Delineating Trees in Noisy 2D Images and 3D Image-Stacks

Germán González [*]

CVLab, EPFL

CH-1015 Lausanne,

Switzerland

german.gonzalez@epfl.ch

Engin Türetken

CVLab, EPFL

CH-1015 Lausanne,

Switzerland

engin.turetken@epfl.ch

François Fleuret [*]

IDIAP Research Institute

Rue Marconi 19,

Martigny, Switzerland

francois.fleuret@idiap.ch

Pascal Fua

CVLab, EPFL

CH-1015 Lausanne,

Switzerland

pascal.fua@epfl.ch

Abstract

We present a novel approach to fully automated delineation of tree structures in noisy 2D images and 3D image stacks. Unlike earlier methods that rely mostly on local evidence, our method builds a set of candidate trees over many different subsets of points likely to belong to the final one and then chooses the best one according to a global objective function. Since we are not systematically trying to span all nodes, our algorithm is able to eliminate noise while retaining the right tree structure.

Manually annotated dendrite micrographs and retinal scans are used to evaluate the performance of our method, which is shown to be able to reject noise while retaining the tree structure.

1. Introduction

Tree-like structures appear at many different scales and in many different contexts. They can be micrometer scale dendrites in light microscopy image-stacks, centimeter-scale blood vessels in retinal scans, or meter-scale road networks in aerial images. Extracting them automatically and robustly is therefore of fundamental relevance to many scientific disciplines. However, even though the topic has received sustained attention ever since the inception of the field of Computer Vision, both robustness and automation remain elusive. Fully automated techniques exist but require very clean data; substantial amounts of manual intervention is required for any other kind.

In this paper, we argue that this is in large part because current approaches are far too local and do not sufficiently take into account the global tree structure when making decisions. By constructing a set of plausible trees and selecting the one with the highest global score, we achieve good
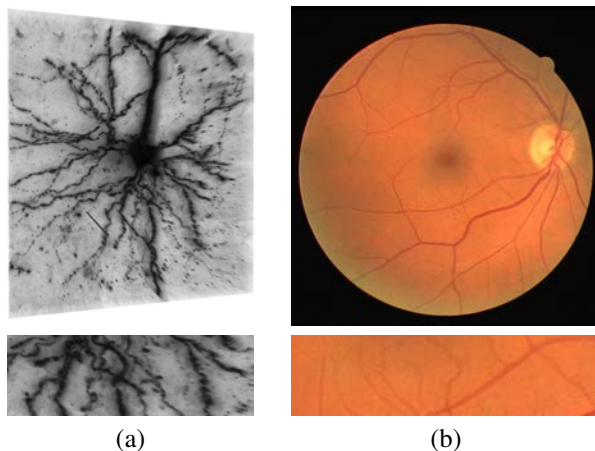
(a)   (b)

Figure 1. Image data used to evaluate our method. (a) 3D volume rendering of a neuron micrograph and zoomed-in view of a detail. The dendrites appear as irregular and often disconnected elongated structures. The data is noisy due to staining artifacts, non-Gaussian blur and irrelevant structures close to the neuron. (b) 2D scan of retinal blood vessels. The main vessels appear as clean structures, but the secondary ones have very low contrast and can easily be lost in the background.

results on very noisy datasets, such as the ones depicted by Fig. 1.

More specifically, we use 3D-steerable features [12] to assign to image points, and to edges connecting them, probabilities of belonging to the tree. We then select evenly spaced high-probability points that we treat as anchor points and connect them using a maximum-probability path. This turns our set of N anchor points into a weighted graph in which we look for minimum-weight trees that span K of its N vertices [3, 11]. We do this for a range of K values and select the resulting tree with the highest global probability. This is in contrast to more traditional approaches [8, 13] that would use a minimum spanning tree to link all vertices and then prune the branches that do not conform to a shape or image appearance criterion. Such methods can eliminate spurious branches but cannot recover from incorrect

connectivity in the initial spanning tree. When K is in the right range, the tree spanning only K vertices does not suffer from this problem, and our probabilistic score provides us with an automated way to select K.

This results in a generic and fully automated technique that we demonstrate on micrographs of eight different neurons and on a database of retinal image scans. In both cases, we have access to ground truth data to validate our results.

2. Related Work

Early 2D-image delineation approaches such as [9], as well as far more recent ones [5], involved first generating a binary overlay of points likely to belong to the linear structures, linking them using a Minimum Spanning Tree, and then pruning the tree to retain only the branches most likely to resemble the target features. While potentially effective, such approaches require complex heuristics and the results need to be edited by hand or using semi-automated tools [8]. A popular alternative has been to use semi-automated techniques in which the user is asked to supply anchor points manually and the system proposes a path linking them, which can then be edited if needed [18].

Algorithms designed to trace 3D dendritic trees in image stacks face many of the same difficulties. Most approaches rely on a *dendriteness* measure of how dendrite-like filaments look. This measure can be postulated [10, 16], optimized given a mathematical dendrite model [15, 18], or learned [21, 12] from training data. It can then be used in one of three ways.

First, filaments can be grown from seed points [1, 17], which has been successfully demonstrated in confocal fluorescent microscopy images. This approach is computationally effective because the dendriteness of filaments need only be evaluated for a small subset of the voxels. However, heuristics are usually required to detect branching points. Worse, such methods may easily fail in noisy data because of their incremental nature: If the growing process diverges at one voxel, the rest of the dendritic tree will be lost.

Second, one can find paths of maximum dendriteness between seed points, which are usually supplied by hand [18, 6, 23, 2]. This can be done using active contour models, geometric constraints, or the live-wire algorithm [19]. This yields good results but is manually intensive.

Third, one can first segment the image volume and then find the skeleton of the resulting binary mask [24]. This requires high-quality segmentations, which can be obtained only in very clean data. Even then, pruning may be required to remove small artifacts [4] and perform topology changes.

By contrast to these methods that rely almost exclusively on local evidence, our approach optimizes an objective function that infers the most probable tree given the whole image data. This eliminates drift, explicitly enforces proper connectivity, and is fully automated.
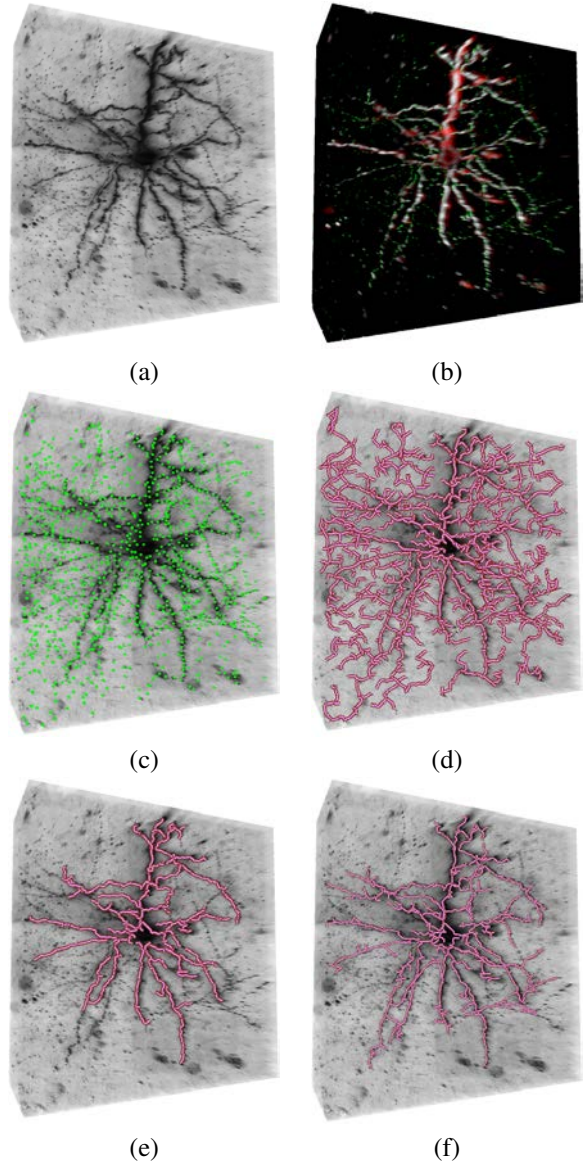

Figure 2. Delineating dendritic trees in brightfield micrographs. (a) Minimum intensity projection of an image stack. Each pixel in the image is the minimum intensity value of the voxels that are touched by the ray cast from the camera through the pixel. (b) The dendriteness computed for individual voxels. The colors represent responses at different scales. (c) The anchor points, which are local maxima of dendriteness. (d) The minimum spanning tree computed over all points of (c). (e) A K-MST for a sub-optimal value of K. (f) The K-MST estimated by Eq. 19.

3. Method

For clarity of explanation, we focus on the problem of dendrite detection in 3D brightfield images, although the method is equally applicable to other image modalities and dimensionalities.

Fig. 2 depicts our method, which goes through the fol-

lowing steps:

1. We compute the *dendriteness* of every voxel, as shown in Fig. 2(b). It encodes how likely it is to be on the centerline of a dendrite. It relies on 3D-steerable features [14] designed to account for the fact that dendrites come in many different widths.

2. We map the *dendriteness* values to probabilities and treat high probability voxels that are local maxima as anchor points. They are shown in Fig. 2(c).

3. We compute the most probable paths between pairs of nearby anchor points and assign them probabilistic costs that are lowest when all pixels along them are dendrite-like.

4. We compute the lowest-cost tree among those that span K of the N anchor points for a wide range of $K \leq N$. This is known as the *K-Minimum Spanning Tree* (K-MST) or *K-Cardinality Tree* (KCT) problem . Even though it is NP-Hard, approximate solutions can nevertheless be computed efficiently and fast [3, 11].

5. This yields trees such as those of Fig. 2(e,f). We select the one that optimizes a global objective function.

Steps 4 and 5 are those that most distinguish our approach from more traditional ones that either build trees spanning all the anchor points and then attempt to eliminate spurious branches, or grow the tree incrementally at the risk of propagating errors. We avoid these problems by minimizing a well-defined global objective function, as the minimum spanning tree approaches do, but with the possibility to explore different topologies because we do not force the tree to systematically connect all vertices.

The approach is exactly the same in regular 2D images, except that voxels should be replaced by pixels and 26-connectivity should be replaced by 8-connectivity.

In the remainder of this section, we first formalize our approach, define our objective function, and argue that, given the huge amounts of data we have to process, the proposed algorithm represents a feasible approach to computing an approximate minimum. In Section 4, we will use ground truth data to show that this minimum is of very good quality indeed.

3.1. Formulation

We map the image data to a grid graph $G = (V, E)$ whose vertices $V = \{x_{ij}\}$ are the voxels and whose undirected edges $E = \{e_{ij}\}$ connect each voxel to its 26 neighbors. A subgraph $\mathbf{s}$ is represented by a set of indicator variables $\{s_{ij}\}$, one for each edge of G, such that

$$\forall e_{ij} \in E, \quad s_{ij} = \begin{cases} 1, & if \quad e_{ij} \in \mathbf{s} \\ 0, & otherwise. \end{cases} \quad (1)$$

For the purpose of this discussion, let us assume that we can associate to each edge e_{ij} a probability $p_{ij} = P(s_{ij} = 1|I)$ of belonging to the skeleton of a dendrite given the image data I. We will propose in Section 3.3 a mechanism for evaluating it.

We take the subgraph we look for to be $\mathbf{t}^*$, the tree of G whose likelihood given the image data is maximal. Let $\mathbf{T} = \{T_{ij}\}$ denote the set of random variables such that T_{ij} stands for the true presence of dendrite skeleton on edge e_{ij}. We write

$$\mathbf{t}^* = \operatorname*{argmax}_{\mathbf{t}} P(\mathbf{T} = \mathbf{t}|I) , \quad (2)$$
$$= \operatorname*{argmax}_{\mathbf{t}} P(I|\mathbf{T} = \mathbf{t})P(\mathbf{T} = \mathbf{t}) ,$$

where we used Bayes' rule to go from the first line to the second and eliminated the constant term $P(I)$. In this work, we assume that all trees have the same prior probability and that subgraphs that are not trees have probability zero. Let $\mathcal{T}(G)$ be the set of all trees in G, and let t_{ij} denote the indicator variable representing the occupancy state of edge e_{ij} by the tree $\mathbf{t}$, we have

$$\mathbf{t}^* = \operatorname*{argmax}_{\mathbf{t} \in \mathcal{T}(G)} P(I|\mathbf{T} = \mathbf{t}) ,$$

$$= \operatorname*{argmax}_{\mathbf{t} \in \mathcal{T}(G)} \prod_{e_{ij} \in G} P(I_{ij}|T_{ij} = t_{ij}) , \quad (3)$$

$$= \operatorname*{argmax}_{\mathbf{t} \in \mathcal{T}(G)} \prod_{e_{ij} \in G} \frac{P(T_{ij} = t_{ij}|I_{ij})P(I_{ij})}{P(T_{ij} = t_{ij})} , \quad (4)$$

$$= \operatorname*{argmax}_{\mathbf{t} \in \mathcal{T}(G)} \prod_{e_{ij} \in G} P(T_{ij} = t_{ij}|I_{ij}) , \quad (5)$$

$$= \operatorname*{argmax}_{\mathbf{t} \in \mathcal{T}(G)} \prod_{e_{ij} \in G} P(T_{ij} = 1|I_{ij})^{t_{ij}}$$
$$\times P(T_{ij} = 0|I_{ij})^{(1-t_{ij})} , \quad (6)$$

$$= \operatorname*{argmin}_{\mathbf{t} \in \mathcal{T}(G)} \sum_{e_{ij} \in G} \left[-\log \frac{P(T_{ij} = 1|I_{ij})}{P(T_{ij} = 0|I_{ij})} \right] t_{ij}$$
$$+ \sum_{e_{ij} \in G} -\log P(T_{ij} = 0|I_{ij}) , \quad (7)$$

$$= \operatorname*{argmin}_{\mathbf{t} \in \mathcal{T}(G)} \sum_{e_{ij} \in G} \left[-\log \frac{P(T_{ij} = 1|I_{ij})}{P(T_{ij} = 0|I_{ij})} \right] t_{ij} , (8)$$

where I_{ij} stands for the image data around edge e_{ij}. Given that $\mathbf{t}$ is represented by a set of 0 and 1 values, Eq. 3 is justified by assuming that image evidence of dendriteness along neighboring edges is conditionally independent if we actually know whether these edges belong to dendrites or not. This amounts to describing the image in terms of a generative model where one first specifies where filaments are and then generate the appearance of every image location given that there is a filament or not. In Eqs. 4 and 5, we

use Bayes' rule again, drop the constant terms, and assume that edges are *a priori* equally likely to belong to a tree. The following equations are obtained by a simple algebraic manipulation in Eq. 6, taking the negative log in Eq. 7, and dropping a constant term in Eq. 8. In other words, since $P(T_{ij} = 1|I_{ij}) = p_{ij}$ and $P(T_{ij} = 0|I_{ij}) = 1 - p_{ij}$, the optimal tree $\mathbf{t}^*$ can be obtained by finding the set of indicator variables $\{t_{ij}^*\}$ such that the corresponding subgraph is a tree and minimizes

$$F(\mathbf{t}) = \sum_{i,j:t_{ij}=1} -\log \frac{p_{ij}}{1 - p_{ij}} \ . \tag{9}$$

In practice, this involves an immense solution space that cannot be easily searched, which is why we rely on the algorithm outlined at the beginning of the section to reduce it to a manageable size.

3.2. Sampling the Image Stack

We run the dendrite detector of [12] at different image scales to obtain a dendriteness measure for each voxel. It is obtained by convolving the image with high-order Gaussian derivatives, rotating the feature vector at each voxel to a reference orientation, and computing the response of an SVM on that feature vector.

The detections at different scales are then combined. Since we are applying the same detector $f(.)$ at different scales σ of the image I, no normalization is needed. For a voxel x_i the dendriteness measure is taken to be

$$\hat{f}(x_i) = \max_{\sigma} f(I_\sigma(x_i)). \tag{10}$$

This measure is then mapped to a posterior probability of being on the skeleton of a dendrite. We take the posterior probability of a pixel x_i to be

$$p_i = P(x_i \in \mathbf{t}^*|f(x_i)) = \frac{1}{1 + e^{-(af(x_i)+b)}}, \tag{11}$$

where the parameters a and b are estimated by cross-validation over a validation set. It has been shown in [20] that using a sigmoid to convert a SVM output into a probability is valid because it preserves the sparseness of the SVM while producing probabilities comparable to those produced by regularized likelihood kernel methods.

The original image is then sampled to retain only the voxels whose posterior probabilities given their dendritness measures are local maxima. The set of such samples is

$$V' = \{a_i = x_i | \forall x_j \in N_{x_i}, \tag{12}$$
$$P(x_j \in \mathbf{t}^*|f(x_j)) \le P(x_i \in \mathbf{t}^*|f(x_i))\},$$

where N_{x_i} is defined as a neighborhood of x_i.

(a)   (b)
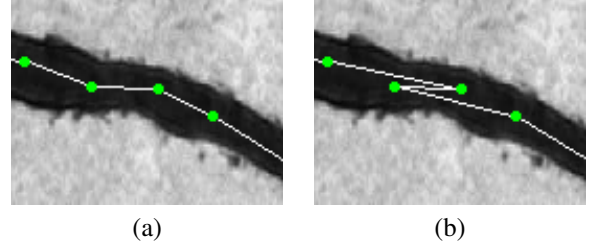
Figure 3. Close up views of delineations obtained using the edge cost assignments given in (a) Eq. 13 and (b) Eq 18. Green blobs represent the anchor points and white lines represent the links between them. Using the costs of Eq. 18 results in overextended paths and double counting of pixels.

3.3. Linking the Samples

We create a reduced graph $G' = (V', E')$ that links all pairs of $a_m \in V'$ and $a_n \in V'$ samples that are within a specified distance of each other by the path $l_{mn} \in E'$ that minimizes

$$c_{mn} = \sum_{e_{ij} \in l_{mn}} -\log p_{ij} \ , \tag{13}$$

which is the highest probability path between a_m and a_n. As mentioned in Section 3.1, p_{ij} are the probabilities that individual edges between neighboring voxels belong to the dendrite skeleton. Note that, in this case, we minimize a sum of negative log likelihoods instead of minimizing the sum of log likelihood ratios, as in Eq. 9. This is because we are not attempting to label all possible dendritic edges. Instead, we are only looking for a single path and are not concerned by the fact that there might be other high-probability edges in the vicinity that do not belong to that path and may be explained by another one.

We find this path by minimizing c_{mn} using Dijkstra's algorithm. In essence, we assume that all paths have the same prior and find the one with the highest probability given the local image evidence.

To derive the p_{ij} values from the p_i values of Eq. 11, let us consider the c_{mn} cost of Eq. 13. It is the sum of negative log likelihoods over the path joining m to n. To enforce consistency between the p_i and p_{ij} values, we require that c_{mn} should be roughly equal to the integral of the $-\log(p_i)$ along the corresponding continuous path. In other words, we should have

$$c_{mn} \approx \int -\log p(s)ds \tag{14}$$

$$\approx \sum_{e_{ij} \in l_{mn}} \int_0^{d_{ij}} -\log p(\frac{d_{ij} - s}{d_{ij}} x_i + \frac{s}{d_{ij}} x_j)ds \ ,$$

where s represents the curvilinear abscissa along the path, $p(s)$ the probability that the point at abscissa s belongs to the dendritic skeleton, and d_{ij} the distance between neighboring points x_i and x_j. Since we work on a voxel grid, to compute the integral of Eq. 15, we only have values p_i and

p_j of $p((1-s/d_{ij})x_i+(s/d_{ij})x_j)$ for $s=0$ and $s=d_{ij}$ respectively. Assuming that p varies linearly between x_i and x_j, we write:

$$c_{mn} \approx \sum_{e_{ij} \in l_{mn}} \int_0^{d_{ij}} -\log(\frac{d_{ij}-s}{d_{ij}}p_i + \frac{s}{d_{ij}}p_j)ds,$$

$$\approx \sum_{e_{ij} \in l_{mn}} -d_{ij}\frac{p_i(\log(p_i)-1)+p_j(1-\log(p_j))}{p_i-p_j}. \quad (15)$$

In practice, to avoid divisions by zero, we therefore take p_{ij} to be equal to $p_i^{d_{ij}}$ if $|p_i - p_j| \le \epsilon$, and so that

$$\log(p_{ij}) = d_{ij}\frac{p_i(\log(p_i)-1)+p_j(1-\log(p_j))}{p_i-p_j} \quad (16)$$

otherwise. Note that this is consistent because when $p_j - p_i$ tends towards zero, $\log(p_{ij})$ defined in this manner tends towards $d_{ij}\log(p_i) = d_{ij}\log(p_j)$.

3.4. Selecting the Best Tree

Assuming that there are N nodes in G', for all $K \le N$, we run a K-MST algorithm based on tabu search optimization [3] to find the tree $\mathbf{t}_K$ that spans exactly K nodes and minimizes

$$\sum_{l_{mn} \in \mathbf{t}_K} \sum_{e_{ij} \in l_{mn}} -\log p_{ij} = \sum_{i,j:t_{ij}=1} -\log p_{ij}, \quad (17)$$

where $\{t_{ij}\}$ is the set of indicator variables describing $\mathbf{t}_K$.

Note that it could have seemed more natural to assign to the edges of G' the cost

$$c_{mn} = \sum_{e_{ij} \in l_{mn}} -\log\frac{p_{ij}}{1-p_{ij}} \quad (18)$$

so that the cost optimized by the K-MST algorithm would have directly been the one of Eq. 9. However, the K-MST algorithm is a greedy algorithm, which selects at each iteration an edge to increase the tree score. If the edges are weighted using the log-likelihood ratios of Eq. 18, the preferred edge is the one explaining the greatest amount of image, ignoring that there still are edges to be added. This biases the optimization towards long edges and produces undesirable effects, such as over-counting voxels and zigzagging paths, as shown in Fig. 3. The same argument is made very convincingly, albeit in a different context, in Section 5 of [7]. By contrast, using the log-likelihood of Eq. 13 as edge weights results in the highest density of probability tree being built. Since we assume that there exists only one tree per image, once the tree t_K is constructed we consider all remaining edges as background, and assign to the tree the score $F(\mathbf{t}_K) = \sum_{i,j:t_{ij}=1}[-\log\frac{p_{ij}}{1-p_{ij}}]$ of Eq. 9.

In summary, for each cardinality $K = 2 \ldots N$, we build the tree that minimizes Eq. 17, and assign to it the score of
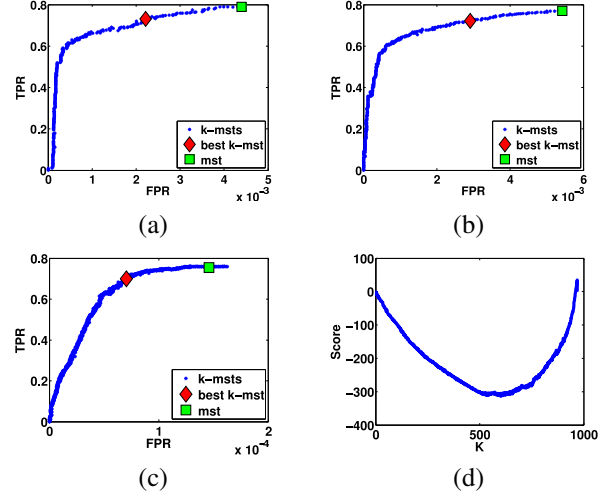

(a)   (b)

(c)   (d)

Figure 4. ROC curves obtained using the evaluation scheme of Fig. 5. Each tree is shown as a star. Since the K-MST trees are built by a randomized algorithm, the curve is not smooth. (a) First retinal scan of Fig. 7. (b) Second retinal scan of Fig. 7. (c) ROC for dendrites. The True Positive Rate of the K-MST that optimizes Eq. 19 is similar to that of the MST. The False Positive Rate, however, is reduced by a factor of two. (d) Cost function of Eq. 9 for different values of K.

Eq. 9. The final result is the tree that minimizes such score among all cardinalities,

$$\hat{\mathbf{t}}^* = \operatorname*{argmin}_{\mathbf{t_K} \in \{\mathbf{t}_2 \ldots \mathbf{t}_N\}} F(\mathbf{t_K}), \quad (19)$$

which yields a result such as the one of Fig. 2(f).

4. Results

We evaluate the performance of our algorithm in eight images of neurons obtained from brightfield microscopy, such as those depicted in Figs. 2 and 6, and on the DRIVE database of retinal scans depicted in Fig. 7. Using the available ground truth data, we compare our algorithm against a minimum spanning tree based approach. An MST spans a tree connecting all vertices of a graph such that the sum of the edge costs is minimized. This includes some anchor points that correspond to false-positives of the dendriteness measure. Furthermore, there is no guarantee that the optimal solution is a subset of such a tree. While pruning approaches [9, 13] can potentially remove false positives, they can never recover from false negatives present in the initial MST.

From these sets of experiments, we derive the ROC curves of Fig. 4 and draw the following conclusions:

1. As can be seen in Figs.6 and 7, our algorithm is able to connect thin and disconnected filaments while removing many false positives.
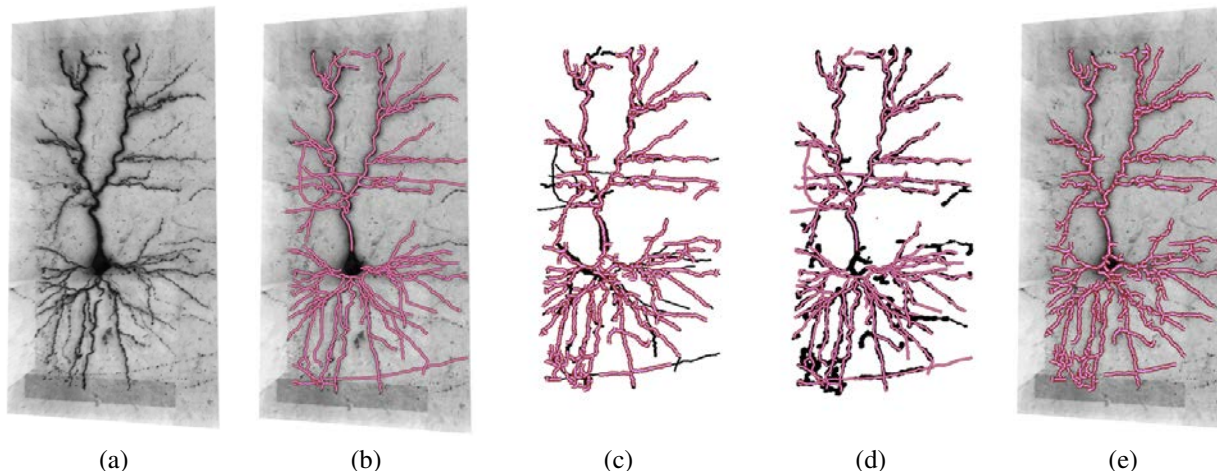
Figure 5. Evaluation procedure. (a) Original image. (b) Ground truth skeleton overlaid on top of the original image. (c) Estimated skeleton overlaid on top of the rasterization of the ground truth skeleton. Those pixels in the estimated skeleton outside of the rasterization are false positives. (d) Ground truth skeleton overlaid on top of the rasterization of the estimated skeleton. The ground truth voxels that lie outside of the black envelope are false negatives. (e) Skeleton recovered overlaid on top of the original image.

2. The stopping criterion on the K-MSTs represents a good compromise between the True Positive Rate (TPR) and the False Positive Rates (FPR). In all experiments the TPR is reduced by at most 10%, while the FPR is reduced by, at least, a factor of two with respect to the MST approach.

In the remainder of this section, we first discuss our evaluation methodology in more detail. We then focus on the two datasets where we evaluate our method.

4.1. Experimental methodology

Ground truth data sets were collected for both data types by experts. The annotations denote the skeleton of the network of filaments on the images. From those annotations, samples inside and outside the filaments are obtained to train the dendrite measure. For each data type we collected a minimum of two fully annotated images. The annotated data was divided into disjoint training, validation and test sets, leaving at least one whole image for testing.

After detection, anchor points are selected by taking the neighborhood of Eq. 12 to be a sphere of radius 20. This produces between 500 to 2000 anchor points per image.

The K-MST algorithm is then run for values ranging from two to the number of anchor points. The optimal k value is obtained using Eq. 19.

The evaluation process is depicted in Fig. 5. Since the ground truth contains width estimates, we render an image stack by assigning a one to all points that are within the corresponding distance from the dendritic spine and zero to the others. We then consider as true positives all vertices of our optimal tree that have received the one label. As shown in Fig. 5(d), to compute the false negative rate, we reverse

the roles of the automatically reconstructed tree and of the ground truth one. We use the former to assign labels using the width associated to the dendriteness measure of Eq. 10 and count as false negatives all vertices of the latter that receive a zero label.

4.2. Dendrites in Brightfield Image Stacks

Images of dendritic trees, shown in Figs. 1, 2, 5 and 6, were obtained from brain sections of rats. The neuron is dyed and then imaged with a brightfield microscope at different tissue depths. Each image stack has an approximate dimensions of $1.5e^3 \times 2e^3 \times 1e^2$ pixels.

Several artifacts appear in these images due to irregularities of the staining process and non-gaussian blur attributed to the image acquisition technique. As a result, many interesting filaments appear as faint structures, present abrupt intensity changes and are severely blurred. Furthermore, the stain can dye irrelevant structures, such as blood vessels close to the neuron under analysis. These structures result in structured and unstructured noise that is difficult to separate from the observations even for a human expert.

As shown in Fig. 2(c), many false anchor points are created in the non-maxima suppression step, which yields over-complete minimum spanning trees, such as the one of Fig. 2(c). By applying our *K-MST* based optimization method we can produce the tree of Fig. 2(f). Such trees halve the False Positive Rates of the MST while reducing the True Positive Rate by $5-10\%$, as shown in the ROC curve of Fig. 4(a).

4.3. Blood vessels in retinal images

Blood vessel images obtained from the DRIVE data set [22] are cleaner than the dendrite images described previ-
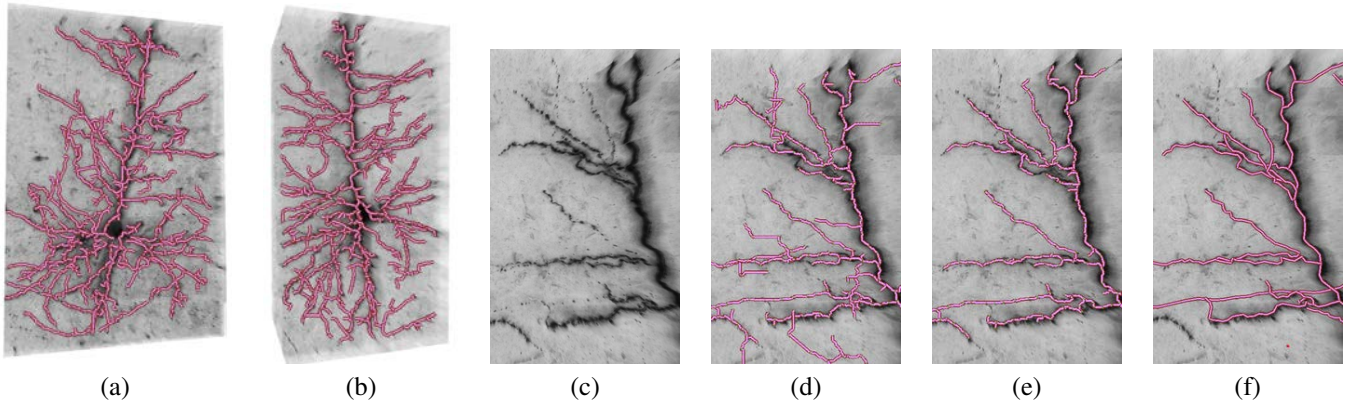
Figure 6. (a),(b) Result of the delineation on two 3D brightfield image stacks. One of the main sources of error is around the soma of the neurons. This is due to the fact that we do not model the topology of such regions and assume that there are only elongated structures in the image. (c) A close-up of an image stack. (d) MST found by processing (c). (e) The skeleton estimated by the proposed algorithm. (f) Ground truth skeleton.
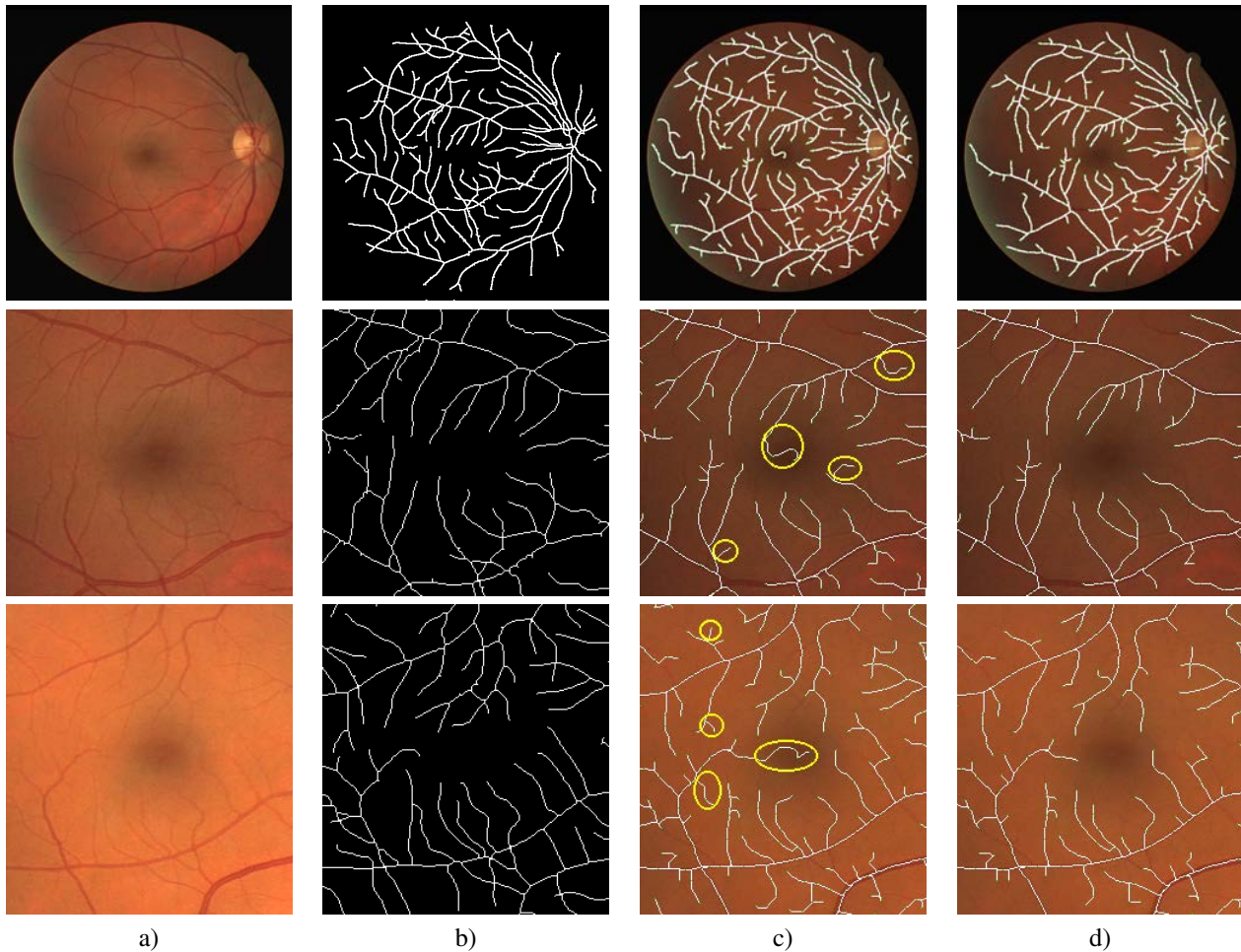

Figure 7. Result of the blood vessel delineation on two retinal images of the DRIVE data set. The second row is close-up of the image of the first row. The third is a close-up of a different image. a) Original Images b) Skeletons of the manual segmentation, which is taken as ground truth. c) MSTs spanning all anchor points d) Optimal K-MSTs spanning a subset of the anchor points. Note the differences highlighted by the yellow circles.

ously. The background is mostly uniform and the vessels appear as dark structures. However, the optic disk region is a localized source of noise, which causes problems to both our classifier and to the algorithms of [15, 10] as can be seen in Fig. 7(a).

The data set contains 20 test images with manual segmentations of blood vessels performed by trained human observers. We applied a simple erosion-based skeletonization algorithm to these segmentations to obtain centerlines of the vessels, and used these skeletons as ground truth in the objective evaluation.

Fig. 7 illustrates two sample results of our algorithm on this data set. While simply spanning all anchor points yields many false positives highlighted by the yellow ellipses in Fig. 7(c) which are hard to prune at a later post processing step, our K-MST based approach eliminates many outliers through its global optimization procedure. This observation is also supported objectively by the ROC curves given in Fig. 4(b),(c), where MST yields a false positive rate almost twice as high as that of our algorithm with an increase of only 7% in the true positive rate.

It is important to note that although the 3D brightfield image stacks occasionally contain very few cases of branch crossings, 2D retinal images are full of such crossings. Therefore, there are many loops in the images, which violates the tree topology model we utilize. As a result, the algorithm removes the least cost edge from each loop, which produces a dropout in the TPR.

5. Conclusion

We have presented a method to automatically infer tree structures present on 2D and 3D images. In contrast to earlier methods that take decisions based only on local image evidence, our algorithm finds the most probable tree by optimizing a global cost function. This methodology allows us to recover faint and disconnected filaments while rejecting structured noise and is not subject to drifting artifacts that affect other methods.

In future work, we will use informative tree priors that encode topological information, such as width and orientation constraints, tortuosity and smoothness into the optimization. They fit perfectly in our framework and should further boost performance.

References

[1] K. Al-Kofahi, A. Can, S. Lasek, D. Szarowski, N. Dowell-Mesfin, W. Shain, J. N. Turner, and et al. Median-based robust algorithms for tracing neurons from noisy confocal microscope images, december 2003.

[2] F. Benmansour, L. D. Cohen, M. W. Law, and A. C. Chung. Tubular anisotropy for 2d vessels segmentation. In *CVPR*, 2009.

[3] C. Blum and M. J. Blesa. New metaheuristic approaches for the edge-weighted k-cardinality tree problem. pages 1355–1377, 2005.

[4] A. R. Cohen, B. Rosyam, and J. N. Turner. Automated tracing and volume measurements of neurons form 3-d confocal fluorescence microscopy data. *Journal of Microscoopy*, 173:103–114, February 1994.

[5] P. Doucette, P. Agouris, A. Stefanidis, and M. Musavi. Self-organised clustering for road extraction in classified imagery. *IJPRS*, 55:347–358, 2001.

[6] J. F. Evers, S. Schmitt, M. Sibila, and C. Duch. Progress in Functional Neuroanatomy: Precise Automatic Geometric Reconstruction of Neuronal Morphology From Confocal Image Stacks. *J Neurophysiol*, 93(4):2331–2342, 2005.

[7] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, 2006.

[8] M. Fischler and A. Heller. Automated Techniques for Road Network Modeling. In *DARPA IUW*, Monterey, CA, 1998. Morgan Kaufmann.

[9] M. Fischler and H. Wolf. Linear Delineation. In *CVPR*, pages 351–356, June 1983.

[10] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496:130–137, 1998.

[11] N. Garg. A 3-approximation for the minimum tree spanning k vertices. In *IEEE Symposium on Foundations of Computer Science*, pages 302–309, 1996.

[12] G. Gonzalez, F. Aguet, F. Fleuret, M. Unser, and P. Fua. Steerable Features for Statistical 3D Dendrite Detection. In *MICCAI*, London, UK, September 2009.

[13] G. Gonzalez, F. Fleuret, and P. Fua. Automated Delineation of Dendritic Networks in Noisy Image Stacks. In *ECCV*, Marseille, France, October 2008.

[14] G. Gonzalez, F. Fleuret, and P. Fua. Learning Rotational Features for Filament Detection. In *CVPR*, Miami, FL, June 2009.

[15] M. Jacob and M. Unser. Design of steerable filters for feature detection using Canny-like criteria. *PAMI*, 26(8):1007–1019, August 2004.

[16] M. Law and A. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *ECCV*, pages 368–382, 2008.

[17] C. McIntosh and G. Hamarneh. Vessel crawlers: 3d physically-based deformable organisms for vasculature segmentation and analysis. In *CVPR*, pages 1084–1091, Washington, DC, USA, 2006.

[18] 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.

[19] E. Mortensen and W. Barrett. Intelligent Scissors for Image Composition. In *SIGGRAPH*, pages 191–198, Los Angeles, CA, August 1995.

[20] J. Platt. *Advances in Large Margin Classifiers*, chapter Probabilistic Outputs for SVM and Comparisons to Regularized Likelihood Methods. MIT Press, 2000.

[21] A. Santamaría-Pang, C. M. Colbert, P. Saggau, and I. Kakadiaris. Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging. In *MICCAI*, pages 486–494, 2007.

[22] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23:501–509, 2004.

[23] A. Szymczak, A. Stillman, A. Tannenbaum, and K. Mischaikow. Coronary vessel trees from 3d imagery: a topological approach. *Medical Image Analisys*, 08 2006.

[24] C. M. Weaver, P. R. Hof, S. L. Wearne, and W. B. Lindquist. Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural Computation*, 16(7), 2004.