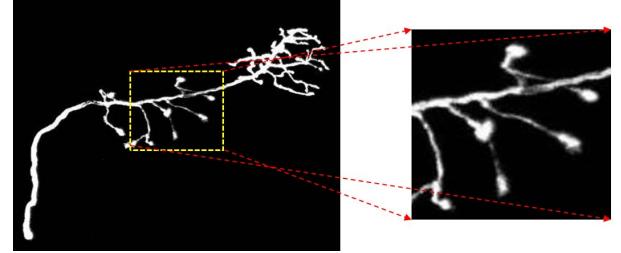


Fig. 2. Maximum intensity projection of a neuron imaged by a confocal microscope. The image suffers from contrast non-uniformity, including gaps that lead to breaks in the segmented neurite structure. The effect is most pronounced in the region bounded by yellow dashed box, magnified here for improved viewing.

not rely on an initial set of seed-points for segmentation; it is automatic. Moreover, it does not require non-trivial post-segmentation analysis to link the disjoint segments. This is performed naturally by using the local attraction force in a level set paradigm. This enables us to connect disunited structures, even if the underlying signal intensity is significantly low. The problem formulation and the design process of the attraction force are discussed in the following sections.

## II. TUBULARITY FLOW FIELD FOR NEURON SEGMENTATION

Active contours or snakes [19], [31]–[34] are an attractive choice for image segmentation due to their ability to elastically delineate object boundaries with sub-pixel accuracy and to incorporate signal and shape based constraints to assist segmentation. Geometric active contours [24], [33]–[37] are appealing due to their inherent ability to deal with topological changes of the foreground in segmentation. Unlike their parametric counterparts which perform segmentation by explicitly updating the position of a parametric curve, geometric active contours perform curve evolution implicitly, by evolving a higher dimensional embedding function  $\phi$ .

Let  $f : \Omega \rightarrow \mathbb{R}$  be an image defined on the continuous domain  $\Omega \subset \mathbb{R}^d$ , where  $d$  is the dimension of the image. In a variational paradigm, implicit motion of the zero level set of  $\phi$  is obtained by minimizing an energy functional  $\mathcal{E}(\phi)$  [24], [36]–[39]. The level set function  $\phi$  is defined to be positive inside the zero level set and negative outside it. The zero level sets define the object boundaries. The energy functional design is application dependent, and is a major engineering aspect for all variational level set based methods. Such methods are popular since the energy functional gives intuition for the segmentation procedure. Furthermore, various shape and smoothness constraints can be easily incorporated to further assist segmentation [33], [40]. For this problem of neuron segmentation, we need to design the energy functional such that it would encourage curve propagation in the filamentous regions of the image, while avoiding the non-tubular structures. Also, the segmentation should allow sufficient local processing to avert fragmented segments in the solution, which may appear as a consequence of using global threshold selection schemes like that of Otsu [41] or

methods assuming piecewise constant intensity models of [36]. We avoid this problem by introducing a local shape prior by way of a specially designed tubularity flow vector field and a local attraction force to link nearby neuronal fragments.

### A. Tubularity Flow Field (TuFF)

As mentioned previously, we assume a locally tubular model for neurite segmentation. The key ingredient of our algorithm is to use this tubularity information to evolve the level set function. A set of vector fields called the tubularity flow field (TuFF) is used to drive the active contour towards the object boundary. The tubularity measure at a point  $\mathbf{x} \in \Omega$  in the image can be obtained by examining the hessian matrix of the gaussian smoothed image over a set of scales. The hessian of the  $d$ -dimensional image  $f(\mathbf{x})$  at a position  $\mathbf{x}$  and scale  $\sigma$  is the square matrix  $H_\sigma(\mathbf{x}) = [h]_{i,j}$  ( $1 \leq i, j \leq d$ ,  $\mathbf{x} \in \Omega$ ) which is given by

$$h_{i,j} = \frac{\partial^2 G(\sigma)}{\partial x_i \partial x_j} * f(\mathbf{x}) \quad (1)$$

where  $\mathbf{x}$  is the  $d$ -dimensional vector  $\mathbf{x} = (x_1, \dots, x_d)^T$ ,  $G(\sigma)$  is the zero mean normalized Gaussian kernel with variance  $\sigma^2$ . Here  $d = 2$  or  $3$  for 2D or 3D images respectively.

Since the imaged neurons are brighter than the background, one can analyze the scale space hessian matrix to obtain evidence of tubularity at a particular image position. Ideally, at a position  $\mathbf{x} \in \Omega$ , 3D tubular structure can be characterized by three principal directions: (i) an axial direction along which the second derivative is negligible, and (ii) two orthogonal directions along which the second derivative magnitude is significant. These directions are given by the orthonormal set of eigenvectors  $\{\mathbf{e}_1(\mathbf{x}), \mathbf{e}_2(\mathbf{x}), \mathbf{e}_3(\mathbf{x})\}$ . The corresponding second derivative magnitudes can be obtained from the respective eigenvalues  $|\lambda_1(\mathbf{x})| \leq |\lambda_2(\mathbf{x})| \leq |\lambda_3(\mathbf{x})|$ .

Analysis of these eigenvalues is essential to preserve the tubular portions of neurons, while suppressing the background clutter [16], [42]. The non tubular clutter are present in most confocal microscopy images due to photon emission from non neuronal tissues and are often referred to as *structure noise*. These structure noise may be bright disc shaped non-neuronal segments in 3D images or blob-like structures. We would like to mention that from here onward we would present our analysis for the 3D case only for better readability. However, the results are easily applicable to the 2D case and there exists an equivalent 2D version of the solutions.

It may be observed that for a voxel  $\mathbf{x}$  to belong to a tube, the eigenvalues of its hessian matrix (computed at scale  $\sigma$ ) should satisfy the following criteria:

$$\begin{aligned} |\lambda_1(\mathbf{x})| &\approx 0 \\ |\lambda_2(\mathbf{x})| &\gg |\lambda_1(\mathbf{x})|, |\lambda_3(\mathbf{x})| \gg |\lambda_1(\mathbf{x})| \\ |\lambda_2(\mathbf{x})| &\approx |\lambda_3(\mathbf{x})| \end{aligned} \quad (2)$$

Also, since the neurites are brighter than the background, we have  $\lambda_2(\mathbf{x}) < 0$  and  $\lambda_3(\mathbf{x}) < 0$ .

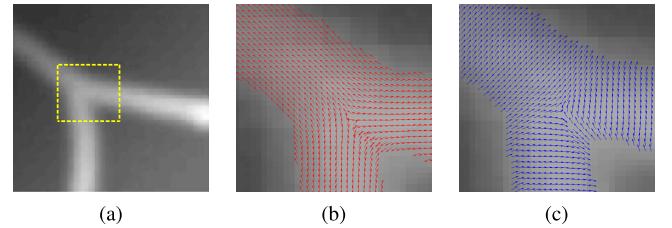


Fig. 3. Illustrative example of the weighted TuFF. A simple tubular structure is shown in (a). (b) The weighted axial vector field and (c) The weighted orthogonal vector field for the sub image enclosed in the yellow rectangle. Weights of the vector fields are computed using (11, 12). Image courtesy of [43].

**1) Scale Selection:** Since neurites vary in thickness, a scale space analysis is required to capture the variability in their width. If  $S = \{\sigma_{min}, \dots, \sigma_{max}\}$  denotes the scale space, for  $\sigma \in S$ , the tubularity measure or *vesselness* score [16] for a 3D image at  $\mathbf{x} \in \Omega$  can be written as

$$N_\sigma(\mathbf{x}) = \begin{cases} \frac{|\lambda_1(\mathbf{x}) - \lambda_2(\mathbf{x})|^2}{|\lambda_1(\mathbf{x})||\lambda_2(\mathbf{x}) - \lambda_3(\mathbf{x})|} & \text{if } \lambda_2(\mathbf{x}), \lambda_3(\mathbf{x}) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The optimal scale  $\sigma^*$  at  $\mathbf{x} \in \Omega$  and its corresponding vesselness score  $N(\mathbf{x})$  is computed as follows:

$$\sigma^*(\mathbf{x}) = \operatorname{argmax}_{\sigma \in S} N_\sigma(\mathbf{x}) \quad (4)$$

$$N(\mathbf{x}) = \max_{\sigma \in S} N_\sigma(\mathbf{x}) \quad (5)$$

The scale space vesselness response  $N(\mathbf{x})$  assumes higher value at locations of local tubularity over non-filamentous positions. It should be noted that (5) yields evidence of the presence of a neurite by suppressing the non-filamentous structures, thus introducing a mechanism for dealing with the structure noise.

Given  $H_{\sigma^*}(\mathbf{x})$ , the hessian matrix of the image  $f(\mathbf{x})$  at the optimal scale  $\sigma^*(\mathbf{x})$ , we can compute the TuFF. For a 3D image, the TuFF consists of a vector field  $\mathbf{v}_1(\mathbf{x})$  along the vessel axial direction and two vector fields  $\mathbf{v}_2(\mathbf{x})$  and  $\mathbf{v}_3(\mathbf{x})$  whose non-zero components are orthonormal to the axial field  $\mathbf{v}_1(\mathbf{x})$  (Fig. 3). Formally, this can be computed as

$$\mathbf{v}_k(\mathbf{x}) = \begin{cases} \mathbf{e}_k^*(\mathbf{x}) & \text{if } \lambda_1^*(\mathbf{x}) \approx 0 \text{ and } \lambda_2^*(\mathbf{x}), \lambda_3^*(\mathbf{x}) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$\mathbf{e}_k^*(\mathbf{x})$  denotes the normalized eigenvector corresponding to the eigenvalue  $\lambda_k^*(\mathbf{x})$  of the hessian matrix  $H_{\sigma^*}(\mathbf{x})$  such that  $|\lambda_1^*(\mathbf{x})| \leq |\lambda_2^*(\mathbf{x})| \leq |\lambda_3^*(\mathbf{x})|$  ( $\forall \mathbf{x} \in \Omega, k = 1, 2, 3$ ). In the following subsections, we show how TuFF can be incorporated in a level set framework to perform neuron segmentation.

### B. Neuron Segmentation Using TuFF

Our method performs segmentation via minimization of the energy functional  $\mathcal{E}(\phi)$ . This energy functional can be

mathematically written as:

$$\mathcal{E}(\phi) = \mathcal{E}_{reg}(\phi) + \mathcal{E}_{evolve}(\phi) + \mathcal{E}_{attr}(\phi) \quad (7)$$

$$\mathcal{E}_{reg}(\phi) = v_1 \int_{\Omega} |\nabla \phi(\mathbf{x})| \delta(\phi) d\mathbf{x} \quad (8)$$

$$\mathcal{E}_{evolve}(\phi) = - \int_{\Omega} \sum_{i=1}^d \alpha_i(\mathbf{x}) \langle \mathbf{v}_i(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle^2 H(\phi) d\mathbf{x} \quad (9)$$

Here  $\mathcal{E}_{reg}$  and  $\mathcal{E}_{evolve}$  are the energy functionals corresponding to the smoothness of the curve and the curve evolution respectively. The functional  $\mathcal{E}_{attr}$  contributes towards creating a local attraction energy. This attraction energy is to be designed in a manner such that minimizing it would result in a force field to join the local, disjoint neuron fragments. For our application, we do not define the attraction energy explicitly; instead, we compute the attraction force resultant from the energy (see Section II-E).

The vector  $\mathbf{n}(\mathbf{x}) = \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|}$  denotes the inward normal unit vector to the level sets of  $\phi$ .  $\langle \cdot, \cdot \rangle$  is the Euclidean inner product operator. The positive scalar  $v_1$  in (8) contributes to the smoothness of the zero level curve. The weighing parameter  $\alpha_i$  determines the contribution of the orthogonal and axial components of the TuFF in curve evolution. Choice of  $\alpha_i$  is an important aspect which would be discussed shortly.

In practice, the ideal Dirac delta function  $\delta(\phi)$  and the Heaviside function  $H(\phi)$  are replaced by their regularized counterparts  $\delta_\epsilon(\phi)$  and  $H_\epsilon(\phi)$  respectively as defined in [36]. Regularization of the functions is controlled by the positive parameter  $\epsilon$ . The regularizing energy term  $\mathcal{E}_{reg}$  in (8) constrains the length of the zero level curve of  $\phi$ . The amount of smoothing is controlled by the parameter  $v_1 \geq 0$ . Using a small value of  $v_1$  has the effect of encouraging presence of smaller, disjoint objects in the final solution. We report the actual values of  $v_1$  while discussing the implementation details.

### C. Discussion of Curve Evolution via TuFF

The essence of our technique lies in the design of curve evolution energy  $\mathcal{E}_{evolve}$  in (9). In absence of the attraction force energy, the level curve evolution (which results from minimizing the energy term (9)) depends on the contribution of the axial and orthogonal components of the tubularity flow field. The design of the functional (9) is such that the axial vector field component  $\mathbf{v}_1$  is responsible for propagating the curve to fill out the vessel thickness. Or in other words, the axial field promotes curve evolution in a direction perpendicular to itself. Identically, the orthogonal components  $\mathbf{v}_2, \mathbf{v}_3$  encourage curve propagation in a direction perpendicular to themselves, i.e. along the axis of the neuron filaments. Let us illustrate this phenomenon by using a 2D synthetic image containing a single tubular structure (Fig. 5).

*1) Effect of the Axial Component of TuFF:* Maximizing the total squared inner product  $\int_{\Omega} \alpha_1(\mathbf{x}) \langle \mathbf{v}_1(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle^2 H_\epsilon(\phi)$  (or minimizing its negative) with respect to the embedding function  $\phi$  results in maximally aligning the inward normal vectors  $\mathbf{n}(\mathbf{x})$  of the zero level sets of  $\phi$  and its inner isocontours with the axial flow field  $\mathbf{v}_1(\mathbf{x})$ . As shown in the first row

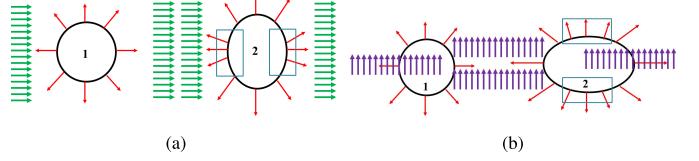


Fig. 4. Illustration of curve evolution due to (a) axial component  $\mathbf{v}_1$  and (b)  $\mathbf{v}_2$ . Note how the contour should change to align the surface normals (shown in red arrow) with the vector fields (shown in green and purple arrows respectively) in order to minimize the evolution energy. The initial curve is marked as 1. The evolution forces create the new curves 2. Note how the curves assume elliptical shape to align the level set normals with the vector fields. The normal vectors are maximally aligned in the region enclosed by the rectangles.

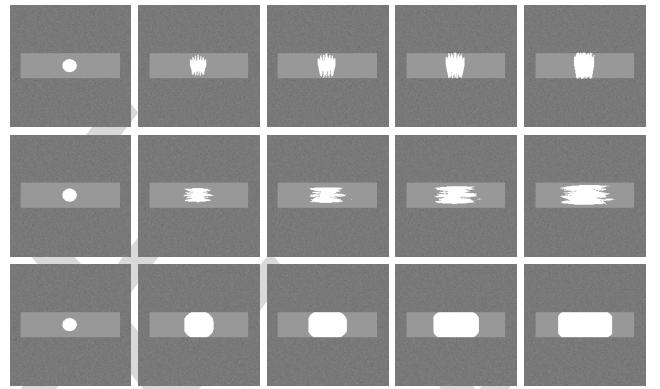


Fig. 5. Segmentation steps for the synthetic rectangular are shown. The first row shows curve evolution due to axial component of TuFF, i.e.  $\alpha_2 = 0$ . The second row shows surface evolution for  $\alpha_1 = 0$ .  $v$  is set to 0 for both the cases. The bottom row displays smooth curve propagation with  $\alpha_0 = 1$ ,  $\alpha_1 = 5$ ,  $\sigma = 3$  and  $v = 0.02$ .

of Fig. 5, this requires the level sets of  $\phi$  to be re-aligned such that the normal vectors  $\mathbf{n}(\mathbf{x})$  aligns itself with the axial field  $\mathbf{v}_1(\mathbf{x})$ . This results in curve evolution in a direction orthogonal to the vessel axis, causing elongation of the level curves along the vessel width.

*2) Effect of the Orthogonal Component of TuFF:* Using a similar argument, maximizing the second term corresponding to the orthogonal component in (9) performs alignment of the inward normal vectors with the vector field  $\mathbf{v}_2(\mathbf{x})$ , creating an elongation force which allows the level curves to propagate along the vessel axis. This is shown in the second row of Fig. 5. For intuitive understanding of the above mentioned phenomenon, Fig. 4(a) and (b) is provided to graphically demonstrate how the curve evolution is affected by the axial and the normal components of TuFF.

*3) Effect of the Vector Field Weights:* Ideally, the parameters  $\alpha_i(\mathbf{x})$ ,  $i = 1, \dots, d$ , should be chosen such that curve propagation is discouraged outside the tubular neurite segments, so as to avoid leakage into the background. i.e. for a voxel  $\mathbf{y}$  with low vesselness score, we require  $\alpha_i(\mathbf{y}) \approx 0$ , for  $i = 1, \dots, d$ . Moreover, since the neurites are elongated structures, it is desired that the contour evolution be more pronounced near the filament centerline than at the edges. This can be stated as

$$\frac{\alpha_j(\mathbf{x})}{\alpha_1(\mathbf{x})} \geq 1 \quad (j = 2, \dots, d) \quad \text{and} \quad \alpha_1(\mathbf{x}), \dots, \alpha_d(\mathbf{x}) > 0 \quad (10)$$

$\forall \mathbf{x} \in \Omega$ . Respecting the above constraints, we propose the following functions for choosing the parameters.

$$\alpha_1(\mathbf{x}) = N(\mathbf{x}) \quad (11)$$

$$\alpha_j(\mathbf{x}) = N(\mathbf{x}) \left( a_0 + \exp \left( -\frac{|\nabla_\sigma f(\mathbf{x})|}{a_1} \right)^2 \right) \quad (12)$$

$\forall \mathbf{x} \in \Omega$  and  $j = 2, \dots, d$ .  $N(\mathbf{x})$  is the vesselness score which is obtained from (5).

Let us discuss the isotropic case, when  $a_0 = 1$  and  $a_1 \rightarrow \infty$ . Since the unit normal vector  $\mathbf{n}(\mathbf{x})$  lies in the vector space spanned by  $\{\mathbf{v}_i(\mathbf{x})\}$ , it can be written as  $\mathbf{n}(\mathbf{x}) = \sum_{i=1}^d m_i \mathbf{v}_i(\mathbf{x})$ . This reduces (9) to

$$\mathcal{E}_{evolve}(\phi) = - \int_{\Omega} N(\mathbf{x}) \sum_i \langle \mathbf{v}_i(\mathbf{x}), \sum_j m_j \mathbf{v}_j(\mathbf{x}) \rangle^2 H(\phi) d\mathbf{x}$$

Since the eigenvectors are orthonormal,  $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 1$  for  $i = j$  and 0 otherwise. Also, since  $|\mathbf{n}(\mathbf{x})| = 1$ , we have  $\sum_i m_i^2 = 1$ . Using this relation, we obtain  $\sum_i \langle \mathbf{v}_i(\mathbf{x}), \sum_j m_j \mathbf{v}_j(\mathbf{x}) \rangle^2 = 1$ . This reduces the evolution equation to

$$\mathcal{E}_{evolve}(\phi) = - \int_{\Omega} N(\mathbf{x}) H(\phi) d\mathbf{x} \quad (13)$$

The energy functional in (13) when minimized performs segmentation via vesselness weighted isotropic region growing along the neuron segments. Leakage of the contour outside vessel boundaries is prohibited by the vessel indicator function  $N(\mathbf{x})$  which provides evidence of tubularity by assuming higher value for the tubular objects than non tubular background.

With the discussion of the isotropic case, it is now easy to visualize the effect of the weights on curve evolution. From our previous discussion, we recall that  $\alpha_1$  and  $\{\alpha_j, j \neq 1\}$  influence curve propagation along the vessel width and axial direction respectively.  $|\nabla_\sigma f(\mathbf{x})|$  denotes the gradient magnitude of the image  $f(\mathbf{x})$ , which is filtered by a Gaussian kernel with variance  $\sigma^2$ . Since this term is high at the vessel boundaries and end points, the negative exponential term in (12) ensures higher response at regions near the vessel centerline. The tuning parameters  $a_0 \geq 1$  and  $a_1$  determine the relative influence of the axial curve motion to the motion along the vessel width. In other words, in an anisotropic setting, (12) suggests that the level curves evolve with higher curvature near the vessel medial axis than at the edges, which percolates to the isotropic case when  $a_0 = 1$  and  $a_1 \rightarrow \infty$ .

Since the neurite filaments are predominantly thin, elongated structures, we observe that the isotropic case yields sufficiently appropriate segmentation results when the initialized zero level set encompasses the filament width. Nevertheless, the proposed framework in (7) is general as demonstrated in Fig. 5, and is applicable to segmentation problems where vessel thickness is significant and the initialized zero level contour does not fill out the vessel width completely. This is in contrast to the approach in [30], where segmentation of thicker vessels needs separate treatment. The bottom row of Fig. 5 shows the successive motion of the level sets for the above mentioned choice of alpha. It is observed that the evolving level set encompasses both the width and length of the vascular structure.

#### D. Level Set Evolution Using Gradient Descent

The energy functional in (7) can be minimized using variational calculus techniques [44]. Taking the Gâteaux variation of  $\mathcal{E}(\phi)$  with respect to  $\phi$ , we obtain from (7)

$$\frac{\delta \mathcal{E}}{\delta \phi} = \frac{\delta \mathcal{E}_{reg}}{\delta \phi} + \frac{\delta \mathcal{E}_{evolve}}{\delta \phi} + \frac{\delta \mathcal{E}_{attr}}{\delta \phi} \quad (14)$$

$\phi$  can be iteratively updated using gradient descent technique, i.e. setting  $\frac{\delta \mathcal{E}}{\delta \phi} = -\frac{\partial \phi}{\partial t}$  with  $t$  denoting the pseudo time parameter for the iterative scheme:

$$\frac{\partial \phi}{\partial t} = \mathcal{F}_{reg}(\mathbf{x}) + \mathcal{F}_{evolve}(\mathbf{x}) + \mathcal{F}_{attr}(\mathbf{x}) \quad (15)$$

$\mathcal{F}_{reg}$  and  $\mathcal{F}_{evolve}$  are the forces due to the regularizing energy and the evolution energy functional respectively. These forces are derived by solving the Euler-Lagrange equation for level set evolution as:

$$\mathcal{F}_{reg}(\mathbf{x}) = v_1 \operatorname{div} [\mathbf{n}(\mathbf{x})] \delta_\epsilon(\phi) \quad (16)$$

$$\begin{aligned} \mathcal{F}_{evolve}(\mathbf{x}) = & \delta_\epsilon(\phi) \sum_{j=1}^d \{\alpha_j(\mathbf{x}) \beta_j^2(\mathbf{x})\} \\ & - 2 \operatorname{div} \left[ \sum_{j=1}^d \eta_j(\mathbf{x}) (\mathbf{v}_j(\mathbf{x}) - \beta_j(\mathbf{x}) \mathbf{n}(\mathbf{x})) \right] \end{aligned} \quad (17)$$

The coefficients  $\beta_j$  and  $\eta_j$  are defined as follows:

$$\beta_j(\mathbf{x}) = \langle \mathbf{v}_j(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle \quad (18)$$

$$\eta_j(\mathbf{x}) = \frac{\alpha_j(\mathbf{x}) \beta_j(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} H_\epsilon(\phi) \quad (19)$$

The derivation details are shown in the Appendix.

#### E. Local Attraction Force Design

The attraction force  $\mathcal{F}_{attr}$  in (15) is introduced to accommodate the signal intensity variation (and signal loss) across the neurite branches, as shown in Fig. 2. Such signal attenuation introduces unwarranted discontinuities in the filamentous objects, resulting in disjoint fragments. Also, discontinuities may be present at the neurite junctions and noisy regions due to the nonlinear response of the vesselness function in (5). In such a scenario, the TuFF based evolution energy term in (9) is not adequate to perform segmentation alone. This insufficiency motivates the inclusion of an attraction force component. Designing this attraction force requires analysis of the connected components at each time epoch of level set propagation. At a time  $t$  for evolution of the level set function  $\phi(\mathbf{x}, t)$ , the set of connected components  $\mathcal{C}(t)$  can be obtained as

$$\mathcal{C}(t) = H(\phi(\mathbf{x}, t))$$

$$\text{where } H(y) = \begin{cases} 1 & \text{for } y \geq 0 \\ 0 & y < 0 \end{cases} \quad (20)$$

The set of connected components  $\mathcal{C}(t) = \{c_1, \dots, c_p\}$  represents the binary segmentation at time  $t$ , which consists of  $p \geq 1$  disjoint connected components. Note that this binarization does not require a sophisticated segmentation, since the binary components are obtained by extracting the

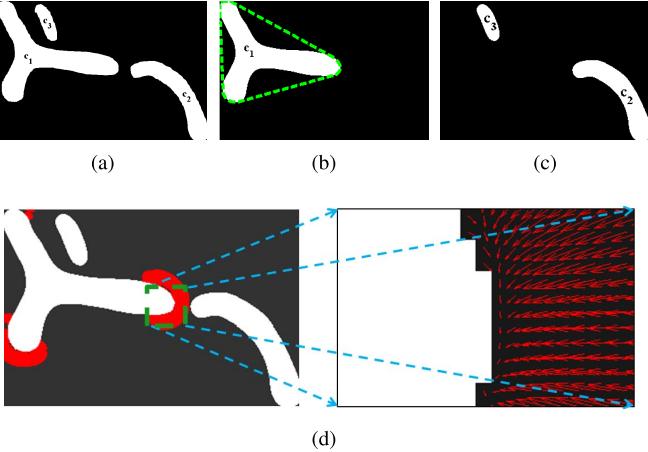


Fig. 6. (a) Set of disjoint connected components  $\{c_1, c_2, c_3\}$  at a particular step of iteration. (b) shows a parent component, the green dotted line marking its convex hull. The remaining children are shown in (c). (d) shows the attraction force obtained via (23) in red arrows, magnified for visual clarity.

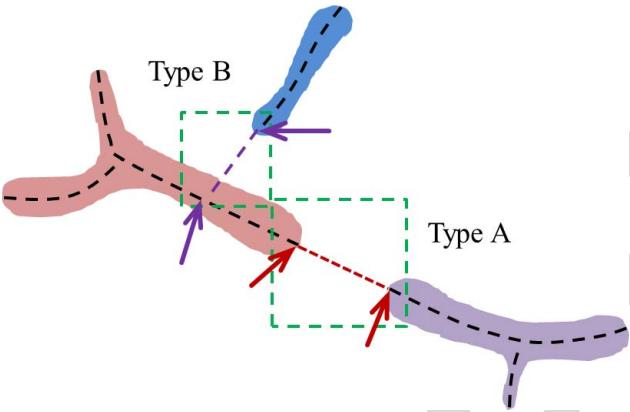


Fig. 7. Two types of discontinuities between the disjoint components. The *Type A* discontinuity can be resolved by joining the end points of the center lines of the respective branches. *Type B* is more difficult, where discontinuity occurs between a branch end point and an intermediate point on the centerline of the other branch.

interior of the zero level sets of the embedding function. Each disjoint component  $c_j$  is a potential candidate or a *parent* which has the capability of attracting the remaining *children*  $c_k$ ,  $k \neq j$ ,  $(j, k = 1, \dots, p)$ . This is illustrated in Fig. 6(a)–(c), where the component  $c_1$  acts as a parent component and  $c_2$  and  $c_3$  are the children.

*1) Candidate Points for Attraction Force Field:* The primary responsibility of the attraction force is to enable the propagating contour surface to attach itself to local disjoint fragments. However, not all points on the connected components are candidates for creating the attraction force. This is because in a majority of the prevalent discontinuities, at least one of the two disconnected portions are likely to be joined via boundary points which represent region of high curvature (see Fig. 7). If we denote the boundary of a component  $c_j$  by  $\delta c_j$ , to enable a parent to attract a child, we need to design an attraction field which is generated by a set of candidate points lying on the parent boundary. Therefore, for a parent component  $c_j$ , a point  $\mathbf{y} \in \delta c_j$  belongs to the candidate set if  $\mathbf{y}$  is a point the convex

hull [45]  $\mathcal{H}_j$  of  $c_j$  (Fig. 6(b)). Formally, the candidate point set  $\mathcal{M}_j$  for the connected component  $c_j$  is defined as

$$\mathcal{M}_j = \{\mathbf{y} \in \delta c_j : \exists \mathbf{x}_j \in \mathcal{H}_j \text{ s.t. } \|\mathbf{y} - \mathbf{x}_j\|_2 \leq \Delta\} \quad (21)$$

$\Delta$  is a positive parameter that includes local boundary coordinates of the neighboring points on the convex hull.

*2) Attraction Force Field Design:* The candidate set of points for a parent component is responsible for generating a force field capable of attracting the candidate children towards itself for potential merging. This needs to be designed such that the attraction field vectors point toward the region of interest, which is the parent candidate point set for this purpose. We show that an efficient solution may be obtained by using vector field convolution (VFC) to create the attraction force field.

VFC [32] is a technique primarily designed to create smooth external force field for parametric active contours. The specially designed vector field kernel (22) generates the desired external force when convolved with the object edge map, with the capability of attracting a contour to the region of interest.

$$\begin{aligned} \mathbf{K}(\mathbf{p}) &= -m(\mathbf{p}) \frac{\mathbf{p}}{\|\mathbf{p}\|} \\ m(\mathbf{p}) &= \exp(-||\mathbf{p}||^2/\gamma^2) \end{aligned} \quad (22)$$

$\mathbf{p} = \mathbf{0}$  denotes the kernel center. The capture range of the vector field is controlled by the parameter  $\gamma$ .

The set of candidate points  $\mathcal{M}_j$  for a parent  $c_j$  serves as the region of interest to which other components are likely to be attracted. Performing convolution of the candidate set with the kernel in (22) results in a vector field where the vectors are directed toward the parent, their magnitude attenuating gradually with distance from the candidate set. If  $E_j(\mathbf{x})$  is a binary edge-map which assumes a value of 1 only at points in  $\mathcal{M}_j$ , we can obtain the attraction force field  $\Gamma_j$  due to the parent  $m_j$  as

$$\Gamma_j(\mathbf{x}) = E_j(\mathbf{x}) * \mathbf{K}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \quad (23)$$

The nature of the attraction force field can be intuitively understood from Fig. 6. Fig. 6(a) shows three connected components and the representative parent  $c_1$  enclosed by its convex hull (shown in (b)). Fig. 6(c) illustrates the attraction force field due to the parent as the red arrows which are oriented in the direction of the parent component. The capture range, which is specified by  $\gamma$ , is shown by the red region.

Adopting this policy for designing the attraction field enjoys a few benefits. First, with a specified capture range, we can impose a locality in the approach, by discouraging distant segments to be connected to the parent. As  $\gamma$  increases, effect of the attraction force field gradually diminishes as one moves further from the parent. Moreover, the candidate set is chosen such that only the convex portions of the parent boundary are capable of generating the force field. This ensures not all local structures are potential candidates for linking. For example, in Fig. 6 the component  $c_3$  is not in the capture range of the force field of  $c_1$ , although it resides in the parent's local neighborhood. To summarize, the attraction force field is designed such that it may attract local connected components

which are present in near vicinity of the parent's boundary convexity.

3) *Attraction Force*: For a parent-child pair  $c_i$  and  $c_j$ , the parent attracts the child with a force  $\mathcal{F}_{attr}^{(i,j)}$  given by

$$\mathcal{F}_{attr}^{(i,j)}(\mathbf{y}) = \kappa_i \langle \Gamma_i(\mathbf{y}), -\mathbf{n}(\mathbf{y}) \rangle \theta_j(\mathbf{y}) \quad (24)$$

The indicator function  $\theta_j(\mathbf{y}) = 1$  if  $\mathbf{y} \in \delta c_j$  and 0 otherwise.  $\kappa_i$  is the normalized mass of the component  $c_i$  which is computed as the ratio of the number of pixels/voxels in  $c_i$  to the total pixels/voxels in  $\{c_1, \dots, c_p\}$ . The inner product term in (24) suggests that higher force of attraction is experienced by a point on a child's boundary if the outward normal at that point is oriented along the attraction field.

By introducing the factor  $\kappa_i$ , we equip heavier connected components with more attractive power. Assuming that the neurites occupy larger volume than the noisy background voxels, we clean the solution of the level set function by performing an area opening operation which eliminates small components with area less than a pre defined threshold [46]. This filtering operation avoids undesired objects to participate in the attraction force field computation. Now, for each parent-child pair in the filtered component space, we can compute the total attraction force  $\mathcal{F}_{attr}$  in (15) as

$$\mathcal{F}_{attr}(\mathbf{y}) = v_2 \sum_{i=1}^p \sum_{j \neq i}^p \mathcal{F}_{attr}^{(i,j)}(\mathbf{y}), \quad \forall \mathbf{y} \in \Omega. \quad (25)$$

The positive scalar  $v_2$  determines the effect of the attraction force on curve evolution. A finite difference scheme is used to solve the PDE in (15) with initial value obtained using Otsu's global segmentation [41] and Neumann boundary condition.

#### F. Handling of Discontinuities

Typically, one may encounter two major sources of structure discontinuity arising from initial segmentation. Fig. 7 shows three synthetic, disjoint components at an arbitrary stage of level set evolution. The type A discontinuity occurs when connectivity is absent between the end points or leaves of the centerline of the respective objects. Type A discontinuities dominate our application, and connectivity analysis of type A may be performed via Tree2Tree [16], by investigating the geometric orientation and Euclidean distance between the end points. However, end-point analysis algorithms like Tree2Tree are unable to process the type B discontinuities, where the link needs to be established between the terminal node of one component with a non-terminal point on the other object. This is where the proposed level set framework wins over conventional component linking algorithms since level sets are proficient in handling topological changes of the evolving segmentation.

1) *Type A Discontinuities*: Type A discontinuities are relatively simpler to analyze. If the neuron filament signal intensity is uniform, then the evolution force component of (15) sufficiently propagates the level sets until they are finally merged. However, when the signal drop is substantial, the attraction force term in (15) assists the parent and the

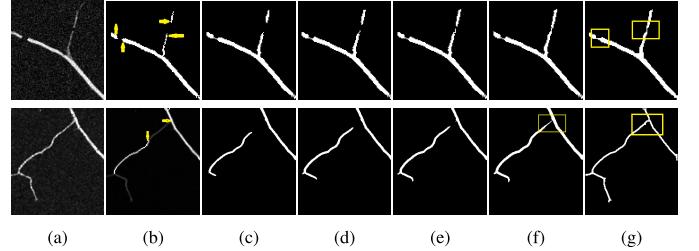


Fig. 8. (a) and (b) show the original image and the initial global segmentation respectively for two cases demonstrating handling of Type A (top row) and Type B (bottom row) discontinuities. (c)-(f) shows segmentation at subsequent time intervals. (g) shows the final segmentation, where the structure gaps have been closed (the merged portions are enclosed in rectangles).

child component to exert attractive forces on one another, thus propagating the curves till they merge. A demonstration is shown in the first row of Fig. 8. The initial segmentation using Otsu's method creates type A gaps, which are ultimately merged. We have intentionally eliminated a portion of the neuron's branch to demonstrate that our methodology works even in complete absence of signal.

2) *Type B Discontinuities*: Type B discontinuity involves two segments, for which connectivity needs to be established between one component's end point (or tip) with the other component's body. In presence of adequate signal intensity, TuFF drives the geometric contours toward the participating structure as per the filament orientation. However, when signal intensity drops, the attraction force takes over. An example is shown in the second row of Fig. 8(b), where the initial segmentation creates a type B gap. The situation is different from that of type A, where both the components may attract each other. In case of type B, only one component can assume a parent's role. Note that this is the extreme scenario, where the underlying signal strength is so feeble that it renders the evolution force term useless. However, assuming that the parent's mass is not negligible, this attraction force is strong enough to pull the local child connected component for potential merging. It should be noted that only those regions on the child's boundary whose outward normals are maximally aligned with the exerted force field are attracted toward the parent.

#### G. Neuron Tracing via Centerline Extraction

Numerical implementation of (15) allows iterative computation of the level set function, which can be expressed as

$$\phi^{(k+1)} = \phi^{(k)} + \Delta t \mathcal{L}^{(k)} \quad (26)$$

The learning rate  $\Delta t$  is fixed to a small value ( $\approx 0.1$ ) to allow stable computation.  $\mathcal{L}^{(k)}$  denotes the discretized version of the right hand side of (15).  $\phi^{(k)}$  is the level set function at iteration  $k$ . To initialize the active contour, we require the initialized curve to be inside the neurite structure. The initial level set function may be easily obtained via few mouse clicks to select a region inside the neuron structure. However, to avoid this human involvement, we perform a global thresholding of the scale space vesselness image (5)

using Otsu's technique [41], followed by noisy binary segment removal using the area open filter [46]. The iterative procedure is halted when no significant change in the length of the zero level curve of  $\phi$  is observed. At convergence, the neuron structure is extracted by selecting the largest binary component in the solution. A cubic spline is then fitted to each branch of the obtained centerline to obtain smooth tracing of neuron centerline.

### H. Summary of TuFF

Before proceeding to experimental results, we provide a summary of the TuFF algorithm and highlight its salient features. First, we avoid human intervention in terms of seed point selection. Automated initialization of the level set is performed by Otsu's global thresholding [41] followed by noise removal using morphological area open operators [46]. The level set function is computed from this initialized segments using binary distance transform.

Second, TuFF presents a natural framework to process both type A and type B discontinuities (Fig. 8). This is a major improvement over the tracer Tree2Tree [16], where the inability to handle type B discontinuity introduces several false connections in the solution.

Finally, TuFF is capable of joining broken neurite fragments even in complete absence of signal. The proposed attraction force field is independent of the local signal intensity and depends only on the morphology and relative positioning of the connected components. This feature improves on the widely used local intensity seeking neuron tracers [13], which are susceptible to illumination variation in the images of neural structure. The TuFF guided evolution energy is combined with the attraction force component in a mathematically elegant, integrated fashion as opposed to a multistage sequential processing pipeline.

## III. EXPERIMENTAL RESULTS

In this section we demonstrate the efficacy of our method by experimental analysis of both 2D and 3D confocal images. We further compare our segmentation accuracy to that of three widely used neuron tracers.

### A. Dataset for Segmentation

We test the performance of TuFF segmentation algorithm on sets of 2D and 3D confocal microscopy images. The 2D images are primarily used to demonstrate the efficacy of TuFF over component analysis algorithms like Tree2Tree [16]. The 3D image data set consists of 24 confocal microscopy images of the Drosophila larva, which are labeled by green fluorescence protein (GFP). Out of these 24 images, 16 images are captured in the Condron Lab of the University of Virginia. The images are captured using a laser scanning confocal microscope and has a horizontal pixel width of  $0.14\mu m$  and vertical pixel width of  $0.18\mu m$ . These images are characterized by intense background clutter from non neuronal objects (such as the food particles, mildly fluorescing tissues etc.) and considerable contrast and intensity variation.

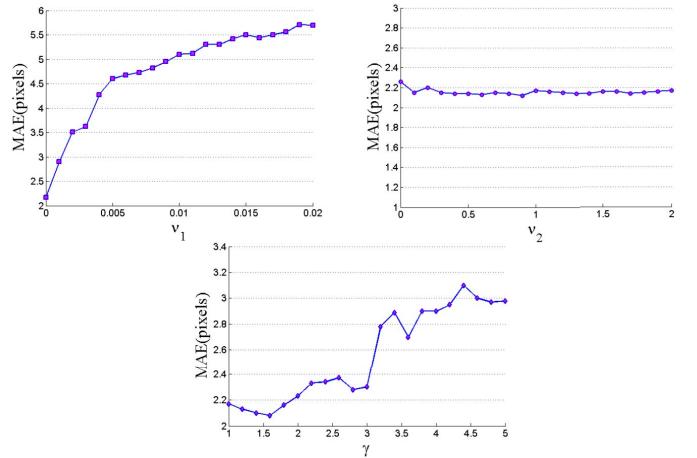


Fig. 9. Sensitivity analysis of the parameters. The mean absolute error of the traced centerline are plotted in the vertical axis for different values of the tuning parameters.

The second data set for 3D analysis consists of 8 olfactory (axonal) projection (OP) image stacks of Drosophila larva. These images were used in the Diadem challenge [47] and like the previous dataset, these neurons are also imaged by a confocal microscope. These OP-data set images are less noisy and the contrast is better than the images in Condron data set. However, the neurons in this data set exhibit acutely complicated structural appearance in addition to occasional intensity heterogeneity along the neurite filaments.

### B. Parameter Selection

The level set evolution equation (15) depends on a few parameters. The evolution force  $\mathcal{F}_{\text{evolve}}$  requires specifying the positive scalars  $a_0$  and  $a_1$  in (12) which controls the anisotropy of curve evolution. As we have discussed before, since the neurite thickness in our case does not vary considerably, we have adopted the isotropic case, as it requires lesser computation. Therefore, we choose  $a_0 = 1$  and a very high value for  $a_1$ .

The smoothness of the evolved curve is controlled by the parameter  $v_1$  in (17). Effect of gradually increasing  $v_1$ , keeping other parameters fixed results in an increased mean absolute error in tracing, as shown in Fig. 9. For our experiments,  $v_1$  is fixed at a small value in the range  $0 - 0.02$ .

The attraction force defined in (25) depends on the weighing parameter  $v_2$  and the parameter  $\gamma$  controlling the local capture range. As we observe in Fig. 9 our algorithm is relatively robust to the choice of  $v_2$ . However, we notice that a very low value of  $v_2$  restricts the attraction force from closing small gaps. For all our experiments, we select  $v_2 = 1$ . The term  $\gamma$  induces locality in the capture range for the attraction force. While a small value of  $\gamma$  can be too restrictive, a relatively high value attracts distant structures to be merged to the attracting component (see Fig. 9). Note that we are interested in connecting the disjoint structures over a local neighborhood. Based on our collaborator's knowledge about the dataset, we observe that typically  $\gamma$  ranges between  $0.2 - 1.5\mu m$  ( $\approx 1-7$  pixels) for our data. Setting these biologically inspired bounds on the range of  $\gamma$ , we proceed to select the value

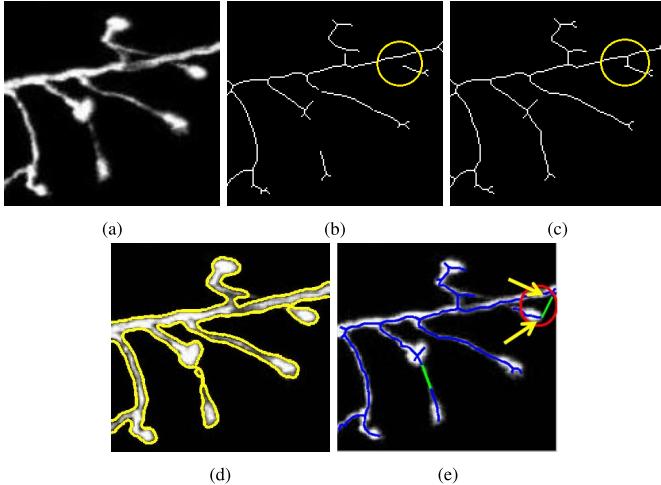


Fig. 10. (a) 2D neuron sub-image. (b) Centerline of the initial segmentation using [41]. The type B discontinuity is highlighted by the yellow circle. (c) Centerline obtained after segmentation using TuFF. (d) Final segmentation via TuFF. (e) Tracing using Tree2Tree. A typical error in connectivity is indicated by the arrows.

in the following manner. First, at any stage of segmentation, we compute the median distance  $\rho$  between all the segments, and update the value of  $\gamma$  as  $\gamma^* = \rho/3$ . If the updated value is beyond the pre-selected upper or lower bounds, we select the closest boundary value for  $\gamma^*$ . This is repeated at each iteration to compute the attraction force.

Experimentally we have observed that the parameters  $\Delta$  and  $\epsilon$  can be prefixed to a particular value without affecting performance. For all experiments we choose  $\Delta = 5$  pixels and  $\epsilon = 1$  as suggested by the authors in [36].

### C. Efficacious Handling of Branch Connectivity

Previously, we have demonstrated the ability of TuFF to handle type A and type B discontinuities. In this section, we demonstrate the advantage of using TuFF over Tree2Tree [16] for determining branch connectivity. For this purpose, we show segmentation results on a few 2D neuron images. The 2D images are obtained from a maximum intensity projection of the corresponding 3D stacks. We also perform experiments on a few synthetically grown neurons, where the 2D imaging is performed by measuring the fluorescence from the fluorophores used to stain these neurons.

To set up Tree2Tree for segmentation, we follow the author's methodology of performing an initial segmentation to obtain a set of binary components. The component analysis stage of Tree2Tree then decides on the connection between the segments by analyzing their relative orientation. To initialize the level set for TuFF, we have used Otsu's segmentation, same as Tree2Tree, and the level set propagates according to (15). Fig. 10 demonstrates an example where Tree2Tree creates improper connection, due to its inability to handle type B discontinuity. The level set based methodology in TuFF performs proper segmentation (shown in Fig. 10(c), (d)). It is evident that the type B gap is closed by TuFF, where Tree2Tree fails to do so (see Fig. 10(c) vs (e)).

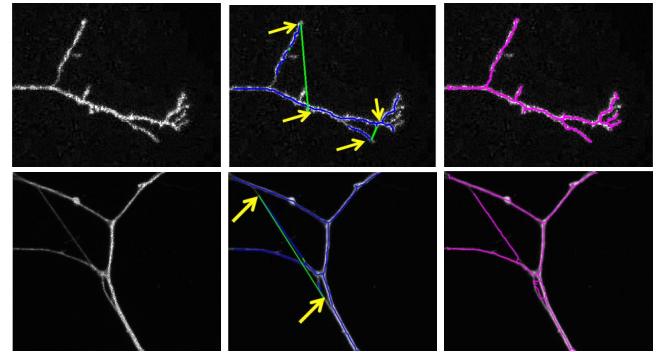


Fig. 11. The first column shows sample 2D neuron images. Tree2Tree [16] segmentation results are displayed in the second column. The edges linked by Tree2Tree are shown in green and the traced centerline is overlaid on the original image in blue. Excessive clutter restricts the efficiency of Tree2Tree, yielding improper connections, which are highlighted by the yellow arrows. The last column shows tracing output for TuFF algorithm, with the traced medial axis plotted in magenta.

Two more examples are shown in Fig. 11 where Tree2Tree's tracing (shown in blue) creates incorrect branch connection as compared to TuFF (shown in magenta). The connection errors are highlighted by the yellow arrows. Tree2Tree segmentation results suggest lack of robustness of the component linking scheme for complex structures embedded in a noisy environment. Furthermore the initial segmentation step in Tree2Tree often fails to detect low contrast objects, which cannot be recovered in future, since the multistage pipeline of Tree2Tree is unable to recover lost neurite portions.

The above examples suggest that TuFF handles bifurcations and component gaps successfully, since level sets are well equipped in handling topological changes. Also, the specially designed attraction force component of TuFF makes segmentation robust in cases where structure gaps result from very weak signal intensity (Fig. 10).

### D. Comparison of Segmentation Performance

In this section we present a comparative segmentation performance analysis of the proposed method TuFF versus three popularly used neuron tracers. The ground truth data for segmentation is obtained by manually selecting points on the neuron structure and joining them manually in a manner that the morphological structure is preserved. The Vaa3d software [48] is used for creating the ground truth. To evaluate the performance of TuFF, we compare its performance to the following algorithms.

1) *Graph Augmented Deformable (GD) Model [9]*: This semi automatic tool is extensively used for its relatively simple working methodology, which consists of a manual seed selection step followed by automated seed joining process by using graph theoretic techniques. Since the algorithm's efficacy is inversely proportional to the spatial distribution of selected seed points, we only select the neuron terminal points as the set of seeds. As the seed selection is performed manually, a practice which TuFF avoids, we believe that selecting the minimal set of seeds is essential to maintain fairness of comparison. Sample tracing results using this algorithm are shown in yellow.

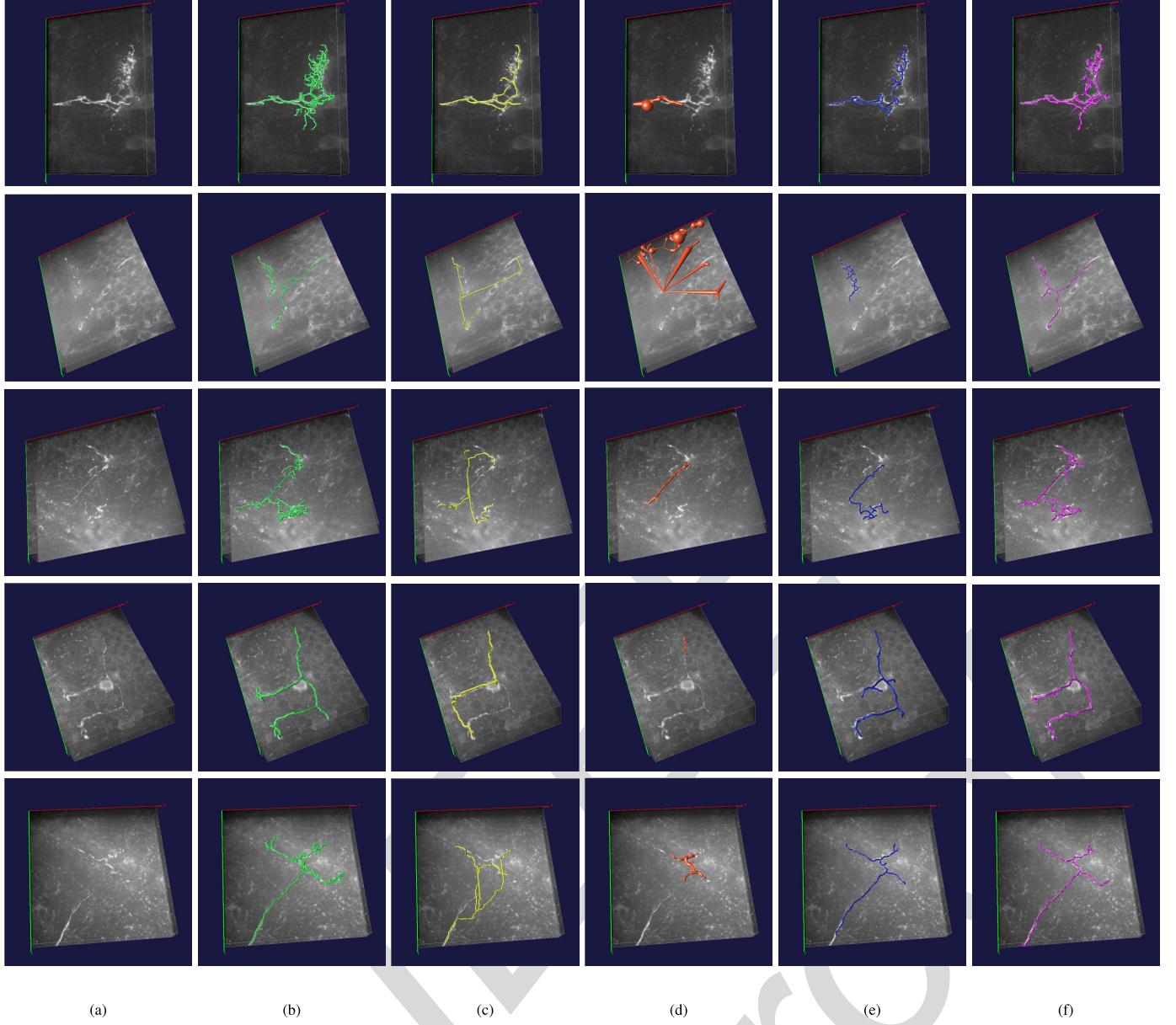


Fig. 12. Tracing results on 3D images of the UVA-Condron dataset. First column shows the original images, followed by the tracing outputs of the different algorithms. Tracing results of TuFF are shown in the last column in magenta. (a) 3D stack. (b) Ground truth. (c) GD model [9]. (d) NeuronStudio [13]. (e) Tree2Tree [16]. (f) TuFF.

2) *Neuronstudio* [13]: Neuronstudio is one of the state of the art publicly available automatic neuron segmentation software which is heavily used by biologists for tracing purpose. We have seen that segmentation accuracy of NeuronStudio is affected by the choice of the initial seed point. For each image in our dataset, we experiment with several initial seed locations and finally choose the one which yields the best visual segmentation result. Neuronstudio segmentation results are shown in orange color.

3) *Tree2Tree* [16]: As discussed earlier, Tree2Tree belongs to the category of seed independent neron segmentation methods. Setting up Tree2Tree requires an initial segmentation stage, followed by graph-theoretic component linking procedure. The segmentation results of Tree2Tree are shown in blue color.

For each of the above mentioned algorithms and TuFF, we first obtain the segmentation followed by neuron centerline detection. A cubic spline is fitted to each branch of the detected centerline. This spline fitted centerline of the neurons represent the tracing results.

#### E. Visual Assesment of Segmentation Results

1) *Results on Condron Data Set*: Fig. 12 shows the performance of the above mentioned neuron tracers on five representative neurons chosen from the Condron dataset. The 3D stacks are shown in the first column, followed by manual ground truth segmentation in the second column (shown in green). Tracing results using GD model [9] is plotted in yellow in the third column. The fourth and fifth columns

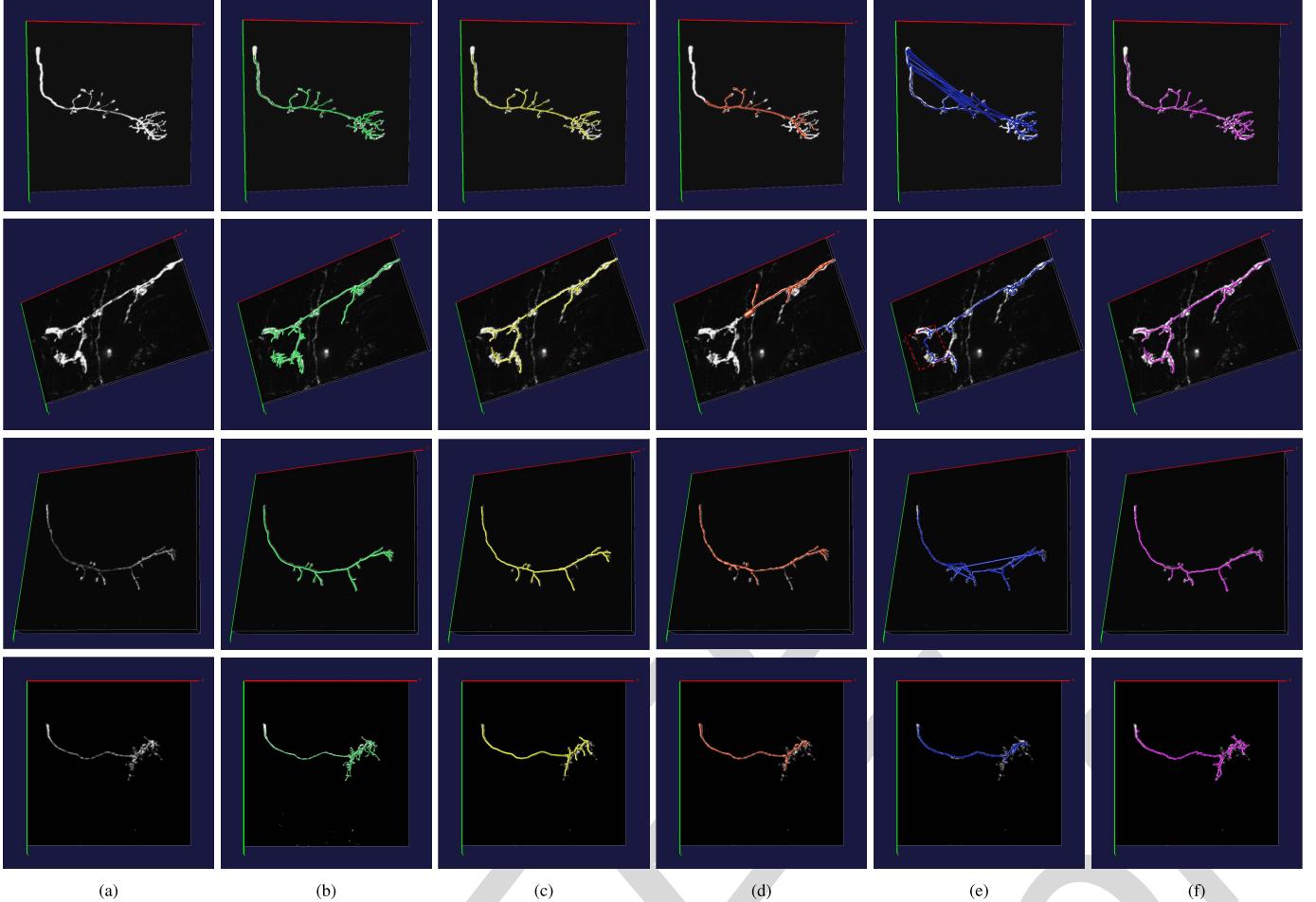


Fig. 13. Results on the images of the OP dataset. First column shows the original images, followed by the tracing outputs of the different algorithms. Tracing results of TuFF are shown in the last column in magenta. (a) 3D stack. (b) Ground truth. (c) GD model [9]. (d) NeuronStudio [13]. (e) Tree2Tree [16]. (f) TuFF.

show segmentation output using the automated techniques Neuronstudio and Tree2Tree (plotted in orange and blue color) respectively. Finally, the last column shows the neuron tracing due to TuFF (plotted in magenta).

It may be observed that these images are in general noisy, which makes the segmentation task difficult. Moreover, high structural complexity of the neurons require sophisticated mechanism to preserve the structural morphology. The severity of contrast variation and low SNR pose difficulty for the GD model. Even with manually selected terminal nodes, it is seen that the semi-manual tracer performs incorrect segmentation (Fig. 12, second column, rows 2–5). This is primarily due to the inability of the local search based technique fails to identify the actual filamentous path in presence of clutter. Furthermore, human assisted neurite termination detection proved to be a difficult and time consuming problem in these images owing to the high structural complexity.

Neuronstudio performs particularly poorly in these examples. The major reason can be attributed to the lack of continuity in the neurite structure and high signal variation, which forces the algorithm to converge prematurely. Also, the cluttered environment is detrimental to the performance of the local voxel scooping process of Neuronstudio. This results

in under segmentation and sometimes, incorrect segmentation due to leakage of the region growing technique.

Tree2Tree outperforms Neuronstudio, especially when the component linking algorithm is able to determine proper connectivity. We observe that Tree2Tree performs well if the initial segmentation step is reliable. However, under segmentation is an inherent problem in Tree2Tree due its inability to incorporate additional neuronal structures in its solution after initial thresholding.

On the other hand, TuFF performs segmentation efficiently, even in cluttered environment. A close inspection would reveal that important morphological entities like bifurcation points and branch locations are preserved (see Fig. 12 rows 2, 3 and 4), while the iterative directional region growing scheme prevents under segmentation of neurons.

*2) Segmentation Results on OP Dataset:* These image stacks exhibit relatively higher signal intensity than the Condon data set. However, neuron tracing is still a challenging task owing to their complicated structure and sudden intensity variations in the neurites, creating a fragmented, discontinuous appearance. This often results in type B discontinuity which demands sophisticated analysis. Fig. 13 compares the segmentation results for above mentioned algorithms.

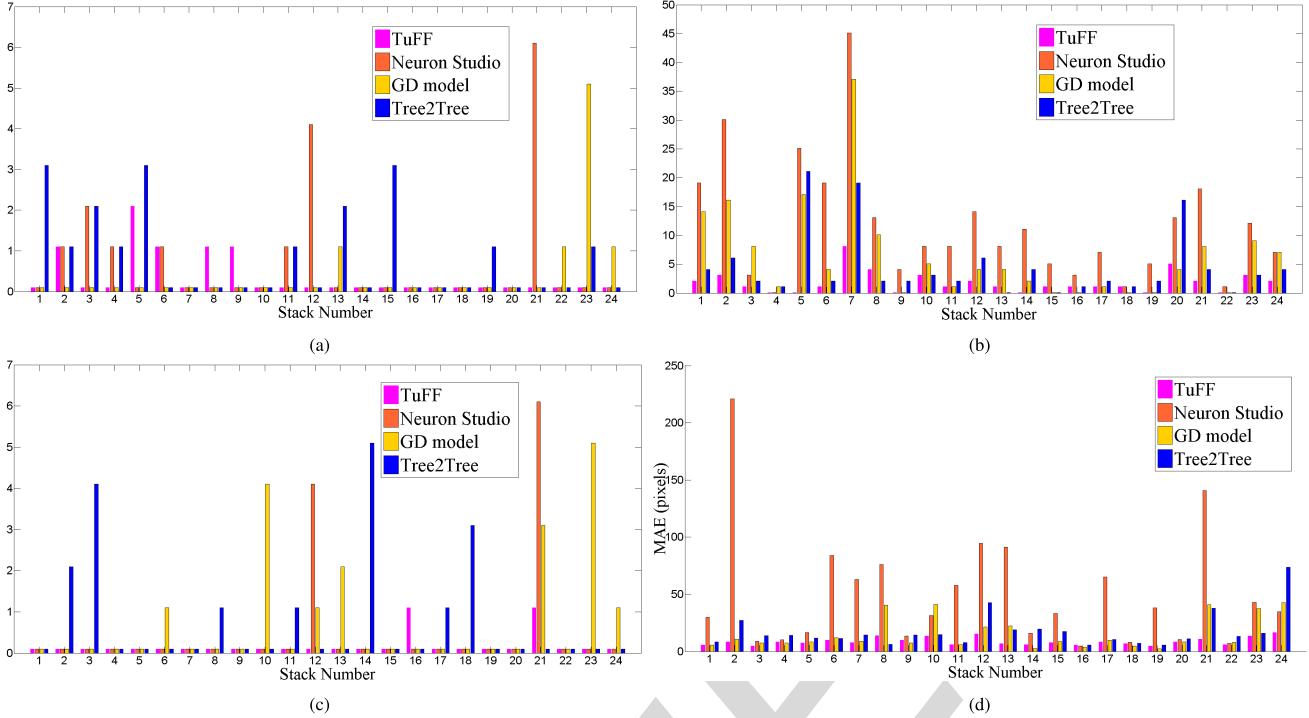


Fig. 14. (a)-(c): Quantitative performance of the four neuron tracers TuFF (pink), Neuron Studio (orange), GD model [9] (yellow) and Tree2Tree (blue) in terms of number of over-estimated branches, number of under-estimated branches and total number of wrong connections respectively. (d) quantifies the tracing accuracy in terms of mean absolute error defined in (27). (a) False positives. (b) False negatives. (c) Incorrect connection. (d) MAE.

Reduction in background clutter and increased signal intensity assists the semi automatic GD-model tracer. Since the images exhibit significant improvement in contrast, manual detection of seeds is less stressful. Still, the complicated structure of a few images (Fig. 13, row 1 for example) makes manual seed selection demanding. Performance of Neuronstudio also shows slight improvement in this dataset. However, despite brighter foreground and less noise, this local tracing scheme shows tendency to stop at intensity gaps, which needs to be modified manually at a later stage. On the other hand, it is observed that Tree2Tree's performance degrades significantly for this dataset. This is primarily due to a large number of improper branch connections. This connectivity error occurs mostly due to Tree2Tree's inability to handle type B discontinuities (Fig. 13, rows 1-3). In fact, even in relatively high SNR images Tree2Tree under performs significantly by extracting an improper structural morphology of the neurons. TuFF, however demonstrates good performance on these images by virtue of its ability to handle structure gaps automatically. The segmentation results are shown in the last column of Fig. 13. A qualitative assessment of the algorithm's performance is presented in the following sections.

#### F. Quantitative Performance Analysis

To quantify the segmentation performance, we identify four measures which reflects the efficiency of a particular neuron tracer. These are as follows: number of over-estimated branches (Fig. 14(a)), number of unidentified/missed branches (Fig. 14(b)), total number of incorrect branch connections (see Fig. 14(c)) and finally the mean absolute error in the

traced centerline with respect to the ground truth. The number of over determined/missed branches reflect the adequacy of an algorithm in respecting the morphology of the imaged neuronal structure. This quantification of the segmentation quality is performed by a human expert. However, since even the ground truth data is susceptible to subtle errors in computing the 3D skeleton, we have disregarded small branches (less than 5 units in length) from the analysis. The graphs in Fig. 14(a) and (b) suggests that over the whole data set, TuFF outperforms the competing algorithms in a majority of cases. It is observed in a few cases that Neuronstudio in particular misses a large number of branches, due to its inability to deal with fragmented structure.

The number of incorrect branch connections (Fig. 14(c)) indicate an algorithm's ability to tackle discontinuities. Indeed, improper connections often result when signal heterogeneity is significant. Apart from a few occasions, TuFF demonstrates its superiority in handling discontinuities better than other automated methods. To perform quantitative analysis of the traced neuron centerline, we compute the mean absolute error (MAE) of the obtained trace against the manually acquired ground truth. If  $\mathcal{P} = \{p_1, \dots, p_n\}$  and  $\mathcal{Q} = \{q_1, \dots, q_m\}$  denote the set of traced coordinates for a neuron, the mean absolute error (in pixels) between the traces is given by

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n \min_j |p_i - q_j| + \frac{1}{m} \sum_{i=1}^m \min_k |q_i - p_k| \quad (27)$$

$\forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, n\}$ . Mean absolute errors for the 24 3D images are plotted for each algorithm in Fig. 14(d). It is observed that TuFF outperforms the automated

TABLE I  
COMPARISON OF MAE

	TuFF	Neuron Studio	GD model	Tree2Tree
Avg. MAE	8.81	79.98	15.41	17.62
Median MAE	7.95	34.06	8.54	13.98
Std. Dev	3.4	50.6	14.03	15.08

tracers Tree2Tree and Neuronstudio in almost all of the 24 cases, except for the 8<sup>th</sup> and 16<sup>th</sup> stack, where Tree2Tree and Neuronstudio perform marginally better. Also, TuFF successfully competes with the semi-automatic GD-model, even outperforming it in some images in the Condron dataset.

The mean, median and standard deviation MAE of the four algorithms are reported in Table I. This suggests that on a whole TuFF outperforms its competitors with a mean and median MAE of 8.81 (pixels) and 7.95 (pixels) respectively. TuFF also exhibits 75% improvement of mean error over the second best performer, which is the semi-automatic tracer of Peng *et al.* If we compare its efficacy against the fully automated techniques, we obtain an improvement of over 98% over Tree2Tree, while Neuron Studio is outperformed with an improvement of greater than 400%. Also, the error standard deviation of TuFF is only 3.4 as compared to 50.6, 14.03 and 15.08 for Neuronstudio, GD-model and Tree2Tree. The visual segmentation results and the quantitative results presented here suggests the efficiency of TuFF in segmenting structurally complex neurons from cluttered confocal microscope images.

#### G. Note on Computational Efficiency

From a computational perspective, TuFF has the disadvantage that the segmentation is performed iteratively. Similar to all numerical PDE based methods, the speed of convergence can be controlled by setting a higher value for the learning rate, albeit at the cost of sacrificing accuracy. However, we should mention that in our implementation we have not concentrated on making the algorithm run faster. In fact, recent research suggest that significant decreases in computational cost can be achieved by using more intelligent numerical algorithms to solve the evolution equation. However, TuFF does hold an advantage over popular semi automatic tracers in the sense that no manual intervention is required. For example, to set up the GD model for segmentation, a human subject was assigned to visually determine around 20-30 end points to be selected for each 3D stack for seed initialization. With the current unoptimized implementation, TuFF takes approximately 300 seconds on average to segment a neuron from a 200 × 200 × 60 dimension 3D stack using Matlab for implementation on a 3.4 GHz Intel i7 processor with 8Gb RAM.

## IV. CONCLUSION

In this paper we have presented an automated neuron segmentation algorithm which can segment neurons from both 2D and 3D images. The proposed framework is suitable for tracing highly fragmented neurite images, and is capable of processing the structure discontinuities automatically,

while respecting the overall neuron morphology. Connectivity analysis is performed in a level set framework which presents a nice and simple alternative to graph based techniques which may introduce undesired branches in segmentation. The efficiency of TuFF is further demonstrated by its superior overall quantitative performance where it outperforms peer algorithms, including a semi manual tracer.

## APPENDIX

We provide the derivation of (17) for 2D, ie.  $\mathbf{x} = (x, y)^T$ . The TuFF vector fields are given by  $\mathbf{v}_1 = (v_{11}, v_{12})^T$  and  $\mathbf{v}_2 = (v_{21}, v_{22})^T$ ; the dependency on  $\mathbf{x}$  implied. The extension to 3D is simple and follows from this derivation. We can rewrite  $\mathcal{E}_{reg}(\phi) = \int E_1(\phi)d\mathbf{x}$ , where  $E_1(\phi) = v_1|\nabla\phi|(\mathbf{x})\delta_\epsilon(\phi)$ . Then by calculus of variation, the Gateaux variation of  $\mathcal{E}_{reg}$  can be obtained as:

$$\frac{\delta\mathcal{E}_{reg}}{\delta\phi} = \frac{\partial E_1}{\partial\phi} - \frac{\partial}{\partial x}\left(\frac{\partial E_1}{\partial\phi_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial E_1}{\partial\phi_y}\right) \quad (28)$$

Since the proof is already shown in [36], we merely state the result as follows:

$$\frac{\delta\mathcal{E}_{reg}}{\delta\phi} = -v_1 \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \delta_\epsilon(\phi) \quad (29)$$

Similarly, we can write the evolution energy as  $\mathcal{E}_{evolve}(\phi) = \int E_2(\phi)d\mathbf{x}$ . This can be expanded as  $E_2(\phi) = A_1(\phi) + A_2(\phi)$ , where  $A_j(\phi) = \alpha_j \langle \mathbf{v}_j, \frac{\nabla\phi}{|\nabla\phi|} \rangle^2 H_\epsilon(\phi)$ . The dependency of  $\alpha, \phi$  and  $\mathbf{v}_j$  on  $\mathbf{x}$  is implied, and hence not mentioned explicitly.

We can further decompose  $A_1$  as

$$A_1(\phi) = -\alpha_1 \frac{(v_{11}\phi_x + v_{12}\phi_y)^2}{\phi_x^2 + \phi_y^2} H_\epsilon(\phi)$$

Let us denote  $\beta_j = \langle \mathbf{v}_j, \mathbf{n} \rangle$ , where the unit normal vector  $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ . Therefore, we can write  $A_1(\phi) = -\alpha_1 \beta_1^2 H_\epsilon(\phi)$ .

As earlier, we compute the Gateaux derivative as follows:

$$\frac{\partial A_1}{\partial\phi} = -\alpha_1 \beta_1^2 \delta_\epsilon(\phi) \quad (30)$$

Also, by simple algebraic manipulation, we obtain

$$\begin{aligned} \frac{\partial A_1}{\partial\phi_x} &= -2 \left[ \frac{\alpha_1 \beta_1}{|\nabla\phi|} v_{11} - \alpha_1 \left( \frac{\beta_1}{|\nabla\phi|} \right)^2 \phi_x \right] H_\epsilon(\phi) \\ \frac{\partial A_1}{\partial\phi_y} &= -2 \left[ \frac{\alpha_1 \beta_1}{|\nabla\phi|} v_{12} - \alpha_1 \left( \frac{\beta_1}{|\nabla\phi|} \right)^2 \phi_y \right] H_\epsilon(\phi) \end{aligned}$$

Therefore, we have

$$\frac{\partial}{\partial x} \left( \frac{\partial A_1}{\partial\phi_x} \right) = -2 \left[ \frac{\partial}{\partial x} (\eta_1 v_{11}) - \frac{\partial}{\partial x} \left( \eta_1 \beta_1 \frac{\phi_x}{|\nabla\phi|} \right) \right] \quad (31)$$

$$\frac{\partial}{\partial y} \left( \frac{\partial A_1}{\partial\phi_y} \right) = -2 \left[ \frac{\partial}{\partial y} (\eta_1 v_{12}) - \frac{\partial}{\partial y} \left( \eta_1 \beta_1 \frac{\phi_y}{|\nabla\phi|} \right) \right] \quad (32)$$

Where  $\eta_j = \frac{\alpha_j \beta_j}{|\nabla\phi|} H_\epsilon(\phi)$ . Therefore, by symmetry we compute

$$\frac{\partial}{\partial x} \left( \frac{\partial A_1}{\partial\phi_x} \right) + \frac{\partial}{\partial y} \left( \frac{\partial A_1}{\partial\phi_y} \right) = -2 \operatorname{div}[(\eta_j)(\mathbf{v}_j - \beta_j \mathbf{n})] \quad (33)$$

The Gateaux variation of  $\mathcal{E}_{\text{evolve}}$  can be obtained as:

$$\frac{\delta \mathcal{E}_{\text{evolve}}}{\delta \phi} = \frac{\partial E_2}{\partial \phi} - \frac{\partial}{\partial x} \left( \frac{\partial E_2}{\partial \phi_x} \right) - \frac{\partial}{\partial y} \left( \frac{\partial E_2}{\partial \phi_y} \right) \quad (34)$$

We now use gradient descent to find the local minima of the functionals. The regularizer force and evolution forces are given by  $\mathcal{F}_{\text{reg}} = -\frac{\delta \mathcal{E}_{\text{reg}}}{\delta \phi}$  and  $\mathcal{F}_{\text{evolve}} = -\frac{\delta \mathcal{E}_{\text{evolve}}}{\delta \phi}$  which leads to the following equations:

$$\mathcal{F}_{\text{reg}} = v_1 \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \quad (35)$$

and

$$\mathcal{F}_{\text{evolve}} = \sum_{j=1}^d \left( \alpha_j \beta_j^2 \delta_\epsilon(\phi) - 2 \operatorname{div} [\eta_j (\mathbf{v}_j - \beta_j \mathbf{n})] \right) \quad (36)$$

## REFERENCES

- [1] C. Koch and I. Segev, "The role of single neurons in information processing," in *Nature Neuroscience*, vol. 3. London, U.K.: Nature Pub. Group, 2000, pp. 1171–1177.
- [2] H. Cuntz, F. Forstner, J. Haag, and A. Borst, "The morphological identity of insect dendrites," *PLoS Comput. Biol.*, vol. 4, no. 12, p. e1000251, 2008.
- [3] J. Chen and B. G. Condron, "Branch architecture of the fly larval abdominal serotonergic neurons," *Develop. Biol.*, vol. 320, no. 1, pp. 30–38, 2008.
- [4] E. A. Daubert, D. S. Heffron, J. W. Mandell, and B. G. Condron, "Serotonergic dystrophy induced by excess serotonin," *Molecular Cellular Neurosci.*, vol. 44, no. 3, pp. 297–306, 2010.
- [5] H. Cuntz, M. W. H. Remme, and B. Torben-Nielsen, *The Computing Dendrite*, vol. 10. New York, NY, USA: Springer-Verlag, 2014, p. 12.
- [6] G. A. Ascoli, D. E. Donohue, and M. Halavi, "Neuromorpho.org: A central resource for neuronal morphologies," *J. Neurosci.*, vol. 27, no. 35, pp. 9247–9251, 2007.
- [7] K. A. Al-Kofahi *et al.*, "Median-based robust algorithms for tracing neurons from noisy confocal microscope images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 302–317, Dec. 2003.
- [8] J. Xie, T. Zhao, T. Lee, E. Myers, and H. Peng, "Anisotropic path searching for automatic neuron reconstruction," *Med. Image Anal.*, vol. 15, no. 5, pp. 680–689, 2011.
- [9] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson, "Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model," *Bioinformatics*, vol. 26, no. 12, pp. i38–i46, 2010.
- [10] H. Peng, F. Long, and G. Myers, "Automatic 3D neuron tracing using all-path pruning," *Bioinformatics*, vol. 27, no. 13, pp. i239–i247, 2011.
- [11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.
- [12] G. González, E. Türetken, F. Fleuret, and P. Fua, "Delineating trees in noisy 2D images and 3D image-stacks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2799–2806.
- [13] A. Rodriguez, D. B. Ehlenberger, P. R. Hof, and S. L. Wearne, "Three-dimensional neuron tracing by voxel scooping," *J. Neurosci. Methods*, vol. 184, no. 1, pp. 169–175, 2009.
- [14] S. L. Wearne, A. Rodriguez, D. B. Ehlenberger, A. B. Rocher, S. C. Henderson, and P. R. Hof, "New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales," *Neuroscience*, vol. 136, no. 3, pp. 661–680, 2005.
- [15] S. Mukherjee and S. T. Acton, "Vector field convolution medialness applied to neuron tracing," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 665–669.
- [16] S. Basu, B. Condron, A. Aksel, and S. T. Acton, "Segmentation and tracing of single neurons from 3D confocal microscope images," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 2, pp. 319–335, Mar. 2013.
- [17] S. Mukherjee, S. Basu, B. Condron, and S. T. Acton, "Tree2Tree2: Neuron tracing in 3D," in *Proc. 10th IEEE Int. Symp. Biomed. Imag.*, Apr. 2013, pp. 448–451.
- [18] S. Basu, A. Aksel, B. Condron, and S. T. Acton, "Tree2Tree: Neuron segmentation for generation of neuronal morphology," in *Proc. IEEE Int. Symp. Biomed. Imag.*, Apr. 2010, pp. 548–551.
- [19] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.
- [20] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam, "A broadly applicable 3D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 193–217, 2011.
- [21] H. Cai, X. Xu, J. Lu, J. Lichtman, S. P. Yung, and S. T. C. Wong, "Shape-constrained repulsive snake method to segment and track neurons in 3D microscopy images," in *Proc. 3rd IEEE Int. Symp. Biomed. Imag.*, Apr. 2006, pp. 538–541.
- [22] A. Narayanaswamy, Y. Wang, and B. Roysam, "3D image pre-processing algorithms for improved automated tracing of neuronal arbors," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 219–231, 2011.
- [23] A. Santamaría-Pang, C. M. Colbert, P. Saggau, and I. A. Kakadiaris, "Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2007, pp. 486–494.
- [24] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, no. 1, pp. 179–195, 1996.
- [25] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea, "A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes," *Med. Image Anal.*, vol. 13, no. 6, pp. 819–845, 2009.
- [26] L. M. Lorigo *et al.*, "CURVES: Curve evolution for vessel segmentation," *Med. Image Anal.*, vol. 5, no. 3, pp. 195–206, 2001.
- [27] A. Gooya, H. Liao, K. Matsumiya, K. Masamune, Y. Masutani, and T. Dohi, "A variational method for geometric regularization of vascular segmentation in medical images," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1295–1312, Aug. 2008.
- [28] A. Gooya, H. Liao, and I. Sakuma, "Generalization of geometrical flux maximizing flow on Riemannian manifolds for improved volumetric blood vessel segmentation," *Comput. Med. Imag. Graph.*, vol. 36, no. 6, pp. 474–483, 2012.
- [29] A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1565–1578, Dec. 2002.
- [30] Y. Shang *et al.*, "Vascular active contour for vessel tree segmentation," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 4, pp. 1023–1032, Apr. 2011.
- [31] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.
- [32] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2096–2106, Aug. 2007.
- [33] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995.
- [34] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.
- [35] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
- [36] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
- [37] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 1164–1170.
- [38] A. Yezzi, Jr., A. Tsai, and A. Willsky, "Binary and ternary flows for image segmentation," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Oct. 1999, pp. 1–5.
- [39] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1940–1949, Oct. 2008.
- [40] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 195–215, 2007.
- [41] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, nos. 285–296, pp. 23–27, 1975.
- [42] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, 1998, pp. 130–137.
- [43] X.-F. Wang, D.-S. Huang, and H. Xu, "An efficient local Chan-Vese model for image segmentation," *Pattern Recognit.*, vol. 43, no. 3, pp. 603–618, 2010.
- [44] J. L. Troutman, *Variational Calculus and Optimal Control: Optimization With Elementary Convexity*. New York, NY, USA: Springer-Verlag, 1995.

- [45] R. L. Graham and F. F. Yao, "Finding the convex hull of a simple polygon," *J. Algorithms*, vol. 4, no. 4, pp. 324–331, 1983.
- [46] S. T. Acton, "Fast algorithms for area morphology," *Digital Signal Process.*, vol. 11, no. 3, pp. 187–203, 2001.
- [47] K. M. Brown *et al.*, "The DIADEM data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 143–157, 2011.
- [48] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnol.*, vol. 28, no. 4, pp. 348–353, 2010.



**Suvadip Mukherjee** (S'11) is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, USA, where he is involved in image analysis methods under the supervision of Dr. S. T. Acton.

He received the bachelor's degree in electrical engineering from Jadavpur University, Kolkata, India, in 2008, and the master's degree in computer science from the Indian Statistical Institute, Kolkata, in 2011. His research interests include image segmentation and analysis techniques applied to biological and biomedical problems. He is also interested in other applied image and video processing research, such as image classification and feature identification techniques for CBIR and object tracking in videos.



**Barry Condron** is currently a Professor with the Department of Biology, University of Virginia (UVA), Charlottesville, VA, USA. He received the B.S. degree in mathematics and biochemistry from University College Cork, Cork, Ireland, in 1985, and the Ph.D. degree in genetics from the University of Utah, Salt Lake City, UT, USA, in 1991, where he was involved in gene regulation in a virus.

He held a post-doctoral position with the Dr. Kai Zinn's Laboratory, California Institute of Technology, Pasadena, CA, USA, where he was involved in how circuits for during development. He was involved in this field first at the School of Medicine, UVA, from 1997 to 2000, and has been at the Department of Biology, since 2000.



**Scott T. Acton** (F'13) is currently a Professor of Electrical and Computer Engineering and of Biomedical Engineering with the University of Virginia (UVA), Charlottesville, VA, USA. He received the M.S. and Ph.D. degrees from the University of Texas at Austin, Austin, TX, USA, and the B.S. degree from the Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.

His laboratory at UVA is called Virginia Image and Video Analysis (VIVA). They specialize in biological and biomedical image analysis problems. The research emphases of VIVA includes tracking, segmentation, representation, retrieval, classification, and enhancement. He has over 250 publications in the image analysis area including the books entitled *Biomedical Image Analysis: Tracking* and *Biomedical Image Analysis: Segmentation*. He serves as an Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING.

# Tubularity Flow Field—A Technique for Automatic Neuron Segmentation

Suvadip Mukherjee, *Student Member, IEEE*, Barry Condron, and Scott T. Acton, *Fellow, IEEE*

**Abstract**—A segmentation framework is proposed to trace neurons from confocal microscopy images. With an increasing demand for high throughput neuronal image analysis, we propose an automated scheme to perform segmentation in a variational framework. Our segmentation technique, called tubularity flow field (TuFF) performs directional regional growing guided by the direction of tubularity of the neurites. We further address the problem of sporadic signal variation in confocal microscopy by designing a local attraction force field, which is able to bridge the gaps between local neurite fragments, even in the case of complete signal loss. Segmentation is performed in an integrated fashion by incorporating the directional region growing and the attraction force-based motion in a single framework using level sets. This segmentation is accomplished without manual seed point selection; it is automated. The performance of TuFF is demonstrated over a set of 2D and 3D confocal microscopy images where we report an improvement of >75% in terms of mean absolute error over three extensively used neuron segmentation algorithms. Two novel features of the variational solution, the evolution force and the attraction force, hold promise as contributions that can be employed in a number of image analysis applications.

**Index Terms**—Confocal microscopy, neuron tracing, level set, vector field convolution.

## I. INTRODUCTION

**S**HAPe based neuron morphology analysis provides important cues in deciphering several functional behaviors of the brain of an individual [1]. Neuronal morphology has been studied to develop a functional model [2] for that neuron category, to analyze the branch patterns of serotonergic neurons [3], [4], or to correlate the structural aberrations in the dendritic arbors of an organism due to genetic factors or degenerative diseases like Alzheimer's [5].

An extensive shape based study of neuron morphology for an organism requires a comprehensive collection of digitally reconstructed neurons [6], which in turn demands intelligent processing tools to reconstruct neurons from the raw microscopy data. Recent advances in microscopy has enabled

Manuscript received June 6, 2014; revised September 22, 2014; accepted November 17, 2014. This work was supported by the National Science Foundation under Grant 1062433. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Olivier Bernard.

S. Mukherjee and S. T. Acton are with the C. L. Brown Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904 USA (e-mail: suvadip21@gmail.com; acton@virginia.edu).

B. Condron is with the Department of Biology, University of Virginia, Charlottesville, VA 22904 USA (e-mail: condron@virginia.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2378052

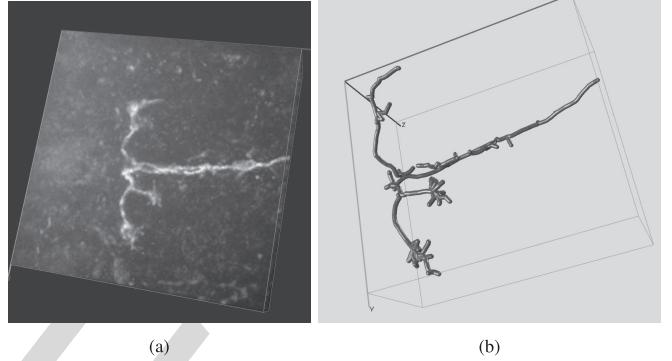


Fig. 1. (a) A *Drosophila* neuron imaged by confocal microscope. The background clutter is due to illuminated non neuronal filaments. (b) The corresponding reconstruction is shown.

imaging scientists to acquire substantial quantity of images. With more than 20,000 neurons in the brain of the fruit fly *Drosophila* and even more for other species such as mice and humans, the task of automated, high throughput neuro-image analysis is both critical and daunting.

Given the complexity of the problem, it is not surprising that automated neuron segmentation still remains a critical open problem in the field. State of the art neuron segmentation methods rely heavily on manual interaction to generate the morphological reconstruction. Complicated branching patterns of the neurons pose challenge to automated tracing. Moreover, the confocal microscopy images are, in general, degraded by low signal to noise ratio and non uniform illumination of the neurites which leads to fragmented appearance of the object. Fig. 1(a) shows a 3D neuron image of *Drosophila* imaged using a laser scanning confocal microscope. Topologically, a neuron resembles a tree, with multiple filamentous branches emerging from a single cell body. This is shown in Fig. 1(b), which is a digital reconstruction of (a), obtained using our algorithm. In this paper, we present an automated neuron segmentation method, based on an energy minimization framework. Segmentation results on GFP-labeled *Drosophila* neurons, imaged using confocal microscope are studied to demonstrate the efficacy of our technique.

## A. Background

In this section we briefly review some relevant research in neuron segmentation. In this paper, we are interested in segmenting neurons from confocal microscopy images only. Therefore, techniques which use other imaging modalities

(such as electron microscopy) are excluded from this discussion.

We can broadly categorize the neuron segmentation schemes in two basic approaches. The first set of methods use user defined (or automatically detected) initial seed points to perform tracing. The second category of algorithms avoid seed initialization and perform segmentation globally.

Manual seed selection has the advantage that the segmentation region is identified a priori by an expert. This introduces locality in processing, which results in higher processing speed. Typically such algorithms generate the neuronal tree from semi-automatically initialized seed points on the neurite centerlines. Al-Kofahi *et al.* [7] used the medial response of multiple directional templates to determine the direction to generate successive seed points along the neuron medial axis. This local tracing method shows good performance in high-contrast images, but requires continuity in the neuron branches for reliable segmentation.

Segmentation performance can be considerably improved if the seed points are selected manually. These seeds are then treated as nodes in a graph, and segmentation is performed using graph theoretic algorithms. When seed selection is done automatically, a pruning step is generally used to eliminate the non-neuronal points. With this optimal set of seeds, the methods in [8]–[10] establish connectivity between the nodes using a shortest path algorithm [11], by suitably selecting the weights on the graph edges. Fast and accurate segmentation is possible using the above mentioned approaches if the neuron structure is morphologically simple and the image noise level is low. Gonzalez *et al.* [12] introduced a graph theoretic technique to delineate the optimal neuronal tree from an initial set of seeds by computing a K-Minimum Spanning Tree. An approximate solution to this NP-hard problem was realized by minimizing a global energy function in a linear integer programming framework. However, due to its greedy nature, the algorithm may converge to undesired local minima.

We hypothesize that seed based techniques are useful if the imaged neurons are not too complicated structurally. In such scenarios, where manual seed selection is easy, reliable segmentation can be obtained. However, since automatically choosing the correct set of seed points is still an open problem, it is difficult to use the above mentioned techniques for high throughput, no intervention analysis. Also, since proper selection of seeds points is instrumental in these methods, the segmentation accuracy is sometimes compromised if a sub-optimal set of points is chosen. Furthermore, the connectivity analysis between the seeds assume uniform signal intensity, and noise and low contrast in the images may degrade the segmentation quality.

In contrast to the seed based local techniques, traditional segmentation approaches are more global, typically requiring an initial pre-processing of the image followed by a specialized segmentation step. Although a global approach may suffer from expensive computation, they are more suitable for neurite junction and end point detection. Typically, such methods rely on a four stage processing pipeline – enhancement, segmentation, centerline detection and post processing. The voxel scooping algorithm proposed in [13] assumes

tubular structure of the neurite filaments and iteratively searches for voxel clusters in a manner similar to region growing. A pruning step is then deployed to eliminate spurious end nodes. A similar region growing method is implemented in the popular automatic neuron tracing tool Neuronstudio [14]. The segmentation step is generally followed by a centerline detection [2], [15] stage to detect the medial axis of the segmented structure. In many cases further smoothing of the medial axis is performed by spline fitting [16]. Since such methods do not rely on human intervention, it is evident that the segmentation quality would depend heavily on the initial segmentation, which may be affected by the noise and clutter in the images.

Tree2Tree [16] and its variants [17], [18] propose to solve the neuron segmentation problem in a graph theoretic framework. However, unlike traditional seed selection approaches, where manually initialized points are treated as the nodes of the graph, an initial segmentation algorithm is devised to produce disjoint connected components. Connectivity between the components is analyzed based on their separating distance and orientation, which determines the weights of the graph edges to perform segmentation using a minimum spanning tree approach.

Although the primary contribution of Tree2Tree is to connect the fragmented neurite segments automatically, this connectivity analysis relies on heavily on the initialization. Noise and clutter in the images create undesired artifacts in the global segmentation, resulting in loss of structural information. Moreover, linking the components based on their relative geometric orientation requires computation of the leaf-tangents from the object centerlines, which is sensitive to the irregularities of the neurite surface. Furthermore, elimination of false nodes from the neuronal tree is difficult, and ultimately requires further manual parameter tuning.

Segmentation based on active contours [19] have also been proposed [20], [21] to directly obtain the neuron centerline, without performing a global thresholding. The algorithm proposed by Wang *et al.* [20] involves evolution of an open ended snake guided by a force field that encourages the neuron trace to lie along the filament centerline. A pre-processing step based on tensor voting [22] was introduced to enhance the vascular structure of the neurites. Combined with a post-processing step to eliminate false filaments, this method is efficient in segmenting neuronal structures from low SNR confocal stacks. However, due to the inability of parametric active contours to naturally handle topological changes such as object merging, neurite branch point detection depends requires a non-trivial post processing to determine snake merging at the junctions. Santamaria-Pang *et al.* [23] use a multistage procedure for detection of tubular structures in multi-photon imagery, which includes a pre-filtering stage to identify the filaments based on supervised learning. This requires offline learning of the model parameters and prior knowledge about the vessel appearance information, which necessitates a set of accurate training examples and demands extensive human involvement to generate the ground truth. Zhou *et al.* [24] propose a variational framework based on geodesic active contours to identify neurite branches from

two photon microscopy. This strategy is effective when the edge information is reliable, and hence depends on efficient pre-processing to eliminate image irregularities. However, both these methods do not deploy additional schemes to identify and analyze the broken neurite fragments in their model, and hence it demands a specialized post-processing step.

The medical imaging community has performed substantial research in developing algorithms to detect and segment filamentous shapes in non-microscopy medical images [25]. The CURVES algorithm by Lorigo *et al.* [26] evolves a 1D curve along a 3D vessel centerline guided by the curvature of a 1D curve.

Gooya *et al.* [27] developed an elegant and generalizable regularization methodology to enhance the performance of the popular geometric curve evolution methods. The method allows for anisotropic curve propagation which minimizes contour leakage when vessel edge information is weak. The only apparent downside of this technique is that the ultimate solution somewhat depends on the shape of the initialized contour. Another recent work by Gooya *et al.* [28] generalizes the flux maximizing flow [29] on Riemann manifolds and uses a vessel enhancing tensor, which improves segmentation when edge information is noisy.

Shang *et al.* [30] propose a vessel tracing method where wider vessels are first segmented using a region based criteria. Then the eigenvectors of the hessian matrix are utilized to derive a geometric flow equation to segment the thinner vessels. The mathematical formulation of the problem involves only a single eigenvector (the one along axial direction of vessel) for curve evolution, and hence is unsuitable for detecting thicker vessels. As we will show later, our formulation presents an unified framework to segment vessels of heterogeneous thickness by utilizing information from all three principal vessel directions (axial and orthogonal). Also, since the above mentioned methods are tailored for applications such as MRA, CT etc, they require further modifications to satisfy the demands of confocal microscopy where noise and clutter is present in a significantly higher proportion.

### B. Our Contribution

We focus on reconstructing single neuron from a confocal microscope image. A robust neuron segmentation scheme needs to address two primary issues. First, the technique should be suited to identify neuron structures from the noisy confocal images. Second, it should be adept at handling the local structure discontinuities (see Fig. 2) resulting from imaging artifacts. We propose a solution to this segmentation problem using a variational framework driven by level sets. The level set evolution is guided by minimizing an application specific energy functional. A tubularity flow field (TuFF) is computed by utilizing the local tubularity of the neurites which guides the segmentation procedure by encouraging curve evolution along the length (axis) and the thickness of the tubular neurites. A specialized local attraction force is also designed to accommodate the intensity variations in the images of neurite structures, thus presenting an unified framework to naturally link the fragmented structures. Our method does

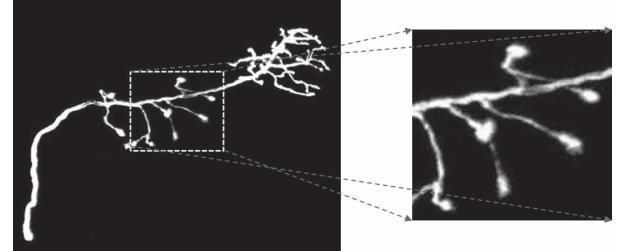


Fig. 2. Maximum intensity projection of a neuron imaged by a confocal microscope. The image suffers from contrast non-uniformity, including gaps that lead to breaks in the segmented neurite structure. The effect is most pronounced in the region bounded by yellow dashed box, magnified here for improved viewing.

not rely on an initial set of seed-points for segmentation; it is automatic. Moreover, it does not require non-trivial post-segmentation analysis to link the disjoint segments. This is performed naturally by using the local attraction force in a level set paradigm. This enables us to connect disunited structures, even if the underlying signal intensity is significantly low. The problem formulation and the design process of the attraction force are discussed in the following sections.

## II. TUBULARITY FLOW FIELD FOR NEURON SEGMENTATION

Active contours or snakes [19], [31]–[34] are an attractive choice for image segmentation due to their ability to elastically delineate object boundaries with sub-pixel accuracy and to incorporate signal and shape based constraints to assist segmentation. Geometric active contours [24], [33]–[37] are appealing due to their inherent ability to deal with topological changes of the foreground in segmentation. Unlike their parametric counterparts which perform segmentation by explicitly updating the position of a parametric curve, geometric active contours perform curve evolution implicitly, by evolving a higher dimensional embedding function  $\phi$ .

Let  $f : \Omega \rightarrow \mathbb{R}$  be an image defined on the continuous domain  $\Omega \subset \mathbb{R}^d$ , where  $d$  is the dimension of the image. In a variational paradigm, implicit motion of the zero level set of  $\phi$  is obtained by minimizing an energy functional  $\mathcal{E}(\phi)$  [24], [36]–[39]. The level set function  $\phi$  is defined to be positive inside the zero level set and negative outside it. The zero level sets define the object boundaries. The energy functional design is application dependent, and is a major engineering aspect for all variational level set based methods. Such methods are popular since the energy functional gives intuition for the segmentation procedure. Furthermore, various shape and smoothness constraints can be easily incorporated to further assist segmentation [33], [40]. For this problem of neuron segmentation, we need to design the energy functional such that it would encourage curve propagation in the filamentous regions of the image, while avoiding the non-tubular structures. Also, the segmentation should allow sufficient local processing to avert fragmented segments in the solution, which may appear as a consequence of using global threshold selection schemes like that of Otsu [41] or

methods assuming piecewise constant intensity models of [36]. We avoid this problem by introducing a local shape prior by way of a specially designed tubularity flow vector field and a local attraction force to link nearby neuronal fragments.

### A. Tubularity Flow Field (TuFF)

As mentioned previously, we assume a locally tubular model for neurite segmentation. The key ingredient of our algorithm is to use this tubularity information to evolve the level set function. A set of vector fields called the tubularity flow field (TuFF) is used to drive the active contour towards the object boundary. The tubularity measure at a point  $\mathbf{x} \in \Omega$  in the image can be obtained by examining the hessian matrix of the gaussian smoothed image over a set of scales. The hessian of the  $d$ -dimensional image  $f(\mathbf{x})$  at a position  $\mathbf{x}$  and scale  $\sigma$  is the square matrix  $H_\sigma(\mathbf{x}) = [h]_{i,j}$  ( $1 \leq i, j \leq d$ ,  $\mathbf{x} \in \Omega$ ) which is given by

$$h_{i,j} = \frac{\partial^2 G(\sigma)}{\partial x_i \partial x_j} * f(\mathbf{x}) \quad (1)$$

where  $\mathbf{x}$  is the  $d$ -dimensional vector  $\mathbf{x} = (x_1, \dots, x_d)^T$ ,  $G(\sigma)$  is the zero mean normalized Gaussian kernel with variance  $\sigma^2$ . Here  $d = 2$  or  $3$  for 2D or 3D images respectively.

Since the imaged neurons are brighter than the background, one can analyze the scale space hessian matrix to obtain evidence of tubularity at a particular image position. Ideally, at a position  $\mathbf{x} \in \Omega$ , 3D tubular structure can be characterized by three principal directions: (i) an axial direction along which the second derivative is negligible, and (ii) two orthogonal directions along which the second derivative magnitude is significant. These directions are given by the orthonormal set of eigenvectors  $\{\mathbf{e}_1(\mathbf{x}), \mathbf{e}_2(\mathbf{x}), \mathbf{e}_3(\mathbf{x})\}$ . The corresponding second derivative magnitudes can be obtained from the respective eigenvalues  $|\lambda_1(\mathbf{x})| \leq |\lambda_2(\mathbf{x})| \leq |\lambda_3(\mathbf{x})|$ .

Analysis of these eigenvalues is essential to preserve the tubular portions of neurons, while suppressing the background clutter [16], [42]. The non tubular clutter are present in most confocal microscopy images due to photon emission from non neuronal tissues and are often referred to as *structure noise*. These structure noise may be bright disc shaped non-neuronal segments in 3D images or blob-like structures. We would like to mention that from here onward we would present our analysis for the 3D case only for better readability. However, the results are easily applicable to the 2D case and there exists an equivalent 2D version of the solutions.

It may be observed that for a voxel  $\mathbf{x}$  to belong to a tube, the eigenvalues of its hessian matrix (computed at scale  $\sigma$ ) should satisfy the following criteria:

$$\begin{aligned} |\lambda_1(\mathbf{x})| &\approx 0 \\ |\lambda_2(\mathbf{x})| &\gg |\lambda_1(\mathbf{x})|, |\lambda_3(\mathbf{x})| \gg |\lambda_1(\mathbf{x})| \\ |\lambda_2(\mathbf{x})| &\approx |\lambda_3(\mathbf{x})| \end{aligned} \quad (2)$$

Also, since the neurites are brighter than the background, we have  $\lambda_2(\mathbf{x}) < 0$  and  $\lambda_3(\mathbf{x}) < 0$ .

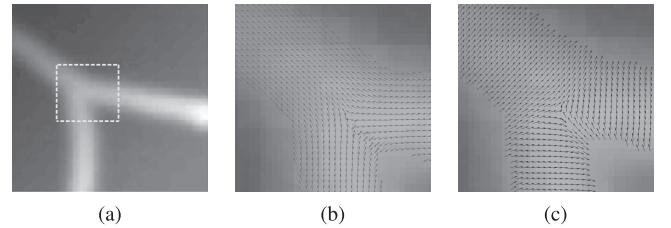


Fig. 3. Illustrative example of the weighted TuFF. A simple tubular structure is shown in (a). (b) The weighted axial vector field and (c) The weighted orthogonal vector field for the sub image enclosed in the yellow rectangle. Weights of the vector fields are computed using (11, 12). Image courtesy of [43].

**1) Scale Selection:** Since neurites vary in thickness, a scale space analysis is required to capture the variability in their width. If  $S = \{\sigma_{min}, \dots, \sigma_{max}\}$  denotes the scale space, for  $\sigma \in S$ , the tubularity measure or *vesselness* score [16] for a 3D image at  $\mathbf{x} \in \Omega$  can be written as

$$N_\sigma(\mathbf{x}) = \begin{cases} \frac{|\lambda_1(\mathbf{x}) - \lambda_2(\mathbf{x})|^2}{|\lambda_1(\mathbf{x})||\lambda_2(\mathbf{x}) - \lambda_3(\mathbf{x})|} & \text{if } \lambda_2(\mathbf{x}), \lambda_3(\mathbf{x}) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The optimal scale  $\sigma^*$  at  $\mathbf{x} \in \Omega$  and its corresponding vesselness score  $N(\mathbf{x})$  is computed as follows:

$$\sigma^*(\mathbf{x}) = \operatorname{argmax}_{\sigma \in S} N_\sigma(\mathbf{x}) \quad (4)$$

$$N(\mathbf{x}) = \max_{\sigma \in S} N_\sigma(\mathbf{x}) \quad (5)$$

The scale space vesselness response  $N(\mathbf{x})$  assumes higher value at locations of local tubularity over non-filamentous positions. It should be noted that (5) yields evidence of the presence of a neurite by suppressing the non-filamentous structures, thus introducing a mechanism for dealing with the structure noise.

Given  $H_{\sigma^*}(\mathbf{x})$ , the hessian matrix of the image  $f(\mathbf{x})$  at the optimal scale  $\sigma^*(\mathbf{x})$ , we can compute the TuFF. For a 3D image, the TuFF consists of a vector field  $\mathbf{v}_1(\mathbf{x})$  along the vessel axial direction and two vector fields  $\mathbf{v}_2(\mathbf{x})$  and  $\mathbf{v}_3(\mathbf{x})$  whose non-zero components are orthonormal to the axial field  $\mathbf{v}_1(\mathbf{x})$  (Fig. 3). Formally, this can be computed as

$$\mathbf{v}_k(\mathbf{x}) = \begin{cases} \mathbf{e}_k^*(\mathbf{x}) & \text{if } \lambda_1^*(\mathbf{x}) \approx 0 \text{ and } \lambda_2^*(\mathbf{x}), \lambda_3^*(\mathbf{x}) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$\mathbf{e}_k^*(\mathbf{x})$  denotes the normalized eigenvector corresponding to the eigenvalue  $\lambda_k^*(\mathbf{x})$  of the hessian matrix  $H_{\sigma^*}(\mathbf{x})$  such that  $|\lambda_1^*(\mathbf{x})| \leq |\lambda_2^*(\mathbf{x})| \leq |\lambda_3^*(\mathbf{x})|$  ( $\forall \mathbf{x} \in \Omega, k = 1, 2, 3$ ). In the following subsections, we show how TuFF can be incorporated in a level set framework to perform neuron segmentation.

### B. Neuron Segmentation Using TuFF

Our method performs segmentation via minimization of the energy functional  $\mathcal{E}(\phi)$ . This energy functional can be

mathematically written as:

$$\mathcal{E}(\phi) = \mathcal{E}_{reg}(\phi) + \mathcal{E}_{evolve}(\phi) + \mathcal{E}_{attr}(\phi) \quad (7)$$

$$\mathcal{E}_{reg}(\phi) = v_1 \int_{\Omega} |\nabla \phi(\mathbf{x})| \delta(\phi) d\mathbf{x} \quad (8)$$

$$\mathcal{E}_{evolve}(\phi) = - \int_{\Omega} \sum_{i=1}^d \alpha_i(\mathbf{x}) \langle \mathbf{v}_i(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle^2 H(\phi) d\mathbf{x} \quad (9)$$

Here  $\mathcal{E}_{reg}$  and  $\mathcal{E}_{evolve}$  are the energy functionals corresponding to the smoothness of the curve and the curve evolution respectively. The functional  $\mathcal{E}_{attr}$  contributes towards creating a local attraction energy. This attraction energy is to be designed in a manner such that minimizing it would result in a force field to join the local, disjoint neuron fragments. For our application, we do not define the attraction energy explicitly; instead, we compute the attraction force resultant from the energy (see Section II-E).

The vector  $\mathbf{n}(\mathbf{x}) = \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|}$  denotes the inward normal unit vector to the level sets of  $\phi$ .  $\langle \cdot, \cdot \rangle$  is the Euclidean inner product operator. The positive scalar  $v_1$  in (8) contributes to the smoothness of the zero level curve. The weighing parameter  $\alpha_i$  determines the contribution of the orthogonal and axial components of the TuFF in curve evolution. Choice of  $\alpha_i$  is an important aspect which would be discussed shortly.

In practice, the ideal Dirac delta function  $\delta(\phi)$  and the Heaviside function  $H(\phi)$  are replaced by their regularized counterparts  $\delta_\epsilon(\phi)$  and  $H_\epsilon(\phi)$  respectively as defined in [36]. Regularization of the functions is controlled by the positive parameter  $\epsilon$ . The regularizing energy term  $\mathcal{E}_{reg}$  in (8) constrains the length of the zero level curve of  $\phi$ . The amount of smoothing is controlled by the parameter  $v_1 \geq 0$ . Using a small value of  $v_1$  has the effect of encouraging presence of smaller, disjoint objects in the final solution. We report the actual values of  $v_1$  while discussing the implementation details.

### C. Discussion of Curve Evolution via TuFF

The essence of our technique lies in the design of curve evolution energy  $\mathcal{E}_{evolve}$  in (9). In absence of the attraction force energy, the level curve evolution (which results from minimizing the energy term (9)) depends on the contribution of the axial and orthogonal components of the tubularity flow field. The design of the functional (9) is such that the axial vector field component  $\mathbf{v}_1$  is responsible for propagating the curve to fill out the vessel thickness. Or in other words, the axial field promotes curve evolution in a direction perpendicular to itself. Identically, the orthogonal components  $\mathbf{v}_2, \mathbf{v}_3$  encourage curve propagation in a direction perpendicular to themselves, i.e. along the axis of the neuron filaments. Let us illustrate this phenomenon by using a 2D synthetic image containing a single tubular structure (Fig. 5).

*1) Effect of the Axial Component of TuFF:* Maximizing the total squared inner product  $\int_{\Omega} \alpha_1(\mathbf{x}) \langle \mathbf{v}_1(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle^2 H_\epsilon(\phi)$  (or minimizing its negative) with respect to the embedding function  $\phi$  results in maximally aligning the inward normal vectors  $\mathbf{n}(\mathbf{x})$  of the zero level sets of  $\phi$  and its inner isocontours with the axial flow field  $\mathbf{v}_1(\mathbf{x})$ . As shown in the first row

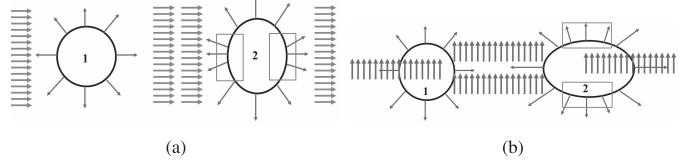


Fig. 4. Illustration of curve evolution due to (a) axial component  $\mathbf{v}_1$  and (b)  $\mathbf{v}_2$ . Note how the contour should change to align the surface normals (shown in red arrow) with the vector fields (shown in green and purple arrows respectively) in order to minimize the evolution energy. The initial curve is marked as 1. The evolution forces create the new curves 2. Note how the curves assume elliptical shape to align the level set normals with the vector fields. The normal vectors are maximally aligned in the region enclosed by the rectangles.

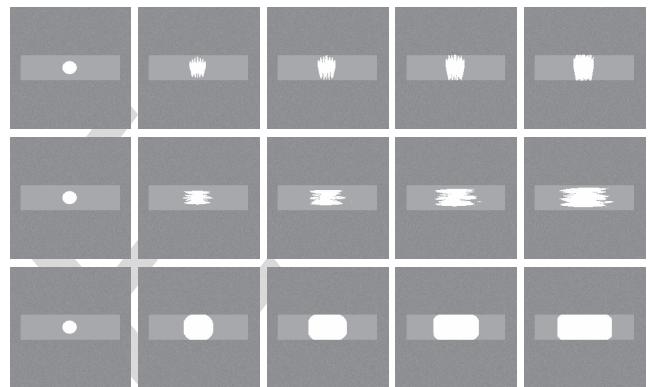


Fig. 5. Segmentation steps for the synthetic rectangular are shown. The first row shows curve evolution due to axial component of TuFF, i.e.  $\alpha_2 = 0$ . The second row shows surface evolution for  $\alpha_1 = 0$ .  $v$  is set to 0 for both the cases. The bottom row displays smooth curve propagation with  $\alpha_0 = 1$ ,  $\alpha_1 = 5$ ,  $\sigma = 3$  and  $v = 0.02$ .

of Fig. 5, this requires the level sets of  $\phi$  to be re-aligned such that the normal vectors  $\mathbf{n}(\mathbf{x})$  aligns itself with the axial field  $\mathbf{v}_1(\mathbf{x})$ . This results in curve evolution in a direction orthogonal to the vessel axis, causing elongation of the level curves along the vessel width.

*2) Effect of the Orthogonal Component of TuFF:* Using a similar argument, maximizing the second term corresponding to the orthogonal component in (9) performs alignment of the inward normal vectors with the vector field  $\mathbf{v}_2(\mathbf{x})$ , creating an elongation force which allows the level curves to propagate along the vessel axis. This is shown in the second row of Fig. 5. For intuitive understanding of the above mentioned phenomenon, Fig. 4(a) and (b) is provided to graphically demonstrate how the curve evolution is affected by the axial and the normal components of TuFF.

*3) Effect of the Vector Field Weights:* Ideally, the parameters  $\alpha_i(\mathbf{x})$ ,  $i = 1, \dots, d$ , should be chosen such that curve propagation is discouraged outside the tubular neurite segments, so as to avoid leakage into the background. i.e. for a voxel  $\mathbf{y}$  with low vesselness score, we require  $\alpha_i(\mathbf{y}) \approx 0$ , for  $i = 1, \dots, d$ . Moreover, since the neurites are elongated structures, it is desired that the contour evolution be more pronounced near the filament centerline than at the edges. This can be stated as

$$\frac{\alpha_j(\mathbf{x})}{\alpha_1(\mathbf{x})} \geq 1 \quad (j = 2, \dots, d) \quad \text{and} \quad \alpha_1(\mathbf{x}), \dots, \alpha_d(\mathbf{x}) > 0 \quad (10)$$

$\forall \mathbf{x} \in \Omega$ . Respecting the above constraints, we propose the following functions for choosing the parameters.

$$\alpha_1(\mathbf{x}) = N(\mathbf{x}) \quad (11)$$

$$\alpha_j(\mathbf{x}) = N(\mathbf{x}) \left( a_0 + \exp \left( -\frac{|\nabla_\sigma f(\mathbf{x})|}{a_1} \right)^2 \right) \quad (12)$$

$\forall \mathbf{x} \in \Omega$  and  $j = 2, \dots, d$ .  $N(\mathbf{x})$  is the vesselness score which is obtained from (5).

Let us discuss the isotropic case, when  $a_0 = 1$  and  $a_1 \rightarrow \infty$ . Since the unit normal vector  $\mathbf{n}(\mathbf{x})$  lies in the vector space spanned by  $\{\mathbf{v}_i(\mathbf{x})\}$ , it can be written as  $\mathbf{n}(\mathbf{x}) = \sum_{i=1}^d m_i \mathbf{v}_i(\mathbf{x})$ . This reduces (9) to

$$\mathcal{E}_{evolve}(\phi) = - \int_{\Omega} N(\mathbf{x}) \sum_i \langle \mathbf{v}_i(\mathbf{x}), \sum_j m_j \mathbf{v}_j(\mathbf{x}) \rangle^2 H(\phi) d\mathbf{x}$$

Since the eigenvectors are orthonormal,  $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 1$  for  $i = j$  and 0 otherwise. Also, since  $|\mathbf{n}(\mathbf{x})| = 1$ , we have  $\sum_i m_i^2 = 1$ . Using this relation, we obtain  $\sum_i \langle \mathbf{v}_i(\mathbf{x}), \sum_j m_j \mathbf{v}_j(\mathbf{x}) \rangle^2 = 1$ . This reduces the evolution equation to

$$\mathcal{E}_{evolve}(\phi) = - \int_{\Omega} N(\mathbf{x}) H(\phi) d\mathbf{x} \quad (13)$$

The energy functional in (13) when minimized performs segmentation via vesselness weighted isotropic region growing along the neuron segments. Leakage of the contour outside vessel boundaries is prohibited by the vessel indicator function  $N(\mathbf{x})$  which provides evidence of tubularity by assuming higher value for the tubular objects than non tubular background.

With the discussion of the isotropic case, it is now easy to visualize the effect of the weights on curve evolution. From our previous discussion, we recall that  $\alpha_1$  and  $\{\alpha_j, j \neq 1\}$  influence curve propagation along the vessel width and axial direction respectively.  $|\nabla_\sigma f(\mathbf{x})|$  denotes the gradient magnitude of the image  $f(\mathbf{x})$ , which is filtered by a Gaussian kernel with variance  $\sigma^2$ . Since this term is high at the vessel boundaries and end points, the negative exponential term in (12) ensures higher response at regions near the vessel centerline. The tuning parameters  $a_0 \geq 1$  and  $a_1$  determine the relative influence of the axial curve motion to the motion along the vessel width. In other words, in an anisotropic setting, (12) suggests that the level curves evolve with higher curvature near the vessel medial axis than at the edges, which percolates to the isotropic case when  $a_0 = 1$  and  $a_1 \rightarrow \infty$ .

Since the neurite filaments are predominantly thin, elongated structures, we observe that the isotropic case yields sufficiently appropriate segmentation results when the initialized zero level set encompasses the filament width. Nevertheless, the proposed framework in (7) is general as demonstrated in Fig. 5, and is applicable to segmentation problems where vessel thickness is significant and the initialized zero level contour does not fill out the vessel width completely. This is in contrast to the approach in [30], where segmentation of thicker vessels needs separate treatment. The bottom row of Fig. 5 shows the successive motion of the level sets for the above mentioned choice of alpha. It is observed that the evolving level set encompasses both the width and length of the vascular structure.

#### D. Level Set Evolution Using Gradient Descent

The energy functional in (7) can be minimized using variational calculus techniques [44]. Taking the Gâteaux variation of  $\mathcal{E}(\phi)$  with respect to  $\phi$ , we obtain from (7)

$$\frac{\delta \mathcal{E}}{\delta \phi} = \frac{\delta \mathcal{E}_{reg}}{\delta \phi} + \frac{\delta \mathcal{E}_{evolve}}{\delta \phi} + \frac{\delta \mathcal{E}_{attr}}{\delta \phi} \quad (14)$$

$\phi$  can be iteratively updated using gradient descent technique, i.e. setting  $\frac{\delta \mathcal{E}}{\delta \phi} = -\frac{\partial \phi}{\partial t}$  with  $t$  denoting the pseudo time parameter for the iterative scheme:

$$\frac{\partial \phi}{\partial t} = \mathcal{F}_{reg}(\mathbf{x}) + \mathcal{F}_{evolve}(\mathbf{x}) + \mathcal{F}_{attr}(\mathbf{x}) \quad (15)$$

$\mathcal{F}_{reg}$  and  $\mathcal{F}_{evolve}$  are the forces due to the regularizing energy and the evolution energy functional respectively. These forces are derived by solving the Euler-Lagrange equation for level set evolution as:

$$\mathcal{F}_{reg}(\mathbf{x}) = v_1 \operatorname{div} [\mathbf{n}(\mathbf{x})] \delta_\epsilon(\phi) \quad (16)$$

$$\begin{aligned} \mathcal{F}_{evolve}(\mathbf{x}) = & \delta_\epsilon(\phi) \sum_{j=1}^d \{\alpha_j(\mathbf{x}) \beta_j^2(\mathbf{x})\} \\ & - 2 \operatorname{div} \left[ \sum_{j=1}^d \eta_j(\mathbf{x}) (\mathbf{v}_j(\mathbf{x}) - \beta_j(\mathbf{x}) \mathbf{n}(\mathbf{x})) \right] \end{aligned} \quad (17)$$

The coefficients  $\beta_j$  and  $\eta_j$  are defined as follows:

$$\beta_j(\mathbf{x}) = \langle \mathbf{v}_j(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle \quad (18)$$

$$\eta_j(\mathbf{x}) = \frac{\alpha_j(\mathbf{x}) \beta_j(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} H_\epsilon(\phi) \quad (19)$$

The derivation details are shown in the Appendix.

#### E. Local Attraction Force Design

The attraction force  $\mathcal{F}_{attr}$  in (15) is introduced to accommodate the signal intensity variation (and signal loss) across the neurite branches, as shown in Fig. 2. Such signal attenuation introduces unwarranted discontinuities in the filamentous objects, resulting in disjoint fragments. Also, discontinuities may be present at the neurite junctions and noisy regions due to the nonlinear response of the vesselness function in (5). In such a scenario, the TuFF based evolution energy term in (9) is not adequate to perform segmentation alone. This insufficiency motivates the inclusion of an attraction force component. Designing this attraction force requires analysis of the connected components at each time epoch of level set propagation. At a time  $t$  for evolution of the level set function  $\phi(\mathbf{x}, t)$ , the set of connected components  $\mathcal{C}(t)$  can be obtained as

$$\begin{aligned} \mathcal{C}(t) &= H(\phi(\mathbf{x}, t)) \\ \text{where } H(y) &= \begin{cases} 1 & \text{for } y \geq 0 \\ 0 & y < 0 \end{cases} \end{aligned} \quad (20)$$

The set of connected components  $\mathcal{C}(t) = \{c_1, \dots, c_p\}$  represents the binary segmentation at time  $t$ , which consists of  $p \geq 1$  disjoint connected components. Note that this binarization does not require a sophisticated segmentation, since the binary components are obtained by extracting the

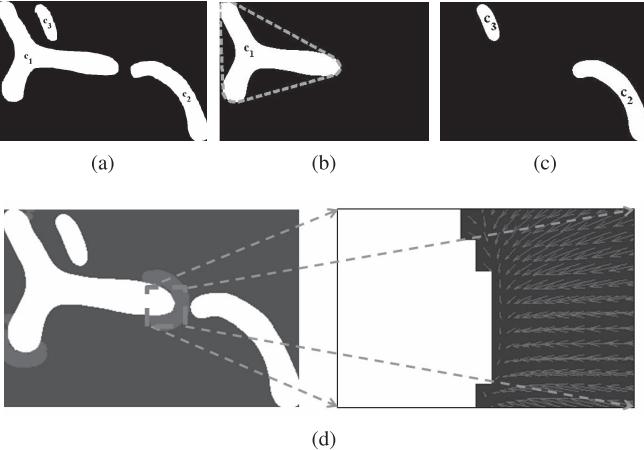


Fig. 6. (a) Set of disjoint connected components  $\{c_1, c_2, c_3\}$  at a particular step of iteration. (b) shows a parent component, the green dotted line marking its convex hull. The remaining children are shown in (c). (d) shows the attraction force obtained via (23) in red arrows, magnified for visual clarity.

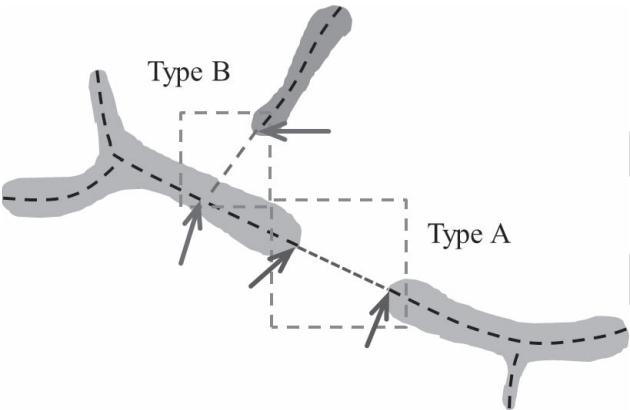


Fig. 7. Two types of discontinuities between the disjoint components. The *Type A* discontinuity can be resolved by joining the end points of the center lines of the respective branches. *Type B* is more difficult, where discontinuity occurs between a branch end point and an intermediate point on the centerline of the other branch.

interior of the zero level sets of the embedding function. Each disjoint component  $c_j$  is a potential candidate or a *parent* which has the capability of attracting the remaining *children*  $c_k$ ,  $k \neq j$ ,  $(j, k = 1, \dots, p)$ . This is illustrated in Fig. 6(a)–(c), where the component  $c_1$  acts as a parent component and  $c_2$  and  $c_3$  are the children.

1) *Candidate Points for Attraction Force Field*: The primary responsibility of the attraction force is to enable the propagating contour surface to attach itself to local disjoint fragments. However, not all points on the connected components are candidates for creating the attraction force. This is because in a majority of the prevalent discontinuities, at least one of the two disconnected portions are likely to be joined via boundary points which represent region of high curvature (see Fig. 7). If we denote the boundary of a component  $c_j$  by  $\delta c_j$ , to enable a parent to attract a child, we need to design an attraction field which is generated by a set of candidate points lying on the parent boundary. Therefore, for a parent component  $c_j$ , a point  $\mathbf{y} \in \delta c_j$  belongs to the candidate set if  $\mathbf{y}$  is a point the convex

hull [45]  $\mathcal{H}_j$  of  $c_j$  (Fig. 6(b)). Formally, the candidate point set  $\mathcal{M}_j$  for the connected component  $c_j$  is defined as

$$\mathcal{M}_j = \{\mathbf{y} \in \delta c_j : \exists \mathbf{x}_j \in \mathcal{H}_j \text{ s.t. } \|\mathbf{y} - \mathbf{x}_j\|_2 \leq \Delta\} \quad (21)$$

$\Delta$  is a positive parameter that includes local boundary coordinates of the neighboring points on the convex hull.

2) *Attraction Force Field Design*: The candidate set of points for a parent component is responsible for generating a force field capable of attracting the candidate children towards itself for potential merging. This needs to be designed such that the attraction field vectors point toward the region of interest, which is the parent candidate point set for this purpose. We show that an efficient solution may be obtained by using vector field convolution (VFC) to create the attraction force field.

VFC [32] is a technique primarily designed to create smooth external force field for parametric active contours. The specially designed vector field kernel (22) generates the desired external force when convolved with the object edge map, with the capability of attracting a contour to the region of interest.

$$\begin{aligned} \mathbf{K}(\mathbf{p}) &= -m(\mathbf{p}) \frac{\mathbf{p}}{\|\mathbf{p}\|} \\ m(\mathbf{p}) &= \exp(-||\mathbf{p}||^2/\gamma^2) \end{aligned} \quad (22)$$

$\mathbf{p} = \mathbf{0}$  denotes the kernel center. The capture range of the vector field is controlled by the parameter  $\gamma$ .

The set of candidate points  $\mathcal{M}_j$  for a parent  $c_j$  serves as the region of interest to which other components are likely to be attracted. Performing convolution of the candidate set with the kernel in (22) results in a vector field where the vectors are directed toward the parent, their magnitude attenuating gradually with distance from the candidate set. If  $E_j(\mathbf{x})$  is a binary edge-map which assumes a value of 1 only at points in  $\mathcal{M}_j$ , we can obtain the attraction force field  $\Gamma_j$  due to the parent  $m_j$  as

$$\Gamma_j(\mathbf{x}) = E_j(\mathbf{x}) * \mathbf{K}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \quad (23)$$

The nature of the attraction force field can be intuitively understood from Fig. 6. Fig. 6(a) shows three connected components and the representative parent  $c_1$  enclosed by its convex hull (shown in (b)). Fig. 6(c) illustrates the attraction force field due to the parent as the red arrows which are oriented in the direction of the parent component. The capture range, which is specified by  $\gamma$ , is shown by the red region.

Adopting this policy for designing the attraction field enjoys a few benefits. First, with a specified capture range, we can impose a locality in the approach, by discouraging distant segments to be connected to the parent. As  $\gamma$  increases, effect of the attraction force field gradually diminishes as one moves further from the parent. Moreover, the candidate set is chosen such that only the convex portions of the parent boundary are capable of generating the force field. This ensures not all local structures are potential candidates for linking. For example, in Fig. 6 the component  $c_3$  is not in the capture range of the force field of  $c_1$ , although it resides in the parent's local neighborhood. To summarize, the attraction force field is designed such that it may attract local connected components

which are present in near vicinity of the parent's boundary convexity.

3) *Attraction Force*: For a parent-child pair  $c_i$  and  $c_j$ , the parent attracts the child with a force  $\mathcal{F}_{attr}^{(i,j)}$  given by

$$\mathcal{F}_{attr}^{(i,j)}(\mathbf{y}) = \kappa_i \langle \Gamma_i(\mathbf{y}), -\mathbf{n}(\mathbf{y}) \rangle \theta_j(\mathbf{y}) \quad (24)$$

The indicator function  $\theta_j(\mathbf{y}) = 1$  if  $\mathbf{y} \in \delta c_j$  and 0 otherwise.  $\kappa_i$  is the normalized mass of the component  $c_i$  which is computed as the ratio of the number of pixels/voxels in  $c_i$  to the total pixels/voxels in  $\{c_1, \dots, c_p\}$ . The inner product term in (24) suggests that higher force of attraction is experienced by a point on a child's boundary if the outward normal at that point is oriented along the attraction field.

By introducing the factor  $\kappa_i$ , we equip heavier connected components with more attractive power. Assuming that the neurites occupy larger volume than the noisy background voxels, we clean the solution of the level set function by performing an area opening operation which eliminates small components with area less than a pre defined threshold [46]. This filtering operation avoids undesired objects to participate in the attraction force field computation. Now, for each parent-child pair in the filtered component space, we can compute the total attraction force  $\mathcal{F}_{attr}$  in (15) as

$$\mathcal{F}_{attr}(\mathbf{y}) = v_2 \sum_{i=1}^p \sum_{j \neq i}^p \mathcal{F}_{attr}^{(i,j)}(\mathbf{y}), \quad \forall \mathbf{y} \in \Omega. \quad (25)$$

The positive scalar  $v_2$  determines the effect of the attraction force on curve evolution. A finite difference scheme is used to solve the PDE in (15) with initial value obtained using Otsu's global segmentation [41] and Neumann boundary condition.

#### F. Handling of Discontinuities

Typically, one may encounter two major sources of structure discontinuity arising from initial segmentation. Fig. 7 shows three synthetic, disjoint components at an arbitrary stage of level set evolution. The type A discontinuity occurs when connectivity is absent between the end points or leaves of the centerline of the respective objects. Type A discontinuities dominate our application, and connectivity analysis of type A may be performed via Tree2Tree [16], by investigating the geometric orientation and Euclidean distance between the end points. However, end-point analysis algorithms like Tree2Tree are unable to process the type B discontinuities, where the link needs to be established between the terminal node of one component with a non-terminal point on the other object. This is where the proposed level set framework wins over conventional component linking algorithms since level sets are proficient in handling topological changes of the evolving segmentation.

1) *Type A Discontinuities*: Type A discontinuities are relatively simpler to analyze. If the neuron filament signal intensity is uniform, then the evolution force component of (15) sufficiently propagates the level sets until they are finally merged. However, when the signal drop is substantial, the attraction force term in (15) assists the parent and the

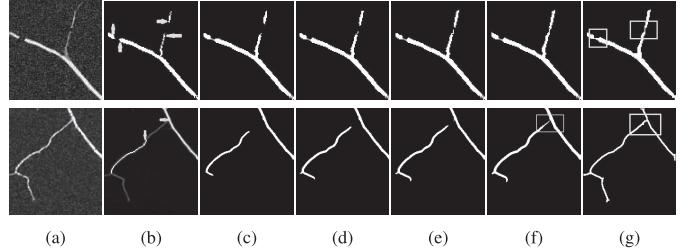


Fig. 8. (a) and (b) show the original image and the initial global segmentation respectively for two cases demonstrating handling of Type A (top row) and Type B (bottom row) discontinuities. (c)-(f) shows segmentation at subsequent time intervals. (g) shows the final segmentation, where the structure gaps have been closed (the merged portions are enclosed in rectangles).

child component to exert attractive forces on one another, thus propagating the curves till they merge. A demonstration is shown in the first row of Fig. 8. The initial segmentation using Otsu's method creates type A gaps, which are ultimately merged. We have intentionally eliminated a portion of the neuron's branch to demonstrate that our methodology works even in complete absence of signal.

2) *Type B Discontinuities*: Type B discontinuity involves two segments, for which connectivity needs to be established between one component's end point (or tip) with the other component's body. In presence of adequate signal intensity, TuFF drives the geometric contours toward the participating structure as per the filament orientation. However, when signal intensity drops, the attraction force takes over. An example is shown in the second row of Fig. 8(b), where the initial segmentation creates a type B gap. The situation is different from that of type A, where both the components may attract each other. In case of type B, only one component can assume a parent's role. Note that this is the extreme scenario, where the underlying signal strength is so feeble that it renders the evolution force term useless. However, assuming that the parent's mass is not negligible, this attraction force is strong enough to pull the local child connected component for potential merging. It should be noted that only those regions on the child's boundary whose outward normals are maximally aligned with the exerted force field are attracted toward the parent.

#### G. Neuron Tracing via Centerline Extraction

Numerical implementation of (15) allows iterative computation of the level set function, which can be expressed as

$$\phi^{(k+1)} = \phi^{(k)} + \Delta t \mathcal{L}^{(k)} \quad (26)$$

The learning rate  $\Delta t$  is fixed to a small value ( $\approx 0.1$ ) to allow stable computation.  $\mathcal{L}^{(k)}$  denotes the discretized version of the right hand side of (15).  $\phi^{(k)}$  is the level set function at iteration  $k$ . To initialize the active contour, we require the initialized curve to be inside the neurite structure. The initial level set function may be easily obtained via few mouse clicks to select a region inside the neuron structure. However, to avoid this human involvement, we perform a global thresholding of the scale space vesselness image (5)

using Otsu's technique [41], followed by noisy binary segment removal using the area open filter [46]. The iterative procedure is halted when no significant change in the length of the zero level curve of  $\phi$  is observed. At convergence, the neuron structure is extracted by selecting the largest binary component in the solution. A cubic spline is then fitted to each branch of the obtained centerline to obtain smooth tracing of neuron centerline.

### H. Summary of TuFF

Before proceeding to experimental results, we provide a summary of the TuFF algorithm and highlight its salient features. First, we avoid human intervention in terms of seed point selection. Automated initialization of the level set is performed by Otsu's global thresholding [41] followed by noise removal using morphological area open operators [46]. The level set function is computed from this initialized segments using binary distance transform.

Second, TuFF presents a natural framework to process both type A and type B discontinuities (Fig. 8). This is a major improvement over the tracer Tree2Tree [16], where the inability to handle type B discontinuity introduces several false connections in the solution.

Finally, TuFF is capable of joining broken neurite fragments even in complete absence of signal. The proposed attraction force field is independent of the local signal intensity and depends only on the morphology and relative positioning of the connected components. This feature improves on the widely used local intensity seeking neuron tracers [13], which are susceptible to illumination variation in the images of neural structure. The TuFF guided evolution energy is combined with the attraction force component in a mathematically elegant, integrated fashion as opposed to a multistage sequential processing pipeline.

## III. EXPERIMENTAL RESULTS

In this section we demonstrate the efficacy of our method by experimental analysis of both 2D and 3D confocal images. We further compare our segmentation accuracy to that of three widely used neuron tracers.

### A. Dataset for Segmentation

We test the performance of TuFF segmentation algorithm on sets of 2D and 3D confocal microscopy images. The 2D images are primarily used to demonstrate the efficacy of TuFF over component analysis algorithms like Tree2Tree [16]. The 3D image data set consists of 24 confocal microscopy images of the Drosophila larva, which are labeled by green fluorescence protein (GFP). Out of these 24 images, 16 images are captured in the Condron Lab of the University of Virginia. The images are captured using a laser scanning confocal microscope and has a horizontal pixel width of  $0.14\mu m$  and vertical pixel width of  $0.18\mu m$ . These images are characterized by intense background clutter from non neuronal objects (such as the food particles, mildly fluorescing tissues etc.) and considerable contrast and intensity variation.

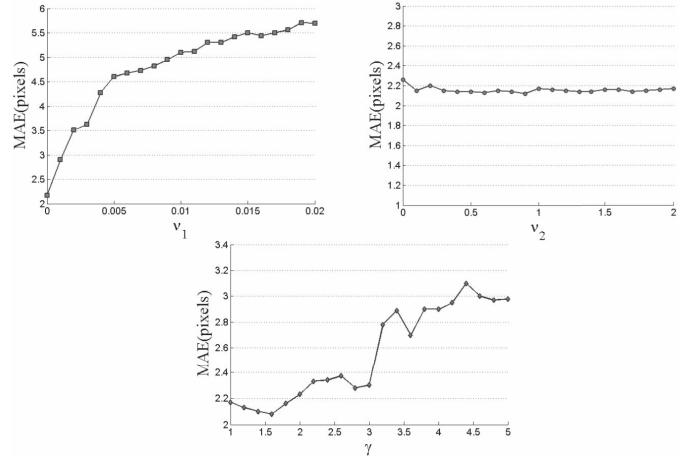


Fig. 9. Sensitivity analysis of the parameters. The mean absolute error of the traced centerline are plotted in the vertical axis for different values of the tuning parameters.

The second data set for 3D analysis consists of 8 olfactory (axonal) projection (OP) image stacks of Drosophila larva. These images were used in the Diadem challenge [47] and like the previous dataset, these neurons are also imaged by a confocal microscope. These OP-data set images are less noisy and the contrast is better than the images in Condron data set. However, the neurons in this data set exhibit acutely complicated structural appearance in addition to occasional intensity heterogeneity along the neurite filaments.

### B. Parameter Selection

The level set evolution equation (15) depends on a few parameters. The evolution force  $\mathcal{F}_{evolve}$  requires specifying the positive scalars  $a_0$  and  $a_1$  in (12) which controls the anisotropy of curve evolution. As we have discussed before, since the neurite thickness in our case does not vary considerably, we have adopted the isotropic case, as it requires lesser computation. Therefore, we choose  $a_0 = 1$  and a very high value for  $a_1$ .

The smoothness of the evolved curve is controlled by the parameter  $v_1$  in (17). Effect of gradually increasing  $v_1$ , keeping other parameters fixed results in an increased mean absolute error in tracing, as shown in Fig. 9. For our experiments,  $v_1$  is fixed at a small value in the range  $0 - 0.02$ .

The attraction force defined in (25) depends on the weighing parameter  $v_2$  and the parameter  $\gamma$  controlling the local capture range. As we observe in Fig. 9 our algorithm is relatively robust to the choice of  $v_2$ . However, we notice that a very low value of  $v_2$  restricts the attraction force from closing small gaps. For all our experiments, we select  $v_2 = 1$ . The term  $\gamma$  induces locality in the capture range for the attraction force. While a small value of  $\gamma$  can be too restrictive, a relatively high value attracts distant structures to be merged to the attracting component (see Fig. 9). Note that we are interested in connecting the disjoint structures over a local neighborhood. Based on our collaborator's knowledge about the dataset, we observe that typically  $\gamma$  ranges between  $0.2 - 1.5\mu m$  ( $\approx 1-7$  pixels) for our data. Setting these biologically inspired bounds on the range of  $\gamma$ , we proceed to select the value

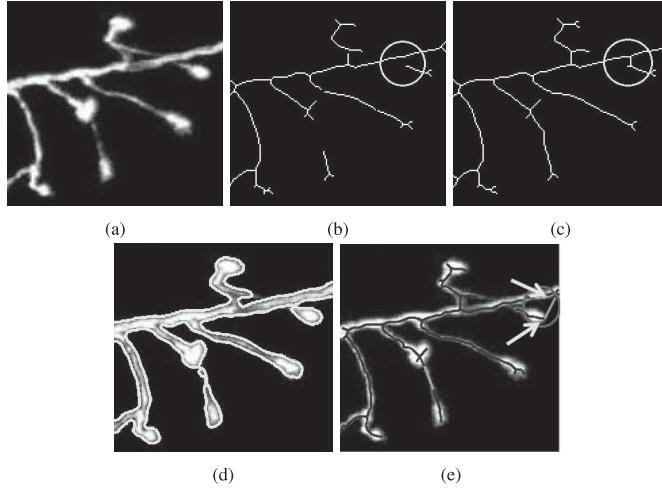


Fig. 10. (a) 2D neuron sub-image. (b) Centerline of the initial segmentation using [41]. The type B discontinuity is highlighted by the yellow circle. (c) Centerline obtained after segmentation using TuFF. (d) Final segmentation via TuFF. (e) Tracing using Tree2Tree. A typical error in connectivity is indicated by the arrows.

in the following manner. First, at any stage of segmentation, we compute the median distance  $\rho$  between all the segments, and update the value of  $\gamma$  as  $\gamma^* = \rho/3$ . If the updated value is beyond the pre-selected upper or lower bounds, we select the closest boundary value for  $\gamma^*$ . This is repeated at each iteration to compute the attraction force.

Experimentally we have observed that the parameters  $\Delta$  and  $\epsilon$  can be prefixed to a particular value without affecting performance. For all experiments we choose  $\Delta = 5$  pixels and  $\epsilon = 1$  as suggested by the authors in [36].

### C. Efficacious Handling of Branch Connectivity

Previously, we have demonstrated the ability of TuFF to handle type A and type B discontinuities. In this section, we demonstrate the advantage of using TuFF over Tree2Tree [16] for determining branch connectivity. For this purpose, we show segmentation results on a few 2D neuron images. The 2D images are obtained from a maximum intensity projection of the corresponding 3D stacks. We also perform experiments on a few synthetically grown neurons, where the 2D imaging is performed by measuring the fluorescence from the fluorophores used to stain these neurons.

To set up Tree2Tree for segmentation, we follow the author's methodology of performing an initial segmentation to obtain a set of binary components. The component analysis stage of Tree2Tree then decides on the connection between the segments by analyzing their relative orientation. To initialize the level set for TuFF, we have used Otsu's segmentation, same as Tree2Tree, and the level set propagates according to (15). Fig. 10 demonstrates an example where Tree2Tree creates improper connection, due to its inability to handle type B discontinuity. The level set based methodology in TuFF performs proper segmentation (shown in Fig. 10(c), (d)). It is evident that the type B gap is closed by TuFF, where Tree2Tree fails to do so (see Fig. 10(c) vs (e)).

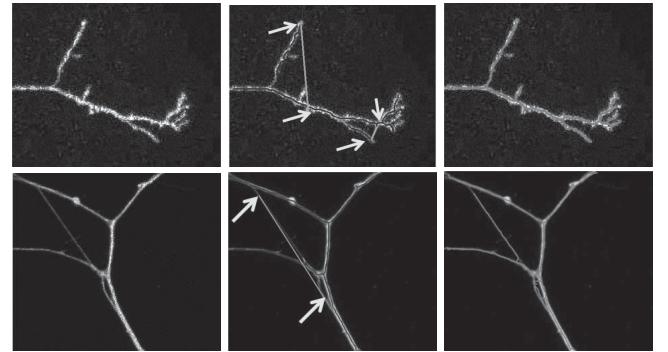


Fig. 11. The first column shows sample 2D neuron images. Tree2Tree [16] segmentation results are displayed in the second column. The edges linked by Tree2Tree are shown in green and the traced centerline is overlaid on the original image in blue. Excessive clutter restricts the efficiency of Tree2Tree, yielding improper connections, which are highlighted by the yellow arrows. The last column shows tracing output for TuFF algorithm, with the traced medial axis plotted in magenta.

Two more examples are shown in Fig. 11 where Tree2Tree's tracing (shown in blue) creates incorrect branch connection as compared to TuFF (shown in magenta). The connection errors are highlighted by the yellow arrows. Tree2Tree segmentation results suggest lack of robustness of the component linking scheme for complex structures embedded in a noisy environment. Furthermore the initial segmentation step in Tree2Tree often fails to detect low contrast objects, which cannot be recovered in future, since the multistage pipeline of Tree2Tree is unable to recover lost neurite portions.

The above examples suggest that TuFF handles bifurcations and component gaps successfully, since level sets are well equipped in handling topological changes. Also, the specially designed attraction force component of TuFF makes segmentation robust in cases where structure gaps result from very weak signal intensity (Fig. 10).

### D. Comparison of Segmentation Performance

In this section we present a comparative segmentation performance analysis of the proposed method TuFF versus three popularly used neuron tracers. The ground truth data for segmentation is obtained by manually selecting points on the neuron structure and joining them manually in a manner that the morphological structure is preserved. The Vaa3d software [48] is used for creating the ground truth. To evaluate the performance of TuFF, we compare its performance to the following algorithms.

1) *Graph Augmented Deformable (GD) Model [9]*: This semi automatic tool is extensively used for its relatively simple working methodology, which consists of a manual seed selection step followed by automated seed joining process by using graph theoretic techniques. Since the algorithm's efficacy is inversely proportional to the spatial distribution of selected seed points, we only select the neuron terminal points as the set of seeds. As the seed selection is performed manually, a practice which TuFF avoids, we believe that selecting the minimal set of seeds is essential to maintain fairness of comparison. Sample tracing results using this algorithm are shown in yellow.

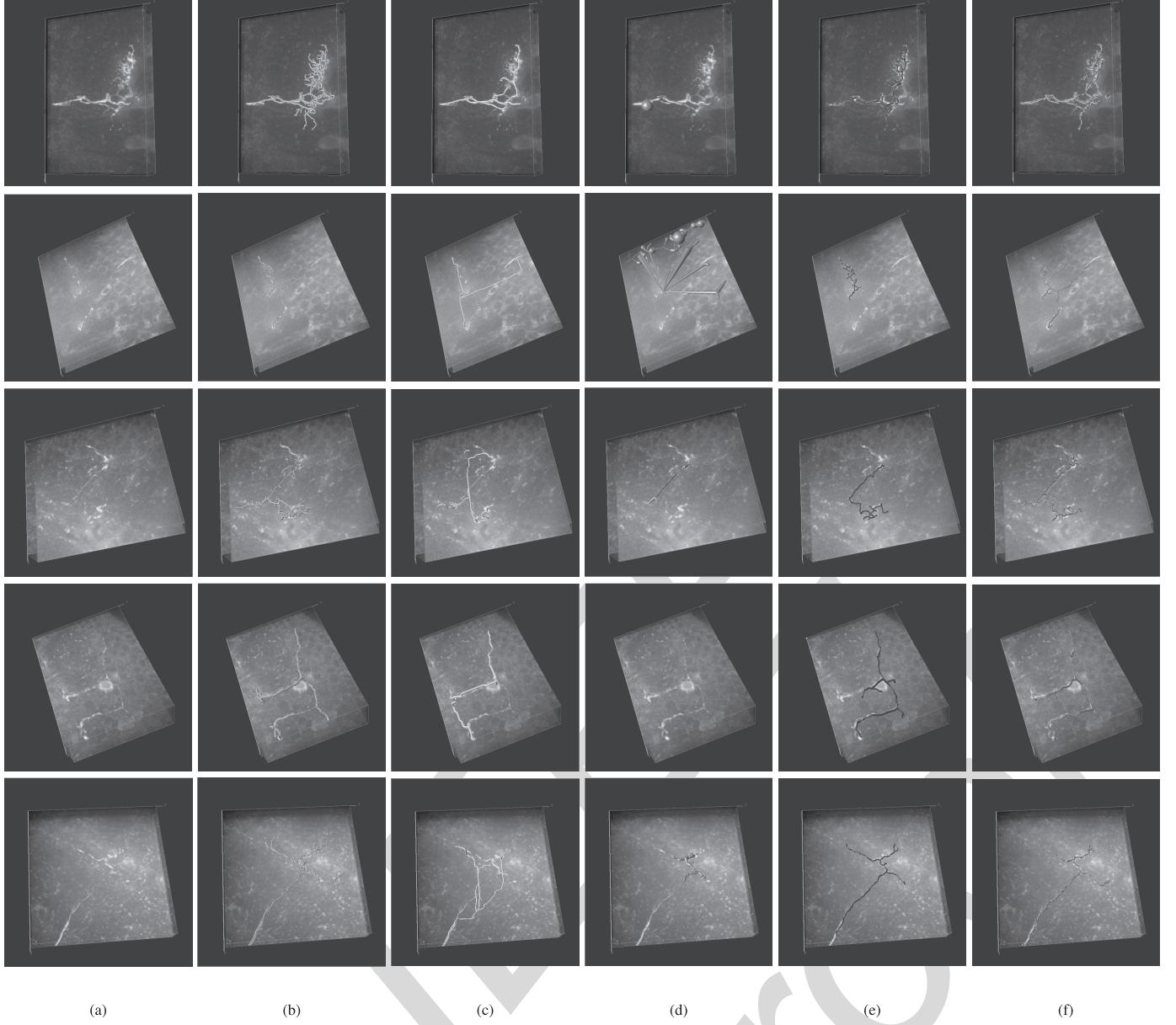


Fig. 12. Tracing results on 3D images of the UVA-Condron dataset. First column shows the original images, followed by the tracing outputs of the different algorithms. Tracing results of TuFF are shown in the last column in magenta. (a) 3D stack. (b) Ground truth. (c) GD model [9]. (d) NeuronStudio [13]. (e) Tree2Tree [16]. (f) TuFF.

2) *Neuronstudio* [13]: Neuronstudio is one of the state of the art publicly available automatic neuron segmentation software which is heavily used by biologists for tracing purpose. We have seen that segmentation accuracy of NeuronStudio is affected by the choice of the initial seed point. For each image in our dataset, we experiment with several initial seed locations and finally choose the one which yields the best visual segmentation result. Neuronstudio segmentation results are shown in orange color.

3) *Tree2Tree* [16]: As discussed earlier, Tree2Tree belongs to the category of seed independent neron segmentation methods. Setting up Tree2Tree requires an initial segmentation stage, followed by graph-theoretic component linking procedure. The segmentation results of Tree2Tree are shown in blue color.

For each of the above mentioned algorithms and TuFF, we first obtain the segmentation followed by neuron centerline detection. A cubic spline is fitted to each branch of the detected centerline. This spline fitted centerline of the neurons represent the tracing results.

#### E. Visual Assesment of Segmentation Results

1) *Results on Condron Data Set*: Fig. 12 shows the performance of the above mentioned neuron tracers on five representative neurons chosen from the Condron dataset. The 3D stacks are shown in the first column, followed by manual ground truth segmentation in the second column (shown in green). Tracing results using GD model [9] is plotted in yellow in the third column. The fourth and fifth columns



Fig. 13. Results on the images of the OP dataset. First column shows the original images, followed by the tracing outputs of the different algorithms. Tracing results of TuFF are shown in the last column in magenta. (a) 3D stack. (b) Ground truth. (c) GD model [9]. (d) NeuronStudio [13]. (e) Tree2Tree [16]. (f) TuFF.

show segmentation output using the automated techniques Neuronstudio and Tree2Tree (plotted in orange and blue color) respectively. Finally, the last column shows the neuron tracing due to TuFF (plotted in magenta).

It may be observed that these images are in general noisy, which makes the segmentation task difficult. Moreover, high structural complexity of the neurons require sophisticated mechanism to preserve the structural morphology. The severity of contrast variation and low SNR pose difficulty for the GD model. Even with manually selected terminal nodes, it is seen that the semi-manual tracer performs incorrect segmentation (Fig. 12, second column, rows 2–5). This is primarily due to the inability of the local search based technique fails to identify the actual filamentous path in presence of clutter. Furthermore, human assisted neurite termination detection proved to be a difficult and time consuming problem in these images owing to the high structural complexity.

Neuronstudio performs particularly poorly in these examples. The major reason can be attributed to the lack of continuity in the neurite structure and high signal variation, which forces the algorithm to converge prematurely. Also, the cluttered environment is detrimental to the performance of the local voxel scooping process of Neuronstudio. This results

in under segmentation and sometimes, incorrect segmentation due to leakage of the region growing technique.

Tree2Tree outperforms Neuronstudio, especially when the component linking algorithm is able to determine proper connectivity. We observe that Tree2Tree performs well if the initial segmentation step is reliable. However, under segmentation is an inherent problem in Tree2Tree due its inability to incorporate additional neuronal structures in its solution after initial thresholding.

On the other hand, TuFF performs segmentation efficiently, even in cluttered environment. A close inspection would reveal that important morphological entities like bifurcation points and branch locations are preserved (see Fig. 12 rows 2, 3 and 4), while the iterative directional region growing scheme prevents under segmentation of neurons.

*2) Segmentation Results on OP Dataset:* These image stacks exhibit relatively higher signal intensity than the Condrone data set. However, neuron tracing is still a challenging task owing to their complicated structure and sudden intensity variations in the neurites, creating a fragmented, discontinuous appearance. This often results in type B discontinuity which demands sophisticated analysis. Fig. 13 compares the segmentation results for above mentioned algorithms.

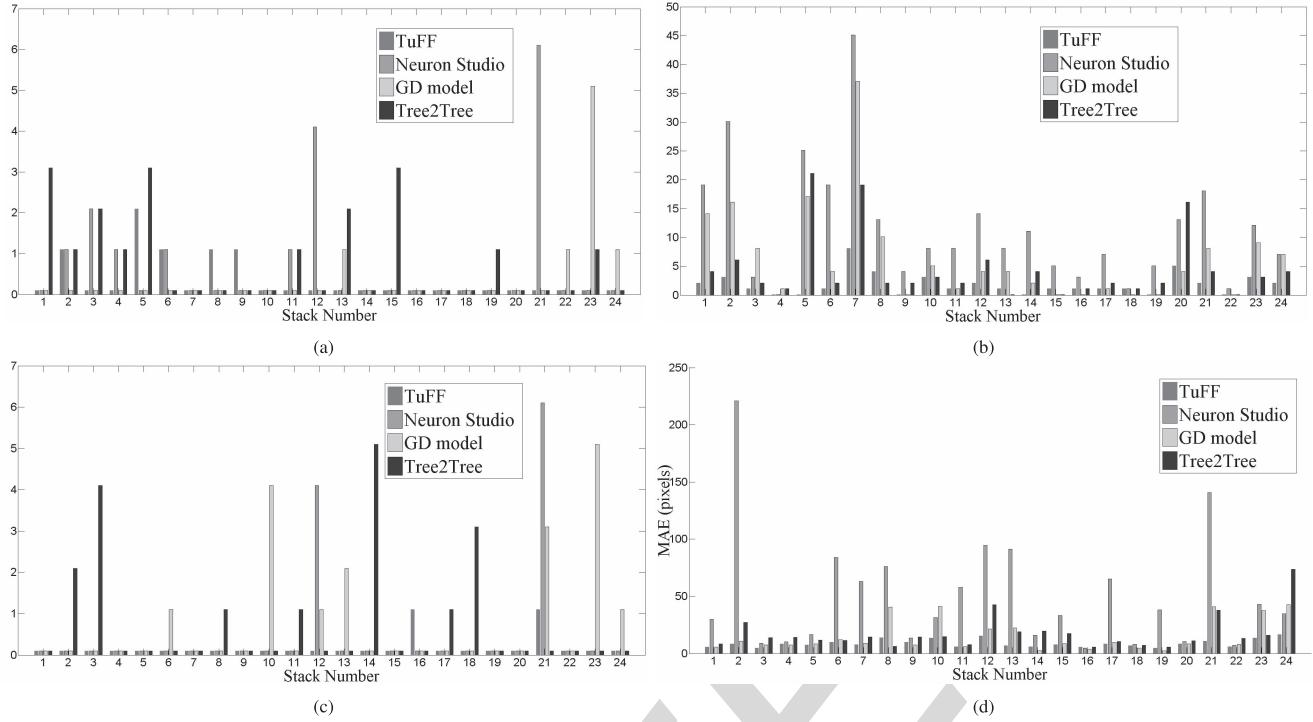


Fig. 14. (a)-(c): Quantitative performance of the four neuron tracers TuFF (pink), Neuron Studio (orange), GD model [9] (yellow) and Tree2Tree (blue) in terms of number of over-estimated branches, number of under-estimated branches and total number of wrong connections respectively. (d) quantifies the tracing accuracy in terms of mean absolute error defined in (27). (a) False positives. (b) False negatives. (c) Incorrect connection. (d) MAE.

Reduction in background clutter and increased signal intensity assists the semi automatic GD-model tracer. Since the images exhibit significant improvement in contrast, manual detection of seeds is less stressful. Still, the complicated structure of a few images (Fig. 13, row 1 for example) makes manual seed selection demanding. Performance of Neuronstudio also shows slight improvement in this dataset. However, despite brighter foreground and less noise, this local tracing scheme shows tendency to stop at intensity gaps, which needs to be modified manually at a later stage. On the other hand, it is observed that Tree2Tree's performance degrades significantly for this dataset. This is primarily due to a large number of improper branch connections. This connectivity error occurs mostly due to Tree2Tree's inability to handle type B discontinuities (Fig. 13, rows 1-3). In fact, even in relatively high SNR images Tree2Tree under performs significantly by extracting an improper structural morphology of the neurons. TuFF, however demonstrates good performance on these images by virtue of its ability to handle structure gaps automatically. The segmentation results are shown in the last column of Fig. 13. A qualitative assessment of the algorithm's performance is presented in the following sections.

#### F. Quantitative Performance Analysis

To quantify the segmentation performance, we identify four measures which reflects the efficiency of a particular neuron tracer. These are as follows: number of over-estimated branches (Fig. 14(a)), number of unidentified/missed branches (Fig. 14(b)), total number of incorrect branch connections (see Fig. 14(c)) and finally the mean absolute error in the

traced centerline with respect to the ground truth. The number of over determined/missed branches reflect the adequacy of an algorithm in respecting the morphology of the imaged neuronal structure. This quantification of the segmentation quality is performed by a human expert. However, since even the ground truth data is susceptible to subtle errors in computing the 3D skeleton, we have disregarded small branches (less than 5 units in length) from the analysis. The graphs in Fig. 14(a) and (b) suggests that over the whole data set, TuFF outperforms the competing algorithms in a majority of cases. It is observed in a few cases that Neuronstudio in particular misses a large number of branches, due to its inability to deal with fragmented structure.

The number of incorrect branch connections (Fig. 14(c)) indicate an algorithm's ability to tackle discontinuities. Indeed, improper connections often result when signal heterogeneity is significant. Apart from a few occasions, TuFF demonstrates its superiority in handling discontinuities better than other automated methods. To perform quantitative analysis of the traced neuron centerline, we compute the mean absolute error (MAE) of the obtained trace against the manually acquired ground truth. If  $\mathcal{P} = \{p_1, \dots, p_n\}$  and  $\mathcal{Q} = \{q_1, \dots, q_m\}$  denote the set of traced coordinates for a neuron, the mean absolute error (in pixels) between the traces is given by

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n \min_j |p_i - q_j| + \frac{1}{m} \sum_{i=1}^m \min_k |q_i - p_k| \quad (27)$$

$\forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, n\}$ . Mean absolute errors for the 24 3D images are plotted for each algorithm in Fig. 14(d). It is observed that TuFF outperforms the automated

TABLE I  
COMPARISON OF MAE

	TuFF	Neuron Studio	GD model	Tree2Tree
Avg. MAE	8.81	79.98	15.41	17.62
Median MAE	7.95	34.06	8.54	13.98
Std. Dev	3.4	50.6	14.03	15.08

tracers Tree2Tree and Neuronstudio in almost all of the 24 cases, except for the 8<sup>th</sup> and 16<sup>th</sup> stack, where Tree2Tree and Neuronstudio perform marginally better. Also, TuFF successfully competes with the semi-automatic GD-model, even outperforming it in some images in the Condron dataset.

The mean, median and standard deviation MAE of the four algorithms are reported in Table I. This suggests that on a whole TuFF outperforms its competitors with a mean and median MAE of 8.81 (pixels) and 7.95 (pixels) respectively. TuFF also exhibits 75% improvement of mean error over the second best performer, which is the semi-automatic tracer of Peng *et al.* If we compare its efficacy against the fully automated techniques, we obtain an improvement of over 98% over Tree2Tree, while Neuron Studio is outperformed with an improvement of greater than 400%. Also, the error standard deviation of TuFF is only 3.4 as compared to 50.6, 14.03 and 15.08 for Neuronstudio, GD-model and Tree2Tree. The visual segmentation results and the quantitative results presented here suggests the efficiency of TuFF in segmenting structurally complex neurons from cluttered confocal microscope images.

#### G. Note on Computational Efficiency

From a computational perspective, TuFF has the disadvantage that the segmentation is performed iteratively. Similar to all numerical PDE based methods, the speed of convergence can be controlled by setting a higher value for the learning rate, albeit at the cost of sacrificing accuracy. However, we should mention that in our implementation we have not concentrated on making the algorithm run faster. In fact, recent research suggest that significant decreases in computational cost can be achieved by using more intelligent numerical algorithms to solve the evolution equation. However, TuFF does hold an advantage over popular semi automatic tracers in the sense that no manual intervention is required. For example, to set up the GD model for segmentation, a human subject was assigned to visually determine around 20-30 end points to be selected for each 3D stack for seed initialization. With the current unoptimized implementation, TuFF takes approximately 300 seconds on average to segment a neuron from a 200 × 200 × 60 dimension 3D stack using Matlab for implementation on a 3.4 GHz Intel i7 processor with 8Gb RAM.

## IV. CONCLUSION

In this paper we have presented an automated neuron segmentation algorithm which can segment neurons from both 2D and 3D images. The proposed framework is suitable for tracing highly fragmented neurite images, and is capable of processing the structure discontinuities automatically,

while respecting the overall neuron morphology. Connectivity analysis is performed in a level set framework which presents a nice and simple alternative to graph based techniques which may introduce undesired branches in segmentation. The efficiency of TuFF is further demonstrated by its superior overall quantitative performance where it outperforms peer algorithms, including a semi manual tracer.

## APPENDIX

We provide the derivation of (17) for 2D, ie.  $\mathbf{x} = (x, y)^T$ . The TuFF vector fields are given by  $\mathbf{v}_1 = (v_{11}, v_{12})^T$  and  $\mathbf{v}_2 = (v_{21}, v_{22})^T$ ; the dependency on  $\mathbf{x}$  implied. The extension to 3D is simple and follows from this derivation. We can rewrite  $\mathcal{E}_{reg}(\phi) = \int E_1(\phi)d\mathbf{x}$ , where  $E_1(\phi) = v_1|\nabla\phi|(\mathbf{x})\delta_\epsilon(\phi)$ . Then by calculus of variation, the Gateaux variation of  $\mathcal{E}_{reg}$  can be obtained as:

$$\frac{\delta\mathcal{E}_{reg}}{\delta\phi} = \frac{\partial E_1}{\partial\phi} - \frac{\partial}{\partial x}\left(\frac{\partial E_1}{\partial\phi_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial E_1}{\partial\phi_y}\right) \quad (28)$$

Since the proof is already shown in [36], we merely state the result as follows:

$$\frac{\delta\mathcal{E}_{reg}}{\delta\phi} = -v_1 \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \delta_\epsilon(\phi) \quad (29)$$

Similarly, we can write the evolution energy as  $\mathcal{E}_{evolve}(\phi) = \int E_2(\phi)d\mathbf{x}$ . This can be expanded as  $E_2(\phi) = A_1(\phi) + A_2(\phi)$ , where  $A_j(\phi) = \alpha_j \langle \mathbf{v}_j, \frac{\nabla\phi}{|\nabla\phi|} \rangle^2 H_\epsilon(\phi)$ . The dependency of  $\alpha, \phi$  and  $\mathbf{v}_j$  on  $\mathbf{x}$  is implied, and hence not mentioned explicitly.

We can further decompose  $A_1$  as

$$A_1(\phi) = -\alpha_1 \frac{(v_{11}\phi_x + v_{12}\phi_y)^2}{\phi_x^2 + \phi_y^2} H_\epsilon(\phi)$$

Let us denote  $\beta_j = \langle \mathbf{v}_j, \mathbf{n} \rangle$ , where the unit normal vector  $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ . Therefore, we can write  $A_1(\phi) = -\alpha_1 \beta_1^2 H_\epsilon(\phi)$ .

As earlier, we compute the Gateaux derivative as follows:

$$\frac{\partial A_1}{\partial\phi} = -\alpha_1 \beta_1^2 \delta_\epsilon(\phi) \quad (30)$$

Also, by simple algebraic manipulation, we obtain

$$\begin{aligned} \frac{\partial A_1}{\partial\phi_x} &= -2 \left[ \frac{\alpha_1 \beta_1}{|\nabla\phi|} v_{11} - \alpha_1 \left( \frac{\beta_1}{|\nabla\phi|} \right)^2 \phi_x \right] H_\epsilon(\phi) \\ \frac{\partial A_1}{\partial\phi_y} &= -2 \left[ \frac{\alpha_1 \beta_1}{|\nabla\phi|} v_{12} - \alpha_1 \left( \frac{\beta_1}{|\nabla\phi|} \right)^2 \phi_y \right] H_\epsilon(\phi) \end{aligned}$$

Therefore, we have

$$\frac{\partial}{\partial x} \left( \frac{\partial A_1}{\partial\phi_x} \right) = -2 \left[ \frac{\partial}{\partial x} (\eta_1 v_{11}) - \frac{\partial}{\partial x} \left( \eta_1 \beta_1 \frac{\phi_x}{|\nabla\phi|} \right) \right] \quad (31)$$

$$\frac{\partial}{\partial y} \left( \frac{\partial A_1}{\partial\phi_y} \right) = -2 \left[ \frac{\partial}{\partial y} (\eta_1 v_{12}) - \frac{\partial}{\partial y} \left( \eta_1 \beta_1 \frac{\phi_y}{|\nabla\phi|} \right) \right] \quad (32)$$

Where  $\eta_j = \frac{\alpha_j \beta_j}{|\nabla\phi|} H_\epsilon(\phi)$ . Therefore, by symmetry we compute

$$\frac{\partial}{\partial x} \left( \frac{\partial A_1}{\partial\phi_x} \right) + \frac{\partial}{\partial y} \left( \frac{\partial A_1}{\partial\phi_y} \right) = -2 \operatorname{div}[(\eta_j)(\mathbf{v}_j - \beta_j \mathbf{n})] \quad (33)$$

The Gateaux variation of  $\mathcal{E}_{\text{evolve}}$  can be obtained as:

$$\frac{\delta \mathcal{E}_{\text{evolve}}}{\delta \phi} = \frac{\partial E_2}{\partial \phi} - \frac{\partial}{\partial x} \left( \frac{\partial E_2}{\partial \phi_x} \right) - \frac{\partial}{\partial y} \left( \frac{\partial E_2}{\partial \phi_y} \right) \quad (34)$$

We now use gradient descent to find the local minima of the functionals. The regularizer force and evolution forces are given by  $\mathcal{F}_{\text{reg}} = -\frac{\delta \mathcal{E}_{\text{reg}}}{\delta \phi}$  and  $\mathcal{F}_{\text{evolve}} = -\frac{\delta \mathcal{E}_{\text{evolve}}}{\delta \phi}$  which leads to the following equations:

$$\mathcal{F}_{\text{reg}} = v_1 \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \quad (35)$$

and

$$\mathcal{F}_{\text{evolve}} = \sum_{j=1}^d \left( \alpha_j \beta_j^2 \delta_\epsilon(\phi) - 2 \operatorname{div} [\eta_j (\mathbf{v}_j - \beta_j \mathbf{n})] \right) \quad (36)$$

## REFERENCES

- [1] C. Koch and I. Segev, "The role of single neurons in information processing," in *Nature Neuroscience*, vol. 3. London, U.K.: Nature Pub. Group, 2000, pp. 1171–1177.
- [2] H. Cuntz, F. Forstner, J. Haag, and A. Borst, "The morphological identity of insect dendrites," *PLoS Comput. Biol.*, vol. 4, no. 12, p. e1000251, 2008.
- [3] J. Chen and B. G. Condron, "Branch architecture of the fly larval abdominal serotonergic neurons," *Develop. Biol.*, vol. 320, no. 1, pp. 30–38, 2008.
- [4] E. A. Daubert, D. S. Heffron, J. W. Mandell, and B. G. Condron, "Serotonergic dystrophy induced by excess serotonin," *Molecular Cellular Neurosci.*, vol. 44, no. 3, pp. 297–306, 2010.
- [5] H. Cuntz, M. W. H. Remme, and B. Torben-Nielsen, *The Computing Dendrite*, vol. 10. New York, NY, USA: Springer-Verlag, 2014, p. 12.
- [6] G. A. Ascoli, D. E. Donohue, and M. Halavi, "Neuromorpho.org: A central resource for neuronal morphologies," *J. Neurosci.*, vol. 27, no. 35, pp. 9247–9251, 2007.
- [7] K. A. Al-Kofahi *et al.*, "Median-based robust algorithms for tracing neurons from noisy confocal microscope images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 302–317, Dec. 2003.
- [8] J. Xie, T. Zhao, T. Lee, E. Myers, and H. Peng, "Anisotropic path searching for automatic neuron reconstruction," *Med. Image Anal.*, vol. 15, no. 5, pp. 680–689, 2011.
- [9] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson, "Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model," *Bioinformatics*, vol. 26, no. 12, pp. i38–i46, 2010.
- [10] H. Peng, F. Long, and G. Myers, "Automatic 3D neuron tracing using all-path pruning," *Bioinformatics*, vol. 27, no. 13, pp. i239–i247, 2011.
- [11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.
- [12] G. González, E. Türetken, F. Fleuret, and P. Fua, "Delineating trees in noisy 2D images and 3D image-stacks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2799–2806.
- [13] A. Rodriguez, D. B. Ehlenberger, P. R. Hof, and S. L. Wearne, "Three-dimensional neuron tracing by voxel scooping," *J. Neurosci. Methods*, vol. 184, no. 1, pp. 169–175, 2009.
- [14] S. L. Wearne, A. Rodriguez, D. B. Ehlenberger, A. B. Rocher, S. C. Henderson, and P. R. Hof, "New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales," *Neuroscience*, vol. 136, no. 3, pp. 661–680, 2005.
- [15] S. Mukherjee and S. T. Acton, "Vector field convolution medialness applied to neuron tracing," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 665–669.
- [16] S. Basu, B. Condron, A. Aksel, and S. T. Acton, "Segmentation and tracing of single neurons from 3D confocal microscope images," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 2, pp. 319–335, Mar. 2013.
- [17] S. Mukherjee, S. Basu, B. Condron, and S. T. Acton, "Tree2Tree2: Neuron tracing in 3D," in *Proc. 10th IEEE Int. Symp. Biomed. Imag.*, Apr. 2013, pp. 448–451.
- [18] S. Basu, A. Aksel, B. Condron, and S. T. Acton, "Tree2Tree: Neuron segmentation for generation of neuronal morphology," in *Proc. IEEE Int. Symp. Biomed. Imag.*, Apr. 2010, pp. 548–551.
- [19] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.
- [20] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam, "A broadly applicable 3D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 193–217, 2011.
- [21] H. Cai, X. Xu, J. Lu, J. Lichtman, S. P. Yung, and S. T. C. Wong, "Shape-constrained repulsive snake method to segment and track neurons in 3D microscopy images," in *Proc. 3rd IEEE Int. Symp. Biomed. Imag.*, Apr. 2006, pp. 538–541.
- [22] A. Narayanaswamy, Y. Wang, and B. Roysam, "3D image pre-processing algorithms for improved automated tracing of neuronal arbors," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 219–231, 2011.
- [23] A. Santamaría-Pang, C. M. Colbert, P. Saggau, and I. A. Kakadiaris, "Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2007, pp. 486–494.
- [24] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, no. 1, pp. 179–195, 1996.
- [25] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea, "A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes," *Med. Image Anal.*, vol. 13, no. 6, pp. 819–845, 2009.
- [26] L. M. Lorigo *et al.*, "CURVES: Curve evolution for vessel segmentation," *Med. Image Anal.*, vol. 5, no. 3, pp. 195–206, 2001.
- [27] A. Gooya, H. Liao, K. Matsumiya, K. Masamune, Y. Masutani, and T. Dohi, "A variational method for geometric regularization of vascular segmentation in medical images," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1295–1312, Aug. 2008.
- [28] A. Gooya, H. Liao, and I. Sakuma, "Generalization of geometrical flux maximizing flow on Riemannian manifolds for improved volumetric blood vessel segmentation," *Comput. Med. Imag. Graph.*, vol. 36, no. 6, pp. 474–483, 2012.
- [29] A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1565–1578, Dec. 2002.
- [30] Y. Shang *et al.*, "Vascular active contour for vessel tree segmentation," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 4, pp. 1023–1032, Apr. 2011.
- [31] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.
- [32] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2096–2106, Aug. 2007.
- [33] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995.
- [34] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.
- [35] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
- [36] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
- [37] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 1164–1170.
- [38] A. Yezzi, Jr., A. Tsai, and A. Willsky, "Binary and ternary flows for image segmentation," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Oct. 1999, pp. 1–5.
- [39] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1940–1949, Oct. 2008.
- [40] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 195–215, 2007.
- [41] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, nos. 285–296, pp. 23–27, 1975.
- [42] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, 1998, pp. 130–137.
- [43] X.-F. Wang, D.-S. Huang, and H. Xu, "An efficient local Chan-Vese model for image segmentation," *Pattern Recognit.*, vol. 43, no. 3, pp. 603–618, 2010.
- [44] J. L. Troutman, *Variational Calculus and Optimal Control: Optimization With Elementary Convexity*. New York, NY, USA: Springer-Verlag, 1995.

- [45] R. L. Graham and F. F. Yao, "Finding the convex hull of a simple polygon," *J. Algorithms*, vol. 4, no. 4, pp. 324–331, 1983.
- [46] S. T. Acton, "Fast algorithms for area morphology," *Digital Signal Process.*, vol. 11, no. 3, pp. 187–203, 2001.
- [47] K. M. Brown *et al.*, "The DIADEM data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 143–157, 2011.
- [48] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnol.*, vol. 28, no. 4, pp. 348–353, 2010.



**Suvadip Mukherjee** (S'11) is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, USA, where he is involved in image analysis methods under the supervision of Dr. S. T. Acton.  
He received the bachelor's degree in electrical engineering from Jadavpur University, Kolkata, India, in 2008, and the master's degree in computer science from the Indian Statistical Institute, Kolkata, in 2011. His research interests include image segmentation and analysis techniques applied to biological and biomedical problems. He is also interested in other applied image and video processing research, such as image classification and feature identification techniques for CBIR and object tracking in videos.



**Barry Condron** is currently a Professor with the Department of Biology, University of Virginia (UVA), Charlottesville, VA, USA. He received the B.S. degree in mathematics and biochemistry from University College Cork, Cork, Ireland, in 1985, and the Ph.D. degree in genetics from the University of Utah, Salt Lake City, UT, USA, in 1991, where he was involved in gene regulation in a virus.

He held a post-doctoral position with the Dr. Kai Zinn's Laboratory, California Institute of Technology, Pasadena, CA, USA, where he was involved in how circuits for during development. He was involved in this field first at the School of Medicine, UVA, from 1997 to 2000, and has been at the Department of Biology, since 2000.



**Scott T. Acton** (F'13) is currently a Professor of Electrical and Computer Engineering and of Biomedical Engineering with the University of Virginia (UVA), Charlottesville, VA, USA. He received the M.S. and Ph.D. degrees from the University of Texas at Austin, Austin, TX, USA, and the B.S. degree from the Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.

His laboratory at UVA is called Virginia Image and Video Analysis (VIVA). They specialize in biological and biomedical image analysis problems. The research emphases of VIVA includes tracking, segmentation, representation, retrieval, classification, and enhancement. He has over 250 publications in the image analysis area including the books entitled *Biomedical Image Analysis: Tracking* and *Biomedical Image Analysis: Segmentation*. He serves as an Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING.