Semi-Automated Reconstruction of Curvilinear Structures in Noisy 2D Images and 3D Image Stacks

Engin Türetken [*] Fethallah Benmansour Pascal Fua
Computer Vision Laboratory (EPFL), Lausanne, Switzerland
CH-1015 Lausanne, Switzerland
{engin.turetken,fethallah.benmansour,pascal.fua}@epfl.ch

Abstract

We propose a new approach to semi-automated delineation of curvilinear structures in a wide range of imaging modalities. Earlier approaches lack robustness to imaging noise, do not provide accurate radius estimates for the structures and operate only on single channel images. In contrast, ours makes use of the color information, when available, and generates accurate and smooth paths with minimal supervision. We demonstrate the applicability and generality of our approach on a wide range of imaging modalities ranging from a 2D dataset of aerial images to various 3D datasets of micrographs.

1. Introduction

Modern microscopy and imaging techniques produce large volumes of data of complex curvilinear structures such as vascular and neuronal arbors. Manually delineating them in image stacks typically requires hours to days of tedious work. Reconstructing them automatically and reliably has therefore emerged as a pressing need for a variety of scientific disciplines. For example, in the DIADEM competition [2], the lack of powerful and effective computational tools to automatically reconstruct neuronal arbors has been recognized as a major technical bottleneck in neuroscience research. The same argument also applies to several other medical fields.

Despite substantial research efforts and recent advances in reconstruction techniques and computing hardware, automated approaches still do not match human performance in terms of quality on noisy and complex data. This is, in part, due to the paucity of ground truth delineations available to serve as training data and as a baseline for algorithm evaluation. As a result of these needs and the need for high quality delineations, a wealth of semi-automated tools have emerged.

Although significant progress has been made over the last decade, current approaches to semi-automated delineation still lack robustness to noise and complex appearance, can't produce accurate radius estimates for curvilinear structures and don't make use of available color information. As a result, many approaches are rather limited to a small number of imaging modalities.

In this work, we address these issues using a two-step approach. The first step involves computing a scale-space representation of the input image using a local *tubularity measure*, which quantifies the likelihood that there exist a curvilinear structure of a certain radius at a pixel. We propose a new measure based on the Optimally Oriented Flux (OOF) descriptor [20]. Unlike the OOF measure that requires the filaments to have circular cross-sections, ours can handle arbitrary shaped cross-section profiles by leveraging the OOF response along multiple directions and radius values. Furthermore, we use an additional color-based term to suppress background noise and nearby irrelevant structures. The second step comprises of successively connecting pairs of user-provided seed points by tubular geodesics that follow the underlying curvilinear structures. This is achieved by applying the fast marching algorithm in the metric scale-space defined by the tubularity scores, which we weight by a color term to enforce color uniformity along the reconstructed paths.

We have demonstrated the effectiveness and versatility of our approach on a wide range of modalities from brightfield and brainbow micrographs of neurons to aerial images of road networks. Our software is incorporated in the tracing tool Simple Neurite Tracer (SNT) [26] and will be made publicly available as open source Fiji [36] plugins.

2. Related Work

Most delineation techniques and curvilinear structure databases use a sequence of cylindrical compartments as a representation of these structures [46, 10, 30, 26, 31, 44, 40,

[*] This work was supported in part by a grant from the Swiss National Science Foundation.

7, 16, 11]. Besides being compact, this geometric representation captures two important properties, namely connectedness and directedness, which are common among many structures such as blood vessels and neurons and are essential in studying their morphology. With respect to the level of user input required to build the compartments, current approaches can be roughly categorized into three major classes: manual, semi-automated and automated.

Manual delineation tools require the compartments or their joint points to be sequentially provided by the user starting from the structure roots such as soma or optic disc and ending at branch tips [13, 6, 7, 30, 5]. Since they are quite labor intensive and time consuming, these tools are most often used for proof-editing the reconstructions obtained by more automated tools.

Automated approaches, on the other hand, require at most one seed point (usually the root) for each connected structure of user's interest. Alternatively, these initial seeds and some additional ones along the filaments can be fully automatically generated by finding the local maxima of a *tubularity measure*, such as functionals of Hessian and Oriented Flux matrices [24, 35, 15, 20], which signifies how likely an image location is along the centerline of a curvilinear structure. Starting from the seeds, they then grow branches that follow high tubularity paths in the image. Depending on how the final solution is obtained, existing algorithms can be categorized broadly into three main groups: greedy tracking methods that incrementally add new paths to the current solution [1, 3, 29, 8], segmentation methods that first threshold the tubularity scores and then skeletonize or voxel code the resulting binary images [22, 18, 47, 46, 34, 42, 48, 10, 9], and finally graph-based methods that first build a graph of candidate paths and then find the solution in one shot through global optimization [14, 49, 44, 41, 40, 39].

Semi-automated techniques are similar in the sense that they also use the tubularity scores to link the seeds. However, in addition to the roots, they require additional seed points along the structure branches to be specified by the user [28, 12, 38, 31, 26, 30, 5]. Most such methods employ an interactive sequential procedure, where the user specifies one seed point at a time and the algorithm constructs a high-tubulariy path that links the given seed to the current reconstruction. Other techniques include those that prompt the user for a new seed only when the algorithm is stuck [38] or require only the root and the branch-terminal points [32, 31, 5].

Table 1 presents an incomplete list of most of the existing software tools for reconstructing curvilinear structures. The columns summarize several key features such as the tubularity measure used and the level of automation. As can be seen from the table, only few tools make use of the color information and allow delineating both 2D and 3D images. Furthermore, most of them don't produce accurate centreline location and radius estimates and can operate only on certain imaging modalities.

To address these problems, we've developed a simple and generic tool that supports delineating both color and grey scale images of various modalities in 2D and 3D. Our approach produces high-quality reconstructions such as the ones illustrated in Fig. 1(c) with minimum user interaction.

3. Method

Our interactive reconstruction algorithm goes through two steps, namely the tubularity computation and the interactive tracing, as depicted by Fig. 1 for three different modalities. We first compute a *tubularity value* at each image location $\mathbf{x}$ and possible radius value r within a range $[r_{min}\ r_{max}]$. It quantifies the likelihood that there exists a curvilinear structure of radius r, at location $\mathbf{x}$. Given an N-D image, this step creates an $(N+1)$-D scale- space tubularity image such as the ones shown in Fig. 1(b), where the scale dimension is color coded. We then use the Fast Marching algorithm [37] to iteratively trace, in the scale space, the curvilinear structure branches that lie between pairs of user-provided points.

Our tubularity measure extends the well-known Optimally Oriented Flux (OOF) [20] filter, which we first review briefly for completeness. We next introduce our approach to handling the irregularity of curvilinear structure cross-sections and then extend it to color images. Finally, we describe the interactive tracing step, which relies on the precomputed tubularity images.

For simplicity's sake and without loss of generality, we will assume that curvilinear structures are brighter than the background, and deal with 3-D image stacks, hence $N = 3$.

3.1. Optimally Oriented Flux

As discussed in Section 2, many existing approaches to enhancing curvilinear structures [24, 15, 35] rely on a Hessian based measure obtained by convolving the image with second order Gaussian derivatives. This involves some amount of smoothing. As a result, they take into account image intensities in the vicinity of the structures, which can adversely affect their accuracy in the presence of adjacent structures. Furthermore, in order to estimate the structure radius from the Gaussian standard deviation, they use an idealized intensity profile model, which is not trivial to obtain and is not applicable to all imaging modalities.

The OOF filter addresses these issues by considering intensity values only within a spherical volume of certain diameter, which provides a good estimate of the structure width [20]. The filter is computed by convolving the second derivatives of the image with the indicator function of

Tool	Type	Output	Tubularity	Platf.	Lang.	Dim.	Color	Radii	3D P.	3D V.	Code	Free
AxonTracker [38]	S	T: Greedy tracking	Gradient Vector Flow (GVF) and smoothness prior	W	C++	3D	no	no	yes	yes	no	yes
Imaris [5]	M/S/A	T: Fast marching minimal path in image space	Hessian functional	M/W	unknown	2D/3D	no	yes	yes	yes	no	no
Farsight [45, 44]	A	T: Open-curve snake	GVF and regularization priors	L/M/W	C++	3D	no	yes	yes	yes	yes	yes
Geodesic-SNT (ours)	S	T: Fast marching minimal path with a color prior in scale space	Multi-directional oriented flux (MDOF)	L/M/W	C++ and Java	2D/3D	yes	yes	yes	yes	yes	yes
HCA-Vision [43]	S	S	Directional template responses	W	C, C++ and C#	2D	no	no	N/A	N/A	no	no
NeuriteTracer [34]	A	S: Thresholding and skeletonization	Pixel intensities after illum. and contrast enhancement	L/M/W	ImageJ Macro	2D	no	no	N/A	N/A	yes	yes
Neurolucida [27]	S/A	T	unknown	W	unknown	2D/3D	unknown	yes	yes	yes	no	no
Neuromantic [30]	M/S	T: Dijkstra shortest path in image space	Hessian eigenvalues and geom. prior	W	C++	2D/3D	no	yes	yes	yes	yes	yes
Neuron_Morpho [6]	M	T	N/A	L/M/W	Java	2D/3D	no	no	no	no	yes	yes
NeuronGrowth [12]	S	T: Greedy tracking based on Hessian eigenvector directions	Hessian max. eigenvalue	L/M/W	Java	2D+t	no	no	no	no	no	yes
NeuronJ [28]	S	T: Dijkstra shortest path in image space	Hessian eigenvalues and geom. prior	L/M/W	Java	2D	no	no	N/A	N/A	no	yes
NeuronStudio [46]	A	T: Thresholding, skeletonization and Rayburst sampling for diameter estimation	Pixel intensity values after denoising and deconvolution	W	C	2D/3D	no	yes	yes	yes	yes	yes
NCTracer [9]	A	T: Voxel coding applied to binarized stacks	Center surround filter: Laplacian of Gaussian	W	Java and Matlab	3D	no	yes	yes	yes	no	yes
Reconstruct [13]	M/S	T: Region growing	A function of the region hue, saturation and brightness	W	C	2D/3D	yes	no	no	yes	yes	yes
Simple Neurite Tracer (SNT) [26]	S	T: Bidirectional A* search	Hessian functional	L/M/W	Java	3D	no	yes	yes	yes	yes	yes
TrakEM2 [7]	M	T (treeline object)	N/A	L/M/W	Java	2D/3D	no	no	no	yes	yes	yes
TREES Toolbox [10]	M/A	T: Thresholding and skeletonization	Pixel intensities	L/M/W	Matlab	2D/3D	no	yes	yes	yes	yes	yes
Vaa3D [32, 31]	S/A	T: Dijkstra shortest path in image space	An exponential function of the inverse pixel intensity	L/M/W	C++	3D	no	yes	yes	yes	yes	yes

Table 1. Existing tools for reconstructing curvilinear structures. From left to right, the columns list tool name with a reference, algorithm type (M: manual, S: semi-automated, A: Automated), output type and short description of the algorithm (T: tracing=vector graphics, S: segmentation=binary or label image), tubularity measure, supported platforms (L: Linux, M: Apple Macintosh, W: Microsoft Windows), language of implementation, supported image dimensions (i.e., application domain), whether color information is taken into account in the processing, whether radius estimates are automatically produced, whether processing takes place directly in 3D or done slice by slice (for 3D images only), whether 3D visualization of the reconstructions is supported, whether source code is publicly available, and whether the tool is free. For more detailed information, see the associated references and tool web-pages.

the sphere. More formally, its value $f(\mathbf{x}, \mathbf{p}, r)$ for image location $\mathbf{x}$, radius r, and orientation $\mathbf{p}$ is obtained by integrating the projected image gradients in the $\mathbf{p}$ direction over a sphere ∂S_r of radius r centered at $\mathbf{x}$. This is written as

$$f(\mathbf{x}, \mathbf{p}, r) = \frac{1}{4\pi r^2} \int_{\partial S_r} ((\nabla(G_{\sigma_0} * I)(\mathbf{x}+\mathbf{h}) \cdot \mathbf{p})\mathbf{p}) \cdot \mathbf{n} \mathrm{d}a, \quad (1)$$

where $\mathbf{n}$ is the outward unit normal of ∂S_r, G_{σ_0} is a regularizing Gaussian with a small standard deviation σ_0 typically equal to voxel spacing, $\mathbf{h} = r\mathbf{n}$ is the position vector with its tip on ∂S_r, and $\mathrm{d}a$ is the infinitesimal area on ∂S_r. The smaller the value of $f(\mathbf{x}, \mathbf{p}, r)$, the more likely it is that $\mathbf{x}$ is the center of a tube with radius r and orientation $\mathbf{p}$.

For each $\mathbf{x}$, we therefore look for the values of $\mathbf{p}$ and r that minimize f. It can be shown that $f(\mathbf{x}, \mathbf{p}, r)$ can be rewritten as the quadratic form $\mathbf{p}^T Q_{\mathbf{x},r} \mathbf{p}$, where $Q_{\mathbf{x},r}$ is known as the *oriented flux matrix* [20]. Using the divergence theorem, its entries can be expressed as

$$Q^{i,j}_{\mathbf{x},r} = \frac{1}{4\pi r^2} \left(\partial_{i,j} G_{\sigma_0}(\mathbf{x}) * \mathbb{1}_r * I(\mathbf{x}) \right), \quad (2)$$

where $\mathbb{1}_r$ is the indicator of the sphere of radius r. The OOF

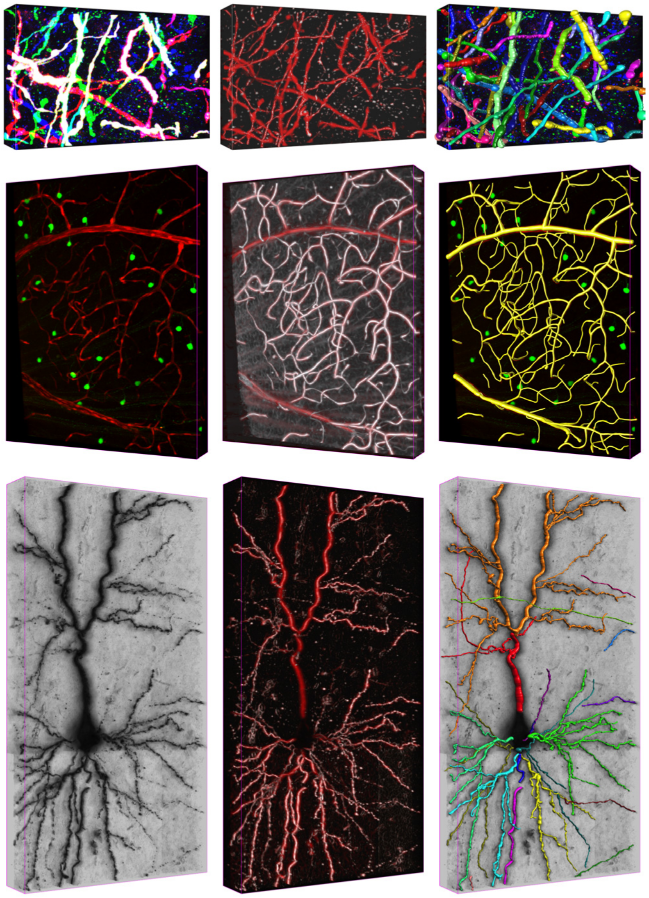

Figure 1. Overview of the approach. Each row corresponds to an image stack of different modality. From top to bottom: a brainbow image stack of neurite fibers [25], a confocal micrograph of blood vessels, which appear in red, and a brightfield image stack of neurons. (a) Original image stacks. (b) Maximum intensity projection of the scale-space tubularity images along the scale dimension. The images are color-coded so that higher tubularity values appear more saturated, going from white to red. (c) Reconstructions obtained with the interactive tracing procedure. For each connected curvilinear structure network (marked with a distinct color), the user selects a sequence of points in the image and the algorithm traces the curvilinear branches that lie between each consecutive pair of them.

tubularity measure is defined as the sum of the two OOF filter responses evaluated along two orthogonal directions $\mathbf{p}_1$ and $\mathbf{p}_2$ defining the structure's cross-sectional plane [20]:

$$\begin{aligned} f_{OF}(\mathbf{x}, r) &= \max_{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_1\perp\mathbf{p}_2} -f(\mathbf{x},\mathbf{p}_1,r) - f(\mathbf{x},\mathbf{p}_2,r) \\ &= \max_{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_1\perp\mathbf{p}_2} -\mathbf{p}_1^T Q_{\mathbf{x},r}\mathbf{p}_1 - \mathbf{p}_2^T Q_{\mathbf{x},r}\mathbf{p}_2 \ , \quad (3) \end{aligned}$$

which can be shown to be equal to the negative sum of the two smallest eigenvalues of $Q_{\mathbf{x},r}$. Their associated eigenvectors $\mathbf{e}^1_{\mathbf{x},r}$ and $\mathbf{e}^2_{\mathbf{x},r}$ provide the optimal directions for the above maximization.

Due to its intrinsic symmetry, the OOF filter yields high responses along centerlines of curvilinear structures at their associated scales. However, it also responds strongly to edges. To prevent this, a gradient antisymmetry function $g(\mathbf{x}, \mathbf{p}, r)$ was introduced in [21]:

$$g(\mathbf{x},\mathbf{p},r) = \frac{1}{4\pi r^2}\int_{\partial S_r} (\nabla(G_{\sigma_0} * I)(\mathbf{x}+\mathbf{h})\cdot\mathbf{p})\mathrm{d}a, \quad (4)$$

which is equal to the inner product $\mathbf{p}^T\mathbf{q}_{\mathbf{x},r}$. The term $\mathbf{q}_{\mathbf{x},r}$ is called the *oriented flux antisymmetry vector* (OFA) whose entries are computed as follows:

$$\mathbf{q}^i_{\mathbf{x},r} = \frac{1}{4\pi r^2}\left(\partial_i G_{\sigma_0}(\mathbf{x}) * \mathbb{1}_r * I(\mathbf{x})\right), \quad (5)$$

The absolute value of the antisymmetry function takes high values at structure boundaries and vanishes on perfectly symmetric shapes, for instance, along the centerline of a tube. Using this principle, the OOF tubularity measure can be refined to take into account structure symmetricity as follows [21]:

$$\begin{aligned} f_{OFA}(\mathbf{x},r) &= f_{OF}(\mathbf{x},r) - \sqrt{g(\mathbf{x},\mathbf{e}^1_{\mathbf{x},r},r)^2 + g(\mathbf{x},\mathbf{e}^2_{\mathbf{x},r},r)^2} \\ &= f_{OF}(\mathbf{x},r) - \sqrt{\mathbf{e}^{1T}_{\mathbf{x},r} H_{\mathbf{x},r}\mathbf{e}^1_{\mathbf{x},r} + \mathbf{e}^{2T}_{\mathbf{x},r} H_{\mathbf{x},r}\mathbf{e}^2_{\mathbf{x},r}} \end{aligned} \quad (6)$$

with $H_{\mathbf{x},r} = \mathbf{q}_{\mathbf{x},r}\mathbf{q}^T_{\mathbf{x},r}$. The two antisymmetry terms above act complementary to the OOF filter as they annihilate the OOF response away from the structure centerlines. However, since both OOF and OFA are evaluated only at two directions and a single radius value, the measure of Eq. 6 tends to favor circular and symmetric cross sections.

3.2. MDOF: Multi-Directional Oriented Flux

Curvilinear structures such as the ones shown in Fig. 1(a), often appear as irregular filaments in microscopic imagery due to reasons such as imaging noise, non-uniform staining and point-spread function. We address this issue by finding multiple directions and radii of maximal response of the joint oriented flux and antisymmetry operators. More specifically, given a set of predetermined radius levels S_r, we compute a number of direction and radius pairs $(\mathbf{p}_i \in \mathbb{R}^3, r_i \in S_r)$ that maximize

$$\begin{aligned} f_{MS}(\mathbf{x}) &= -\sum_{i=1}^{d} \mathbf{p}_i^T(Q_{\mathbf{x},r_i} + \alpha H_{\mathbf{x},r_i})\mathbf{p}_i \quad (7) \\ &= -\sum_{i=1}^{d} \mathbf{p}_i^T F_{\mathbf{x},r_i}\mathbf{p}_i \ \ , \end{aligned}$$

subject to the constraint that all $\mathbf{p_i}$ lie on the same plane at regular angular intervals.

As in Eq. 6, the matrix $H_{\mathbf{x},r_i}$ is equal to the quadratic form $\mathbf{q}_{\mathbf{x},r_i}\mathbf{q}^T_{\mathbf{x},r_i}$ of the antisymmetry vector as defined in Eq. 5 and α is a regularization parameter that determines the amount of symmetry enforced along each direction. It is set to 0.1 in all our experiments. Note that, in contrast to [21] which first finds the two dominant eigenvalues of $Q_{\mathbf{x},r}$ and then plugs their eigenvectors into the antisymmetry term, we jointly optimize for the oriented flux and the antisymmetry terms.

An important consideration when solving Eq. 7 is to do it fast because it needs to be done for each individual voxel. To this end, we employ a greedy approach, which first finds the dominant direction $\mathbf{p}_1^*$ and the associated radius r_1^* that maximizes $f_{MS}(\mathbf{x})$ for $d = 1$ and then $(\mathbf{p}_2^*, r_2^*)$ that maximizes the same function subject to the constraint that $\mathbf{p}_2^*$ is perpendicular to $\mathbf{p}_1^*$. The first step involves computing $\mathbf{p}_1^*$ as the eigenvector associated with the smallest eigenvalue among those of $F_{\mathbf{x},r_i}$, for all $r_i \in S_r$. Although solving the second step requires satisfying the linear constraints $\mathbf{p}_2^T\mathbf{p}_1^* = 0$ and $\mathbf{p}_2^T\mathbf{p}_2 = 1$, it can be done in closed form by solving the following eigenvalue problem [17]:

$$(\mathbf{e}^*, r_2^*) = \underset{\mathbf{e}, r_i \in S_r}{\operatorname{argmax}} \quad -\mathbf{e}^T P\; F_{\mathbf{x},r_i}\; P\;\mathbf{e}\;, \tag{8}$$

where the matrix P is defined as $P = I - \mathbf{p}_1^*\mathbf{p}_1^{*T}$. The vector $\mathbf{e}^*$ can be computed as the eigenvector associated with the smallest eigenvalue among those of $P\;F_{\mathbf{x},r_i}\;P$, for all $r_i \in S_r$. The optimal second direction $\mathbf{p}_2^*$ can then be found as $\mathbf{p}_2^* = P\;\mathbf{e}^*$ [17].

We then find the additional $d-2$ directions by sampling them at equiangular intervals to span π radians in the plane defined by $\mathbf{p}_1^*$ and $\mathbf{p}_2^*$. The corresponding radius values are found by maximizing the same cost function of Eq. 7 in the set S_r. This produces a 3-D score image, which we then extend to 4-D by adding the strongest response of $\mathbf{p}_i^T F_{\mathbf{x},r_i}\mathbf{p}_i$ for each radius level. We take the final score for radius r at location $\mathbf{x}$ to be

$$f_{MDOF}(\mathbf{x}, r) = -\lambda^1_{\mathbf{x},r} + \frac{1}{d}\max_{\{\mathbf{p}_i, r_i\}_{i=1}^d} f_{MS}(\mathbf{x}), \tag{9}$$

where $\lambda^1_{\mathbf{x},r}$ is the smallest eigenvalue of $F_{\mathbf{x},r}$. At each radius level, we use only a single eigenvalue, which gives the strongest response, because, in most datasets, the gradient information is strong and provides a reliable estimate only along one direction in the irregular cross-sectional profiles. For instance, in the brightfield stack shown in the last row of Fig. 1, this direction usually lies in the lateral (x-y) plane of the in-focus z slice. Taking into account the gradient in out-of-focus regions therefore usually results in overestimation of the scale and mislocalization of the structure centreline.

The second term in Eq. 9 measures the likelihood of $\mathbf{x}$ being on the centerline of a curvilinear structure, while the first one measures the contribution of the radius level r. As a result, at each image point $\mathbf{x}$, the radius level r that yields the strongest gradient flux on the associated sphere S_r will be assigned the highest score.

3.3. Handling Color Images

With the advent of new imaging techniques, color information has gained increasing importance in the analysis of complex curvilinear structures. The Brainbow [25] and STORM [19] techniques, for instance, allow labelling neuron cells in a tissue slice with multiple colors using a random mixture of few fluorescent proteins. This results in each neuron being labelled with a sufficiently distinct color, which is nearly constant along its axon and dendrites. In fact, color constancy property also widely holds for other curvilinear structures such as road networks.

We use this property to compute, for each image location $\mathbf{x}$, a similarity image, which signifies how likely both $\mathbf{x}$ and another location $\mathbf{y}$ in its vicinity belong to the same curvilinear structure based on their color. We then use the resulting similarity image to compute the tubularity measure of Section 3.2 at location $\mathbf{x}$. More precisely, given an RGB image stack I, we first convert it into the CIELAB space. We then compute the similarity image $J_\mathbf{x}$ for $\mathbf{x}$ as a function $s(.)$ of the color distance between $\mathbf{x}$ and $\mathbf{y}$, weighted by the value of the brightness image L, which is equivalent to max(R,G,B). We write

$$J_\mathbf{x}(\mathbf{y}) = L(\mathbf{y})\; s(I(\mathbf{x}), I(\mathbf{y})), \;\; \forall \mathbf{x}. \tag{10}$$

In this work, we take the similarity function s to be a multivariate Gaussian distribution with mean vector $I(\mathbf{x})$ and a diagonal covariance matrix with diagonal entries σ. In practice, computing the similarity image for each location is quite expensive. That is why we limit the number of similarity images to a number of color clusters, which we compute using the k-means algorithm. Let C be the set of resulting cluster centroids. Then, Eq. 10 becomes:

$$J_\mathbf{x}(\mathbf{y}) = L(\mathbf{y})\; s(c_{h(\mathbf{x})}, I(\mathbf{y})), \;\; \forall c_{h(\mathbf{x})} \in C, \tag{11}$$

where $h(\mathbf{x})$ is a function that returns the cluster label for location $\mathbf{x}$. Note that, Eq. 11 requires sequentially computing only $|C|$ similarity images, which serve as input to the tubularity computation described in the previous section. The result is that, for each location, only image evidence from nearby locations of similar color are taken into account.

3.4. Interactive Tracing

The interactive tracing starts by the user selecting a root point along a curvilinear structure and then an end point to connect it. It then proceeds sequentially by adding each time a new path that links a user-provided point to the rest of the delineation. The resulting reconstructions can either be trees without any cycles or loopy graphs whose branches intersect each other.

We compute tubular paths connecting pairs of points using a minimal path method applied in the scale space. A *geodesic tubular path* connecting a source point $\mathbf{x_s}$ to a target one $\mathbf{x_t}$ in the scale space can be obtained by first computing a *minimal action map* $U(\mathbf{x})$ using the Fast Marching algorithm [37], and then sliding back from $\mathbf{x_t}$ to $\mathbf{x_s}$ on this map using a gradient descent procedure [23, 4]. $U(\mathbf{x})$ is defined as the minimum energy integrated along a path between $\mathbf{x_s}$ and any other point $\mathbf{x}$:

$$U(\mathbf{x}) = \inf_{C(s) \in A_{\mathbf{x_s},\mathbf{x}}} \left\{ \int_{\Omega} \mathcal{P}\left(C(s)\right) \mathrm{d}s \right\}, \tag{12}$$

where $s \in \Omega = [0, 1]$ is the arc-length parameter, $C(s) \in \mathbb{R}^4$ is a curve parametrized along its length (i.e., $||C'(s) = 1||$), and $A_{\mathbf{x_s},\mathbf{x}}$ is the set of all paths linking $\mathbf{x}$ to $\mathbf{x_s}$. $\mathcal{P}$ is a potential function, which we take as the following exponential mapping of the tubularity measure:

$$\mathcal{P}(\mathbf{x}) = \exp(\alpha\, \mathcal{T}(\mathbf{x}) + \beta), \tag{13}$$

where $\mathcal{T}(\mathbf{x})$ is the scale-space tubularity score of Eq. 9 at point $\mathbf{x}$. The coefficients α and β are computed such that $\min\{\mathcal{P}(\mathbf{x}) \mid \forall \mathbf{x}\} = 1.0$ and $\max\{\mathcal{P}(\mathbf{x}) \mid \forall \mathbf{x}\} = T_{max} > 1.0$, where T_{max} is a smoothness parameter with smaller values corresponding to smoother paths. Its value is dynamically determined by the user during the tracing.

For color images, we apply a linear mapping to the tubularity scores to enforce the paths to pass through points of uniform color similar to the mean color of the source and the target points:

$$\mathcal{T}_{\mathbf{x_s},\mathbf{x_t}}(\mathbf{y}) = \mathcal{T}(\mathbf{y})\ \ s((I(\mathbf{x_s}) + I(\mathbf{x_t}))/2, I(\mathbf{y})), \tag{14}$$

where we use the same color similarity function s of Eq. 10.

4. Results

In this section, we first describe briefly our evaluation datasets and then present our results on them.

4.1. Datasets and Parameters

We evaluated our approach on six different datasets described below:

- *Aerial Images of Road Networks*: Color images of loopy road networks. We used 29 images to evaluate our approach. We sampled 14 radius levels linearly from [5 18] pixels to compute the scale-space tubularity volumes described in Section 3.2.
- *Brainbow Stacks of Visual Cortical Neurites*: Four micrographs of mice primary visual cortex acquired using the brainbow staining technique [25]. We used 9 radius levels uniformly sampled from the range [3 11] voxels.
- *Confocal Stacks of Olfactory Projection Fibers*: Eight image stacks of Drosophila fly brains taken from the DIADEM challenge [2]. We used 11 radius levels ranging between 1 to 6 voxels.
- *Confocal Stacks of Retinal Vasculatures*: Two confocal micrographs of direction selective retinal ganglion cells and loopy vasculature networks. We sampled 9 radius levels from the range [3 16] voxels.
- *Brightfield Stacks of Neurons*: Five brightfield image stacks of biocytin-stained rat brains. We sampled 12 radius levels from 1 to 12 voxels.
- *Brightfield Stacks of Visual Cortical Neurites*: Six brightfield micrographs of biocytin-labeled cat primary visual cortex layer 6 taken from the DIADEM challenge [2]. We sampled 12 radius levels from 1 to 12 voxels.

We used the same parameters mentioned in Section 3 for all the six datasets: the number of direction and radius pairs $L = 10$, color similarity sigma $\sigma = 50$ in the the CIELAB space and the number of color clusters $|C| = 50$. We set the smoothness parameter T_{max} to $1e^3$ and very rarely needed to adjusted its value.

Our implementation of the MDOF measure employs the Fast Fourier Transform (FFT) algorithm to efficiently compute the entries of the oriented flux matrix and the antisymmetry vector of Section 3.1. Our software uses the ITK library and will be made available as open-source.

4.2. Reconstructions

Fig. 2 shows representative results on the *aerial*, *brainbow* and the two *confocal* datasets. We used the color information available for the *aerial* and the *brainbow* datasets to compute the tubularity measure of Section 3.2. For the *confocal* stacks of vasculatures, we only considered the red channel since it is the only one used to label the blood vessels.

The *brainbow* and the *confocal* fibers form tree structures without cycles. The road networks and the blood vessel arbors, however, are inherently loopy, which is why their reconstructions contain cycles.

As can be observed from the *aerial* and the *brainbow* images, radius estimates along the filaments undergo abrupt changes. In the case of the road networks, this is because trees occlude and cast shadows on them and in the case of brainbow neurites, it is because of the high density of blob-like synapses. In both cases, the regularization parameter T_{max} can be reduced to obtain smoother reconstructions.

In contrast, neurites and vasculatures in the *confocal* stacks have smoothly changing scale values. Furthermore, these stacks are fairly clean with a relatively high signal-to-noise ratio and a small point spread function (PSF). As a result, their networks are almost perfectly recovered both in terms of centreline location and scale accuracy.

The neurites in the brightfeld stacks shown in Fig. 3, however, appear as faint and irregular filaments due to the staining and imaging techniques used. Furthermore, the strong PSF distorts their cross-sectional profile with halo-like shadows. Despite these adverse factors, our approach consistently produces high quality reconstructions with minimal user input.

5. Conclusion

We have presented an interactive approach to delineation of curvilinear structure networks and demonstrated its effectiveness on a wide range of 2D and 3D imaging modalities. While existing approaches can only operate on a single color channel and require the structures to have circular cross-sections, ours can handle irregular structures both in gray-scale and color images. This results in a simple and generic tool, which we plan to release as open source and platform independent Fiji [36] plugins.

The current approach, however, can not handle large image stacks mainly due to the demanding scale-space processing. Future work will therefore focus on sparse processing techniques and data structures that will reduce memory requirements and computational complexity.

References

[1] K. Al-Kofahi, S. Lasek, D. Szarowski, C. Pace, G. Nagy, J. Turner, and B. Roysam. Rapid Automated Three-Dimensional Tracing of Neurons from Confocal Image Stacks. TITB, 6(2):171–187, 2002. 2

[2] G. A. Ascoli, K. Svoboda, and Y. Liu. Digital Reconstruction of Axonal and Dendritic Morphology Diadem Challenge, 2010. http://diademchallenge.org/. 1, 6

[3] E. Bas and D. Erdogmus. Principal Curves as Skeletons of Tubular Objects - Locally Characterizing the Structures of Axons. Neuroinformatics, 9(2-3):181–191, 2011. 2

[4] F. Benmansour and L. Cohen. Tubular Structure Segmentation Based on Minimal Path Method and Anisotropic Enhancement. IJCV, 92(2):192–210, 2011. 6

[5] Bitplane. Imaris: 3D and 4D Real-Time Interactive Image Visualization, 2013. http://www.bitplane.com/go/products/imaris/. 2, 3

[6] K. Brown, D. E. Donohue, G. DAlessandro, and G. A. Ascoli. A cross-platform freeware tool for digital reconstruction of neuronal arborizations from image stacks. Neuroinformatics, 3(4):343–359, 2005. 2, 3

[7] A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tomancak, V. Hartenstein, and R. J. Douglas. TrakEM2 Software for Neural Circuit Reconstruction. PLoS One, 7(6):e38011, 2012. 2, 3

[8] A. Choromanska, S. Chang, and R. Yuste. Automatic Reconstruction of Neural Morphologies with Multi-Scale Graph-Based Tracking. Frontiers in Neural Circuits, 6(25), 2012. 2

[9] P. Chothani, V. Mehta, and A. Stepanyants. Automated Tracing of Neurites from Light Microscopy Stacks of Images. Neuroinformatics, 9:263–278, 2011. 2, 3

[10] H. Cuntz, F. Forstner, A. Borst, and M. Häusser. One rule to grow them all: a general theory of neuronal branching and its practical application. PLoS Computational Biology, 6(8):e1000877, 2010. 2, 3

[11] Duke University and University of Southampton. Duke-Southampton Archive of Neuronal Morphology. http://neuron.duke.edu/cells/. 2

[12] Z. Fanti, F. F. De-Miguel, and M. E. Martinez-Perez. A Method for Semiautomatic Tracing and Morphological Measuring of Neurite Outgrowth from DIC Sequences. In Engineering in Medicine and Biology Society, pages 1196–1199, 2008. 2, 3

[13] J. C. Fiala. Reconstruct: a free editor for serial section microscopy. Journal of microscopy, 218:52–61, 2005. 2, 3

[14] M. Fischler, J. Tenenbaum, and H. Wolf. Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique. CVIP, 15(3):201–223, March 1981. 2

[15] A. Frangi, W. Niessen, K. Vincken, and M. Viergever. Multiscale Vessel Enhancement Filtering. Lecture Notes in Computer Science, 1496:130–137, 1998. 2

[16] George Mason University. NeuroMorpho.Org. http://neuromorpho.org. 2

[17] G. Golub. Some Modified Matrix Eigenvalue Problems. SIAM Review, 15(2):318–334, 1973. 5

[18] K. P. gyi and A. Kuba. A 3D 6-subiteration thinning algorithm for extracting medial lines. Pattern Recognition Letters, 19(7):613–627, 1998. 2

[19] M. Lakadamyali, H. Babcock, M. Bates, X. Zhuang, and J. Lichtman. 3D Multicolor Super-Resolution Imaging Offers Improved Accuracy in Neuron Tracing. PloS ONE, 7(1):e30826, 2012. 5

[20] M. Law and A. Chung. Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux. In ECCV, 2008. 1, 2, 3, 4

[21] M. Law and A. Chung. An Oriented Flux Symmetry Based Active Contour Model for Three Dimensional Vessel Segmentation. In ECCV, pages 720–734, 2010. 4, 5

[22] T. Lee, R. Kashyap, and C. Chu. Building Skeleton Models via 3D Medial Surface Axis Thinning Algorithms. CVGIP: Graphical Models and Image Processing, 56(6):462–478, 1994. 2

[23] H. Li and A. Yezzi. Vessels as 4-D Curves: Global Minimal 4D Paths to Extract 3-D Tubular Surfaces and Centerlines. TMI, 26(9):1213–1223, 2007. 6

[24] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. In CVPR, page 465, 1996. 2

[25] J. Livet, T. Weissman, H. Kang, R. Draft, J. Lu, R. Bennis, J. Sanes, and J. Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. Nature, 450(7166):56–62, 2007. 4, 5, 6

[26] M. Longair, D. A. Baker, and J. D. Armstrong. Simple Neurite Tracer: open source software for reconstruction, visualization and analysis of neuronal processes. Bioinformatics, 27(17):2453–2454, 2011. 1, 2, 3

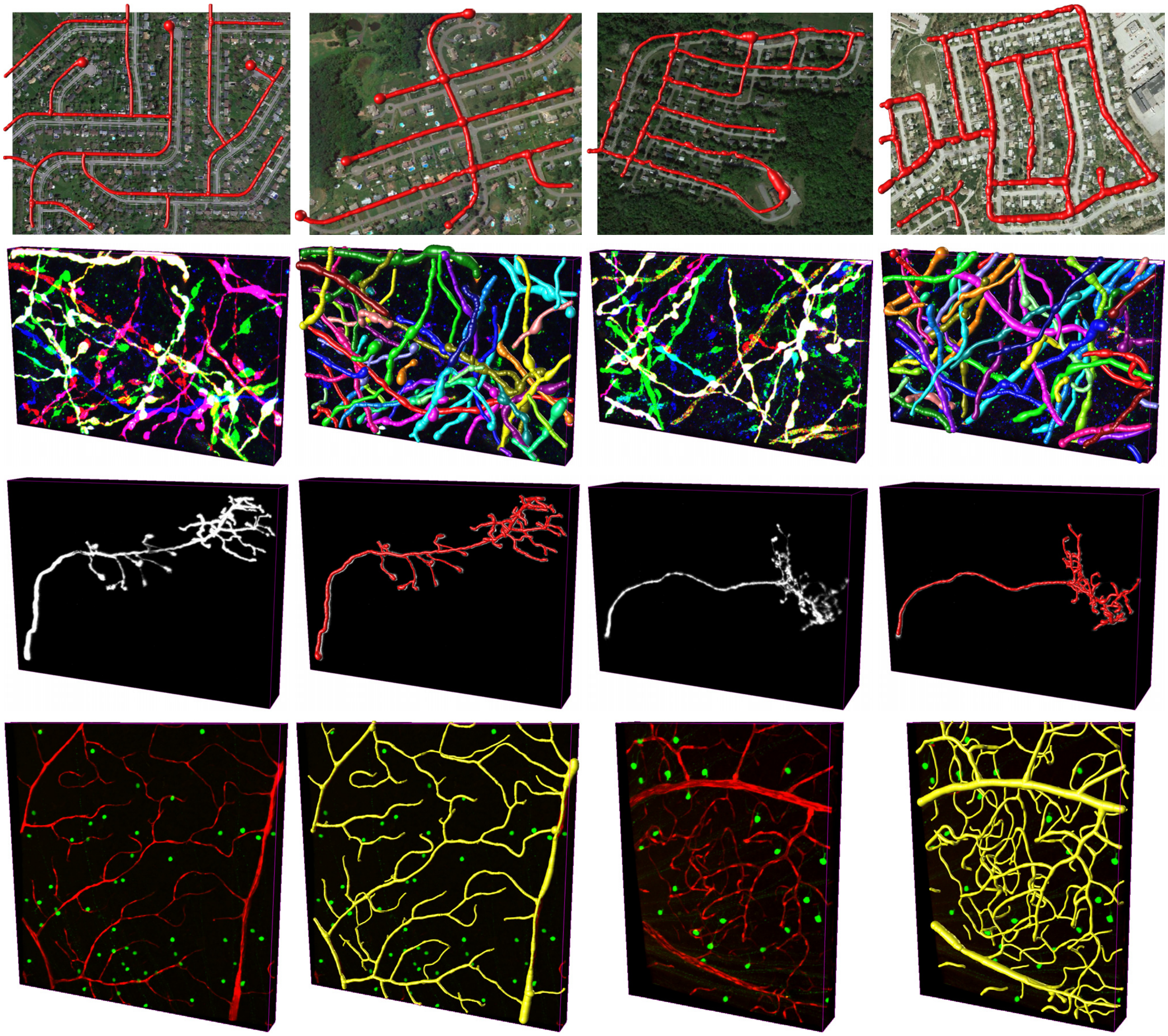

Figure 2. Reconstruction results, best viewed in color. **Top Row:** Four road images with final reconstructions shifted and overlaid. **Bottom Rows:** Two maximal intensity projections and overlaid reconstructions for the brainbow and the confocal datasets in the order described in Section 4.1. All the renderings are created using the Vaa3D software [33].

[27] MBF Bioscience. Neurolucida: Microscope Systems for Stereology and Neuron Morphology, 2013. http://www.mbfbioscience.com/neurolucida/. 3

[28] E. Meijering, M. Jacob, J.-C. F. Sarria, P. Steiner, H. Hirling, and M. Unser. Design and Validation of a Tool for Neurite Tracing and Analysis in Fluorescence Microscopy Images. Cytometry Part A, 58A(2):167–176, April 2004. 2, 3

[29] A. Mukherjee and A. Stepanyants. Automated Reconstruction of Neural Trees Using Front Re-initialization. In SPIE, volume 8314, 2012. 2

[30] D. R. Myatt, T. Hadlington, G. A. Ascoli, and S. J. Nasuto. Neuromantic - from semi manual to semi automatic reconstruction of neuron morphology. Frontiers in Neuroinformatics, 6(4), 2012. 2, 3

[31] H. Peng, F. Long, and G. Myers. Automatic 3D Neuron Tracing Using All-Path Pruning. Bioinformatics, 27(13):239–247, 2011. 2, 3

[32] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson. Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model. Bioinformatics, 26(12):i38–i46, 2010. 2, 3

[33] H. Peng, Z. Ruan, F. Long, J. Simpson, and E. Myers. V3D Enables Real-Time 3d Visualization and Quantitative Analysis of Large-Scale Biological Image Data Sets. Nature Biotechnology, 28(4):348–353, 2010. 8

[34] M. Pool, J. Thiemann, A. Bar-Or, and A. E. Fournier. NeuriteTracer: A Novel Imagej Plugin for Automated Quantification of Neurite Outgrowth. Journal of Neuroscience Methods, 168(1):134–139, 2008. 2, 3

[35] Y. Sato, S. Nakajima, H. Atsumi, T. Koller, G. Gerig,

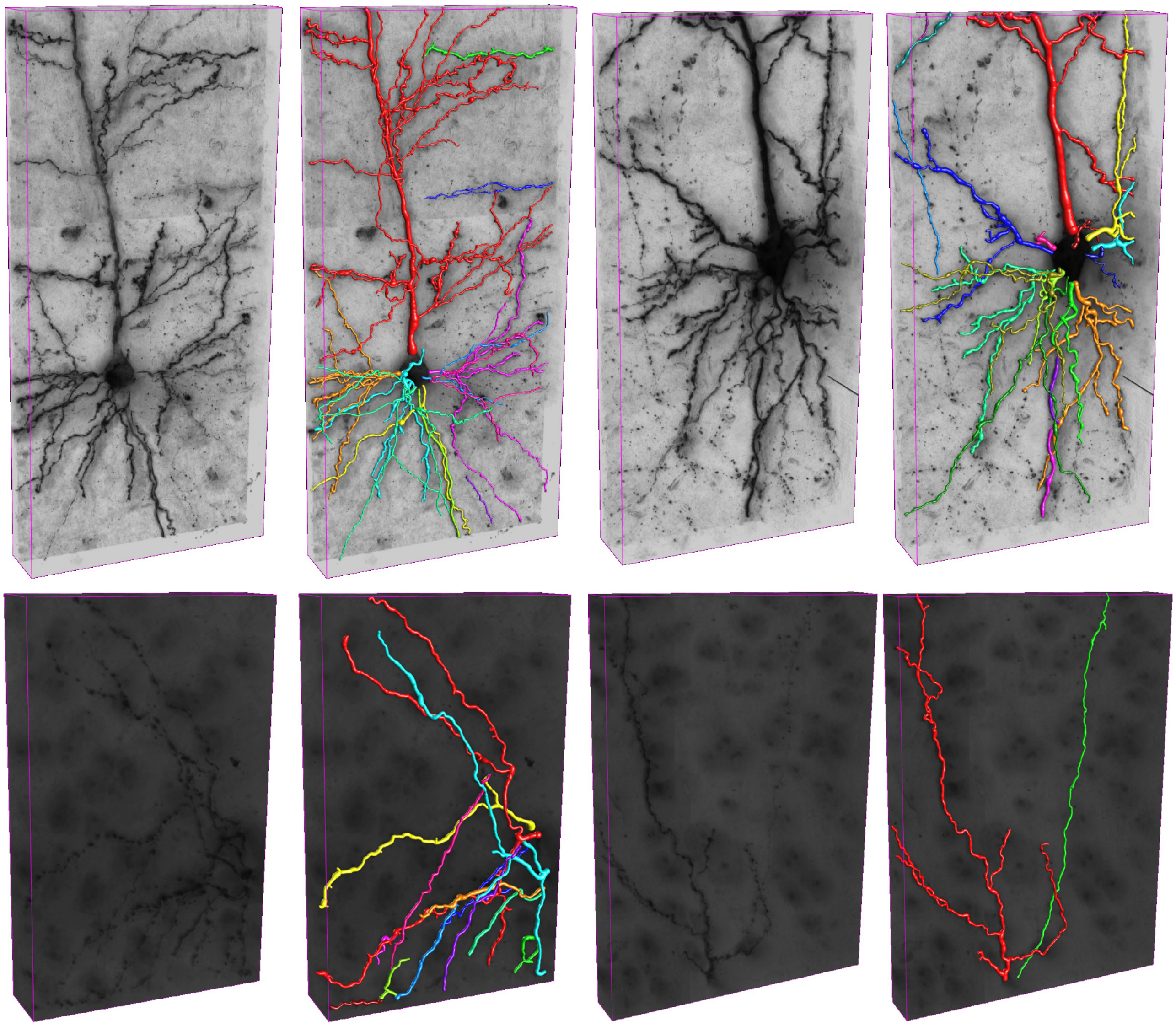

Figure 3. Reconstruction results for the brightfield datasets. Best viewed in color. Top and bottom rows show two minimal intensity projections and overlaid reconstructions for the neurons and visual cortical neurites of Section 4.1, respectively.

S. Yoshida, and R. Kikinis. 3D Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. MIA, 2:143–168, June 1998. 2

[36] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, and B. Schmid. Fiji: an open-source platform for biological-image analysis. Nature Methods, 9(7):676–682, 2012. 1, 7

[37] J. A. Sethian. Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, 1999. 2, 6

[38] R. Srinivasan, Q. Li, X. Zhou, J. Lu, J. Lichtman, and S. T. C. Wong. Reconstruction of the Neuromuscular Junction Connectome. Bioinformatics, 26(12):i64–i70, 2010. 2, 3

[39] E. Turetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua. Reconstructing Loopy Curvilinear Structures Using Integer Programming. In CVPR, June 2013. 2

[40] E. Turetken, F. Benmansour, and P. Fua. Automated Reconstruction of Tree Structures Using Path Classifiers and Mixed Integer Programming. In CVPR, June 2012. 2

[41] E. Turetken, G. Gonzalez, C. Blum, and P. Fua. Automated Reconstruction of Dendritic and Axonal Trees by Global Optimization with Geometric Priors. Neuroinformatics, 9(2-3):279–302, 2011. 2

[42] Z. Vasilkoski and A. Stepanyants. Detection of the Optimal Neuron Traces in Confocal Microscopy Images. Journal of Neuroscience Methods, 178(1):197–204, 2009. 2

[43] D. Wang, R. Lagerstrom, C. Sun, L. Bishof, P. Valotton, and M. Gtte. HCA-Vision: Automated Neurite Outgrowth Analysis. Journal of Biomolecular Screening, 15(9):1165–1170, 2010. 3

[44] Y. Wang, A. Narayanaswamy, and B. Roysam. Novel 4D Open-Curve Active Contour and Curve Completion Approach for Automated Tree Structure Extraction. In CVPR, pages 1105–1112, 2011. 2, 3

[45] Y. Wang, A. Narayanaswamy, C. Tsai, and B. Roysam. A Broadly Applicable 3D Neuron Tracing Method Based on Open-Curve Snake. Neuroinformatics, 9(2-3):193–217, 2011. 3

[46] S. Wearne, A. Rodriguez, D. Ehlenberger, A. Rocher, S. Henderson, and P. Hof. New Techniques for Imaging, Digitization and Analysis of Three-Dimensional Neural Morphology on Multiple Scales. Neuroscience, 136(3):661–680, 2005. 2, 3

[47] C. Weaver, P. Hof, S. Wearne, and L. Brent. Automated Algorithms for Multiscale Morphometry of Neuronal Dendrites. Neural Computation, 16(7):1353–1383, 2004. 2

[48] J. Xu, J. Wu, D. Feng, and Z. Cui. Dsa Image Blood Vessel Skeleton Extraction Based on Anti-Concentration Diffusion and Level Set Method. Computational Intelligence and Intelligent Systems, 51:188–198, 2009. 2

[49] T. Zhao, J. Xie, F. Amat, N. Clack, P. Ahammad, H. Peng, F. Long, and E. Myers. Automated Reconstruction of Neuronal Morphology Based on Local Geometrical and Global Structural Models. Neuroinformatics, 9:247–261, May 2011. 2