

Automated Reconstruction of Tree Structures using Path Classifiers and Mixed Integer Programming

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

Although tracing linear structures in 2D images and 3D image stacks has received much attention over the years, full automation remains elusive. In this paper, we formulate the delineation problem as one of solving a Quadratic Mixed Integer Program (Q-MIP) in a graph of potential paths, which can be done optimally up to a very small tolerance. We further propose a novel approach to weighting these paths, which results in a Q-MIP solution that accurately matches the ground truth.

We demonstrate that our approach outperforms a state-of-the-art technique based on the k -Minimum Spanning Tree formulation on a 2D dataset of aerial images and a 3D dataset of confocal microscopy stacks.

1. Introduction

Fully automated reconstruction of tubular tree-like structures such as the neuronal arbors in optical microscopy image stacks, blood vessels in retinal scans, or road networks in aerial images remains an open computer vision problem. Current techniques still lack robustness to imaging artifacts such as noise, inhomogeneous contrasts, non-uniform illumination, and scene clutter. As a result, practical systems require extensive manual intervention. For example, in the recently completed DIADEM challenge [2], the algorithms that proved best at tracing dendritic trees were also those that provided the best tools for manual editing. In neuroscience research, the requirement for such editing dramatically slows down the process and makes it impossible to exploit the vast amount of data that modern microscopes can produce.

Part of the problem comes from the fact that techniques that rely on local criteria to create the trees are generally greedy in nature and lack robustness to large gaps in the image data while techniques that involve minimizing a global objective function usually do so using heuristics that can get

trapped in undesirable local minima.

To overcome these limitations, we introduce a novel framework, which involves building a tree that is provably very close to the global optimum of a Quadratic Mixed Integer Program (Q-MIP). This is the first contribution of this paper, which is made practical by the second one, an original classification-based approach to assessing the probability that a tubular path corresponds to a real curvilinear structure. The resulting probability estimates are reliable enough so that the corresponding log-likelihood ratios can be directly injected into the Q-MIP and that its solution corresponds to the desired response. This is in contrast to more traditional approaches to scoring paths by integrating pixel values along their length, which often fails to adequately penalize short-cuts and makes it difficult to compute commensurate scores for paths of different lengths.

We evaluated both the path classification and the tree reconstruction components of our approach on a 3D dataset of confocal micrographs containing neurites and a 2D dataset of aerial images containing road networks. We compared our results against manually annotated tubular delineations in both cases and achieved high classification and reconstruction accuracies on the evaluated datasets.

2. Related Work

The analysis and reconstruction of tree-like structures [15, 7, 19] has recently received much attention. Most automated techniques rely on a local *tubularity* measure that returns the likelihood of a point being on the centerline of a tubular structure. Examples include matched filter scores [1, 25], Hessian and Oriented Flux functionals [21, 12, 16], and classification scores derived from steerable filter responses [14, 13]. They are used to find the most likely paths using a search mechanism that can be either local or global.

Local search methods involve algorithms, which, starting from a set of seed points, incrementally grow branches by iteratively adding new points and paths [1, 3, 23]. These algorithms are computationally efficient since the tubularity

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

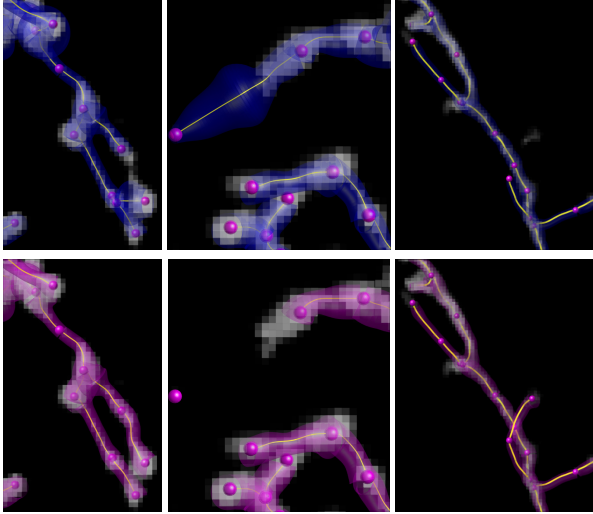


Figure 1: Path classification vs Integration in portion of microscopy stacks from the DIADEM data [2]. (Top) Scoring paths by summing tubularity values [22] results in, from left to right, shortcuts, spurious branches, and missing branches. (Bottom) Our classification approach to scoring paths yields the right answer in all three cases.

measure needs only be evaluated on a small subset of the image points in the vicinity of the seeds. However, due to their greedy nature, they lack robustness to imaging noise, especially when there are large gaps along the filaments. Furthermore, they typically require separate procedures to detect branching points.

Global search methods aim at achieving greater robustness by computing the tubularity measure everywhere. They then extract paths from the resulting tubularity image by either thresholding and thinning it [5] or extracting local maximum seed points and connecting nearby points by high-tubularity paths [22]. These paths are then represented as a weighted graph and the tree-reconstruction problem is formulated as one of finding the minimum-weight tree that resides within this graph. Algorithms that find a Minimum Spanning Tree (MST) [11, 25, 23] or a Shortest Path Tree (SPT) [20] belong to this class. They can handle gaps in the image data but can easily produce spurious branches when seed points that are not part of the tree structure are mistakenly detected, which happens all too often in noisy data. As a result, a post-processing step is usually required and is not guaranteed to correct all such mistakes.

The recent k -Minimum Spanning Tree (k -MST) formulation addresses this issue by posing the problem as one of finding the minimum cost tree that spans only an *a priori* unknown subset of k seed points [22]. However, the method relies on a heuristic search algorithm and a dual objective function, one for searching and the other for scoring, without guaranteeing the global optimality of the final reconstruction. Furthermore, it requires an explicit optimization

over the auxiliary variable k , which is not relevant to the problem. By contrast, the Mixed Integer Program formulation we advocate in this paper involves minimization of a single global objective function that allows us to link legitimate seed points while rejecting spurious ones by finding the optimum solution to within a small user-specified tolerance.

The key ingredient that makes this approach practical is our ability to assign paths connecting seed points probabilities that faithfully represent their true likelihood of belonging to the real tree structure. As a result, the global optimum we find fits better to the desired tree in the image, with fewer missing or spurious branches. To this end, we develop novel appearance features based on gradient histograms to perform path classification on the basis of training data. This is in contrast to all other approaches currently used by both local and global search methods to score the paths they construct. For example, global methods that rely on geodesic distances express this cost as an integral of a function of the tubularity values [22, 18]. Similarly, active contour-based methods typically define their energy terms as such integrals over the paths [23, 17]. Since the tubularity values only depend on local image evidence, they are not particularly effective at ignoring paths that are mostly on the curvilinear structures but also partially on the background. Moreover, because the scores are computed as sums of values along the path, normalizing them so that paths of different lengths can be appropriately compared is non-trivial. By contrast, we will show that our approach returns comparable probabilistic costs for tubular paths of arbitrary length. Furthermore, our path features capture global appearance, while being robust to noise and inhomogeneities.

3. Approach

We first briefly outline our reconstruction algorithm, which goes through the following steps depicted by Fig. 2:

1. We compute a *tubularity* value at each image location x_i and radius value r_i , which quantifies the likelihood that there exists a tubular structure of radius r_i , at location x_i . Given an N -D image, this creates an $(N + 1)$ -D scale-space tubularity image such as the one shown in Fig. 2(b).
2. We select high-probability points in this tubularity image as seed points and connect them through high-tubularity paths in scale-space. This results in a directed graph (see Fig. 2(c)).
3. Having trained a path classifier using such graphs and ground-truth trees, we assign log-likelihood ratio weights to pairs of consecutive edges of a given graph at detection time (see Fig. 2(d)).
4. We use these weights and solve a Mixed Integer Program to compute the maximum-likelihood directed tree in this graph (see Fig. 2(e)).

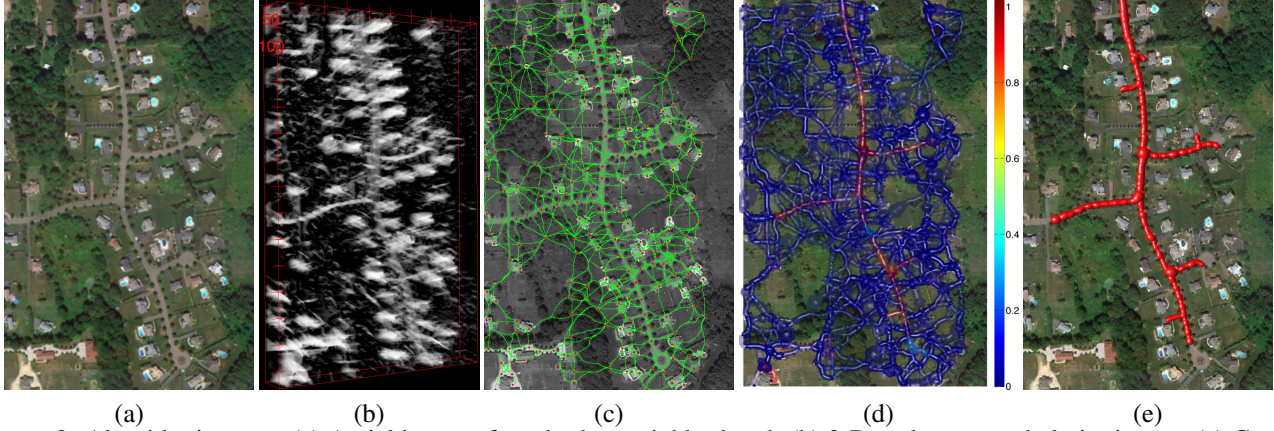


Figure 2: Algorithmic steps. (a) Aerial image of a suburban neighborhood. (b) 3-D scale-space tubularity image. (c) Graph obtained by linking the seed points. They are shown in red with the path centerlines overlaid in green. (d) The same graph with probabilities assigned to paths using our path classification approach. Blue and transparent denote low probabilities, red and opaque high ones. Note that only the paths lying on roads appear in red. (e) Final reconstruction obtained by solving the Q-MIP problem.

These four steps come in roughly the same sequence as those used in most algorithms that build trees from seed points, such as [11, 25, 23, 22], but with two key differences. First, whereas heuristic optimization algorithms such as MST followed by pruning or the k-MST algorithm offer no guarantee of optimality, our approach guarantees that the solution is within a small tolerance of the global optimum. Second, our approach to scoring individual paths using a classifier instead of integrating pixel values as usually done gives us more robustness to image noise and helps ensure that the global optimum is close to the ground truth.

4. Q-MIP Formulation

In this section, we first discuss the construction of the graph of Fig. 2(c), which is designed to be an over-complete representation for the underlying network of tubular structures. We then show how finding the most likely *arborescence*¹ can be formulated as a Q-MIP problem.

4.1. Graph Construction

Building graphs such as the one depicted by Fig. 2(c) is done in three steps.

First, we compute a scale space tubularity measure based on the oriented flux cross-section trace measure [16]. This measure is used to assess if a voxel lies on a centerline of a filament at a given scale.

Second, we sample seed points from the tubularity image by iteratively selecting the maximum tubularity points and then suppressing their neighborhoods. Finally, we compute paths linking the seed points using a variant of the minimal path method applied in the scale space domain [18]. Therefore, a *geodesic tubular path* connecting seed points v_i and

¹An arborescence is a directed tree with a distinguished vertex, called the root, with in-degree zero and such that there is a unique directed path from it to every other vertex in the tree.

v_j is taken to be

$$p_{ij} = \operatorname{argmin}_{\gamma} \int_0^L \mathcal{P}(\gamma(s)) ds, \quad (1)$$

where \mathcal{P} is an exponential mapping of the tubularity measure, $s \in [0, L]$ is the arc-length parameter and γ is a parametrized curve mapping s to a location in \mathbb{R}^{N+1} [4]. The first N dimensions are spatial ones while the last one denotes the scale.

4.2. Standard Formulation

Formally, the procedure described above yields a graph $G = (V, E)$, whose vertices $V = \{v_i\}$ represent the seed points and directed edges $E = \{e_{ij} = (v_i, v_j)\}$ represent geodesic tubular paths linking them. Algorithms [11, 25, 23, 22] that rely on this kind of formalism can all be understood as maximizing an *a posteriori* probability given a tubularity image and optionally a set of meta parameters that encode geometric relations between vertices and edge pairs. For example, in the recently published [22], building the tree is shown to be equivalent to solving

$$\min_{\mathbf{t} \in \mathcal{T}(G)} \left\{ \sum_{e_{ij} \in E} c_{ij}^a t_{ij} + \sum_{e_{ij}, e_{jk} \in E} c_{ijk}^g t_{ij} t_{jk} \right\}, \quad (2)$$

where $\mathcal{T}(G)$ denotes the set of all arborescences in G , t_{ij} is a binary variable indicating the presence or absence of e_{ij} in arborescence \mathbf{t} . c_{ij}^a represents the cost of an edge, which can be either negative or positive and is computed by integrating pixelwise negative log-likelihood ratio values along the path connecting the vertices, while c_{ijk}^g encodes the geometric compatibility of consecutive edges. As shown in Fig. 3, these geometric terms are important to eliminate edge sequences that backtrack or curve unnaturally. This

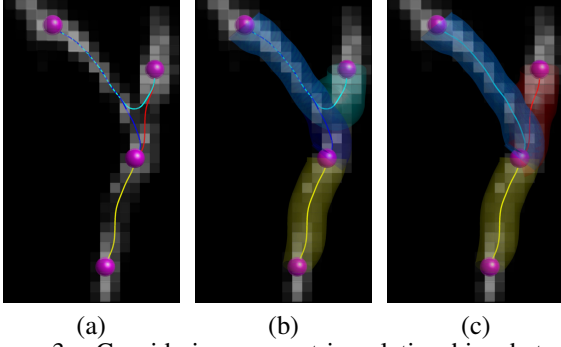


Figure 3: Considering geometric relationships between edges helps at junctions. (a) A closeup view of the graph built by our algorithm at a branching point. (b) Minimizing a sum of individual path costs yields these overlapping paths. (c) Accounting for edge-pair geometry yields the correct connectivity.

approach, however, has two severe shortcomings. First, because the c_{ij}^a and c_{ijk}^g are computed independently, they are not necessarily commensurate or consistent with each other. As a consequence, careful weighting of the two terms is required for optimal performance. Second, optimizing using a heuristic algorithm [8] does not guarantee a global optimum.

We address the first issue by computing probability estimates, not on single edges, but on edge pairs so that both appearance and geometry can be accounted for simultaneously, as will be discussed in Section 5. Given such estimates, we solve the second problem by reformulating the tree reconstruction problem not as the heuristic minimization of an energy such as the one of Eq. 2 but as a solution of a Q-MIP problem, for which we can find exact solutions.

4.3. Integer Program in Terms of Edge Pairs

Let $F = \{e_{ijk} = (e_{ij}, e_{jk})\}$ be the set of pairs of consecutive edges in G and $S = \{p_{ijk}\}$ be the paths corresponding to these pairs. By analogy to the binary variable t_{ij} of Eq. 2, let t_{ijk} denote the presence or absence of e_{ijk} in the arborescence. Let T_{ijk} be the corresponding hidden variable denoting whether p_{ijk} truly corresponds to a tubular structure in the image. Let also \mathbf{t} and \mathbf{T} be the set of all t_{ijk} and T_{ijk} variables respectively.

Using Bayes' rule and assuming a uniform distribution over the image I and the paths S , the optimal arborescence $\mathbf{t}^* \in \mathcal{T}(G)$ maximizes

$$P(\mathbf{T} = \mathbf{t}|I, S) \propto P(I, S|\mathbf{T} = \mathbf{t})P(\mathbf{T} = \mathbf{t}) . \quad (3)$$

Since we encode geometric relationships in the pairwise edge terms, we use a uniform prior for the arborescences and drop the prior term $P(\mathbf{T} = \mathbf{t})$ from Eq. 3. Furthermore, as hinted above, we assume conditional independence of image evidence along the tubular paths $\{p_{ijk}\}$, given that we know whether their edge pairs $\{e_{ijk}\}$ belong to the

tree structure. We therefore represent the likelihood term $P(I, S|\mathbf{T} = \mathbf{t})$ as a product of individual edge pair likelihoods. Following similar steps as in [22], this leads to

$$\begin{aligned} \mathbf{t}^* &= \underset{\mathbf{t} \in \mathcal{T}(G)}{\operatorname{argmin}} \sum_{e_{ijk} \in F} -\log \left(\frac{P(T_{ijk} = 1|I_{ijk}, p_{ijk})}{P(T_{ijk} = 0|I_{ijk}, p_{ijk})} \right) t_{ijk} \quad (4) \\ &= \underset{\mathbf{t} \in \mathcal{T}(G)}{\operatorname{argmin}} \sum_{e_{ijk} \in F} c_{ijk} t_{ijk}, \quad (5) \end{aligned}$$

where I_{ijk} represents image data around the path p_{ijk} . The probability $P(T_{ijk} = 1|I_{ijk}, p_{ijk})$ denotes the likelihood of the path p_{ijk} belonging to the tree structure, which we compute based on global appearance and geometry of the paths as described in Section 5. The c_{ijk} variables represent the probabilistic likelihood ratios we assign to the edge-pairs. As we will show next, optimizing the objective function of Eq. 5 with respect to the constraints $\mathbf{t} \in \mathcal{T}(G)$ amounts to solving a minimum arborescence problem [9] with a quadratic cost.

By decomposing the indicator variable t_{ijk} introduced above as the product of the two variables t_{ij} and t_{jk} , we can express the cost of Eq. 5 as a quadratic function of these variables. The constraints $\mathbf{t} \in \mathcal{T}(G)$ can also be defined in terms of these variables by adapting the network flow formulation presented in [9], which provides a compact system with a polynomial number of variables and constraints. Assuming that the root vertex v_r of the optimal arborescence is given, the minimization of Eq. 5 can then be reformulated as the Q-MIP

$$\begin{aligned} &\underset{\mathbf{t} \in \mathcal{T}(G)}{\operatorname{argmin}} \sum_{e_{ij}, e_{jk} \in E} c_{ijk} t_{ij} t_{jk} \quad (6) \\ \text{s.t. } &\sum_{v_j \in V \setminus \{v_r\}} y_{rj}^l \leq 1, \quad \forall v_l \in V \setminus \{v_r\}, \\ &\sum_{v_j \in V \setminus \{v_k\}} y_{jk}^l \leq 1, \quad \forall v_l \in V \setminus \{v_r\}, \\ &\sum_{v_j \in V \setminus \{v_i, v_r\}} y_{ij}^l - \sum_{v_j \in V \setminus \{v_i, v_l\}} y_{ji}^l = 0, \quad \forall v_l \in V \setminus \{v_r\}, \\ &\quad \quad \quad \forall v_i \in V \setminus \{v_r, v_l\}, \\ &y_{ij}^l \leq t_{ij}, \quad \forall e_{ij} \in E, v_l \in V \setminus \{v_r, v_i, v_j\}, \\ &y_{il}^l = t_{il}, \quad \forall e_{il} \in E, \\ &y_{ij}^l \geq 0, \quad \forall e_{ij} \in E, v_l \in V \setminus \{v_r, v_i\}, \\ &t_{ij} \in \{0, 1\}, \quad \forall e_{ij} \in E, \end{aligned}$$

where the y_{ij}^l are auxiliary continuous variables that denote the flow from the root vertex to all others. More specifically, y_{ij}^l indicates whether the unique directed path from the root v_r to vertex v_l traverses the edge e_{ij} . If the optimal arborescence \mathbf{t}^* does not contain v_l and hence such a path does not exist, then $y_{ij}^l = 0$. The first two constraints ensure that

there can be at most one path in \mathbf{t}^* from the root to each vertex in the graph. The third one enforces conservation of flow at intermediate vertices v_l . The remaining constraints guarantee that \mathbf{t}^* includes a path from the root to the vertex v_l passing through edge e_{il} if \mathbf{t}^* contains e_{il} .

Even though this Q-MIP problem is NP-Hard [9], its solution can be found up to an arbitrarily small tolerance² from the true optimum using a branch-and-cut strategy³. As a result, this optimization took only a few minutes on a dual core PC for the examples presented in the result section.

5. Path Classification

The outcome of the Q-MIP procedure introduced in the previous section depends critically on the probabilistic weights c_{ijk} of Eq. 5, which are assigned to edge pairs.

A standard approach to computing such weights is to integrate tubularity values along the paths, as in Eq. 1. However, as shown in Fig. 4, the resulting estimates are often unreliable because a few very high values along the path might offset low values and, as a result, fail to adequately penalize spurious branches and short-cuts. Furthermore, it is often difficult to find an adequate balance between allowing paths to deviate from a straight line and preventing them from meandering too much.

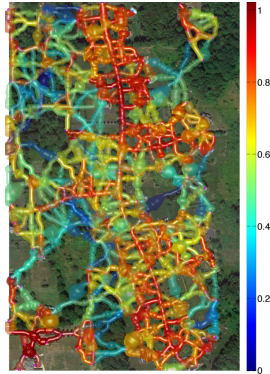


Figure 4: Tubular graph of Fig. 2(c) with edge weights computed by integrating tubularity values along the paths instead of using our path classification approach. We use the same color scheme as in Fig. 2(d) to demonstrate how much less informative these weights are.

In this section, we propose a path-classification approach to computing the probability estimates that we found to be more reliable. More specifically, given a tubular path computed as discussed in Section 4.1, we break it down into several segments and compute one feature vector based on gradient histograms for each. We then use an embedding approach [24] to compute fixed-size descriptors from the potentially arbitrary number of feature vectors we obtain. Finally, we feed these to an SVM classifier and turn its output into a probability estimate.

As shown in Fig. 1, this approach penalizes paths that mostly follow the true tree structure but cross the background. Thus, it discourages shortcuts, which is something that integrating along the path fails to do.

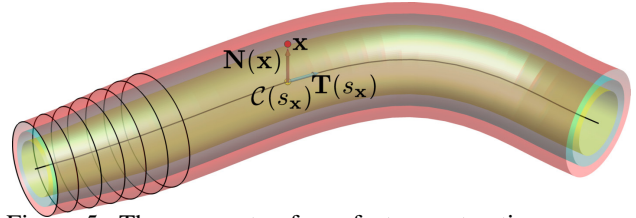


Figure 5: Three aspects of our feature extraction process. An extended neighborhood of points around the path centerline $\mathcal{C}(s)$ is defined as the envelope of cross-sectional circles shown in black. This neighborhood is divided into R radius intervals highlighted by the yellow, green and red tubes (here $R = 3$) and a histogram is created for each such interval. A point \mathbf{x} contributes a weighted vote to an angular bin according to the angle between the normal $\mathbf{N}(\mathbf{x})$ and the image gradient $\nabla I(\mathbf{x})$ at that point.

In the remainder of this section, we describe our path features, embedding scheme, and training data collection mechanism in more details.

5.1. Histogram of Gradient Deviation Descriptors

Gradient orientation histograms have been successfully applied to detecting objects in images and recognizing actions in videos [6, 10, 24]. In a typical setup, the image is first divided into a grid of fixed-size blocks, called cells, and then for each cell, a 1-D histogram of orientated gradients (HOG) is formed from the pixel votes within it. Histograms from neighboring cells are then combined and normalized to form feature vectors invariant to illumination and contrast changes. Finally, these features are fed into a classifier to detect objects of interest. We adapt this strategy by defining *Histogram of Gradient Deviation* (HGD) descriptors as follows.

Given a tubular path $\gamma(s)$ such as the one depicted by Fig. 5, with s being the curvilinear abscissa, let $\mathcal{C}(s)$ be the centerline and $r(s)$ the corresponding radius mappings. We partition the path into equal-length overlapping segments and, for each, we compute histograms of gradient orientation deviations from the normal vectors emanating from the centerline. The histograms are populated by points belonging to a certain neighborhood $\mathcal{N}(\gamma)$ around the centerline of the path. This neighborhood is defined as the envelope of cross-sectional circles as illustrated by Fig 5. To ensure that all the gradient information surrounding the tube is captured, we extend this neighborhood by a margin $m(s) = K * r(s)$ proportional to the radii values.

For a given image point $\mathbf{x} \in \mathcal{N}(\gamma)$, let $\mathbf{N}(\mathbf{x})$ be the *normal ray vector* emanating from the centerline \mathcal{C} passing by \mathbf{x} , and $\mathcal{C}(s_x)$ the closest point to it. Each such point contributes a weighted vote $\|\nabla I(\mathbf{x})\|$ to a histogram bin, which we take to be

$$\Psi(\mathbf{x}) = \begin{cases} \text{angle}(\nabla I(\mathbf{x}), \mathbf{N}(\mathbf{x})), & \text{if } \|\mathbf{x} - \mathcal{C}(s_x)\| > \varepsilon \\ \text{angle}(\nabla I(\mathbf{x}), \mathbf{\Pi}(\mathbf{x})), & \text{otherwise,} \end{cases} \quad (7)$$

²We used an absolute MIP gap tolerance of $1e-4$ in all our experiments.

³We used the Gurobi optimization software for solving the Q-MIPs.

where $\Pi(\mathbf{x})$ is the cross-sectional plane, which we use to compute the deviation angle when \mathbf{x} belongs to the centerline and the normal ray vector is not defined.

To obtain a description of paths’ appearance on the cross-sectional plane, we further split the neighborhood $\mathcal{N}(\gamma)$ into R equally spaced radius intervals as shown in Fig. 5 and create a histogram for each such interval. Given B orientation bins, the radius interval and the angular bin indices for a point \mathbf{x} are then given by $\min(R - 1, \lfloor R\|\mathbf{N}(\mathbf{x})\|/(r(s_{\mathbf{x}}) + m(s_{\mathbf{x}})) \rfloor$ and $\min(B - 1, \lfloor B\Psi(\mathbf{x})/\pi \rfloor$ respectively. For each segment, this produces R histograms, each one corresponding to a radius interval. We interpolate points within each such interval to ensure that enough votes are used to form the histograms. Finally, we normalize each histogram by the number of points that voted for it.

This yields a set of histograms for each segment, which we combine into a single HGD descriptor.

5.2. Embedding

The above procedure produces an arbitrary number of HGD descriptors per path. To derive from them a fixed-size descriptor, we first use a Bag-of-Words (BoW) approach to compactly represent their feature space. The words of the BoW model are generated by randomly sampling a pre-defined number of descriptors from the training data. For a given path of arbitrary length, we then compute an embedding of the path’s HGD descriptors into the *codewords* of the model. Adapting the sequence embedding approach of [24], we find the minimum Euclidean distance from the path’s descriptors to each word in the model. This yields a feature vector of minimal distances that has the same length as the number of elements in the BoW model.

To account for geometry and characterize paths that share a common section, such as the one shown in Fig. 3(a), we incorporate into these descriptors the maximum curvature along the centerline curve \mathcal{C} . It is computed as $\arg\max\|\mathbf{T}'(s)\|$, where $\mathbf{T}(s)$ is the unit tangent vector.

5.3. Parameters

In all our experiments, we used the same parameters to compute our HGD descriptors: Segment length 2 pixels; segment sampling step 0.5 pixels implying a 75% overlap; 9 angular bins; two radius intervals; radius margin $K = 0.33$, and a randomly sampled BoW model of 300 codewords.

For classification purposes, we used an SVM classifier with an RBF kernel. During training, the C and the γ parameters were optimized by performing a grid search using a standard 5-fold cross validation procedure on the confocal microscopy images of Fig. 7. At run time, the resulting parameters are used for both the neurite and the road images.

5.4. Collecting Training Data

To train the SVM classifier, we obtain positive samples by simply sampling the ground-truth trees associated to our

training images. To obtain negative samples, we first build tubular graphs in these training images using the method of Section 4.1. We then randomly select paths from these graphs and attempt to find *matching* paths in the ground truth tree. For a given path, this is done by finding the two nodes of the tree that are closest to the start and end points of the path. It is considered as a true negative if the length of its largest centerline section that is outside the volume of the matched path is larger than a threshold, taken to be 4 pixels in all our experiments.

6. Results

We evaluated our approach both on 3D confocal microscopy image stacks of Olfactory Projection Fibers (OPF) of the *Drosophila* fly and on 2D aerial images of a suburban neighborhood. The OPF dataset consists of 9 images, which we split into a training and a validation set, leaving 3 images in the latter. Similarly, the Road dataset consists of 14 images, 7 of which are used for training, and 7 for testing. Sample images from both datasets and our results are depicted by Fig 7.

In both cases, we used a semi-automated delineation tool we developed to obtain the ground truth tracings⁴. Note that the OPF dataset was used in the DIADEM challenge [2] discussed in the introduction and that ground truth was available for it. However, we found it to be both incomplete and imprecise, in particular with respect to the width and precise centerline location of the paths, which is why we chose to use ours.

The DIADEM challenge involved many groups worldwide. Of these, five made it into the final round and the ones that scored highest relied on a various levels of manual intervention. We compare ourselves to the k-MST approach of [22], which was both one of the five and fully automated. To ensure that the comparison is fair, we extended the original implementation of [22] by taking into account path radius values when scoring graph edges. This is done by integrating the tubularity values along the paths in the scale-space tubularity image, such as the one in Fig. 2(b). This is how we computed the weighted graph of Fig. 4.

6.1. Path Classification

To train the SVM classifier, we randomly sampled 10000 positive and 10000 negative paths from the graphs. For assessing the ROC performance, we used 2500 positive and negative samples at detection time.

Fig. 6(a) shows the classification performance of our path features on the two datasets. The roads yield a higher true positive rate (TPR) at a fixed false positive rate (FPR) than the OPF one, which is mostly attributable to the rich gradient information available in the road images. Fig. 6(b-e) provides an analysis of the effect of the feature parameter

⁴The ground truth tracings and our acquisition software are publicly available at <http://cvlab.epfl.ch/data/delin>.

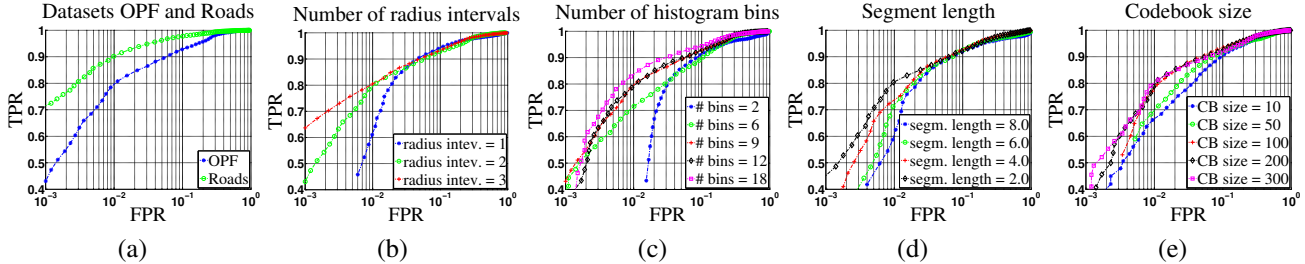


Figure 6: (a) ROC curves for the Road and OPF datasets. (b-e) Influence of the feature extraction parameters on the classification accuracy for the OPF dataset.

	OP4	OP6	OP8	ny1	ny3	ny6	ny7	ny12	ny14	ny15
k-MST [22]	0.865	0.898	0.722	0.314	0.209	0.195	0.159	0.24	0.32	0.235
HGD-QMIP	0.923	0.911	0.722	0.92	0.936	0.907	0.915	0.565	0.308	0.683

Table 1: Tree reconstruction accuracy measured by the DI-DEM [2] scores on our test datasets. OP_i and ny_i denote images from the OPF and the road datasets respectively.

settings discussed in Section 5.3 on classification accuracy. Briefly, increasing the number of radius intervals, the angular bins and the codebook entries improves the accuracy at the expense of a higher computational cost. Short segment lengths provides a fine grained sampling of paths and hence yields a higher performance.

6.2. Tree Reconstruction

Fig. 7 illustrates three sample reconstructions for the OPF and Road datasets. Additional ones are supplied as supplementary material. While the k-MST algorithm generates spurious branches and shortcuts, ours consistently eliminates outliers without either shortcuts and missing branches. This is especially visible in the road images for which the tubular graphs are dense.

For a more quantitative evaluation, we use the DIADEM metric [2], which computes a similarity score between a reconstruction and the ground-truth tree. Table 1 provides the DIADEM scores for the test images of the two datasets. Our approach systematically outperforms the k-MST algorithm except in one case on which both methods perform poorly.

7. Conclusion

We have proposed a novel approach to tubular tree structure delineation that lets us find the desired tree as the global optimum of a well-designed objective function. A key ingredient is a classification-based approach to scoring the quality of paths, which allows us to outperform a state-of-the-art method without having to manually tune the algorithm parameters for each new dataset.

This stability of the parameters, however, comes at the cost of requiring training data, which can be tedious to obtain. Future work will therefore focus on Transfer Learning techniques that should allow us to retrain our system using minimal amounts of such training data for each new modality we have to deal with.

References

- [1] K. A. Al-Kofahi, S. Lasek, D. H. Szarowski, C. J. Pace, G. Nagy, J. N. Turner, and B. Roysam. Rapid Automated Three-Dimensional Tracing of Neurons from Confocal Image Stacks. *TITB*, 6(2):171–187, 2002. 1
- [2] G. A. Ascoli, K. Svoboda, and Y. Liu. Digital Reconstruction of Axonal and Dendritic Morphology Diadem Challenge, 2010. <http://diademchallenge.org/>. 1, 2, 6, 7
- [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. 1
- [4] F. Benmansour and L. D. Cohen. Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision*, 92(2):192–210, 2011. 3
- [5] P. Chothani, V. Mehta, and A. Stepanyants. Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, 9:263–278, 2011. 2
- [6] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005. 5
- [7] D. Donohue and G. Ascoli. Automated reconstruction of neuronal morphology: An overview. *Brain Research Reviews*, 67:94–102, 2011. 1
- [8] M. Dorigo and T. Stütale. *Ant Colony Optimization*. MIT press, 2004. 4
- [9] C. Duhamel, L. Gouveia, P. Moura, and M. Souza. Models and Heuristics for a Minimum Arborescence Problem. *Networks*, 51(1):34–47, 2008. 4, 5
- [10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection With Discriminatively Trained Part Based Models. *PAMI*, 2009. 5
- [11] 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, 3
- [12] 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. 1
- [13] G. Gonzalez, F. Aguet, F. Fleuret, M. Unser, and P. Fua. Steerable Features for Statistical 3D Dendrite Detection. In *MICCAI*, pages 625–32, September 2009. 1
- [14] M. Jacob and M. Unser. Design of Steerable Filters for Feature Detection Using Canny-Like Criteria. *PAMI*, 26(8):1007–1019, August 2004. 1
- [15] C. Kirbas and F. Quek. Vessel Extraction Techniques and Algorithms: a Survey. In *Symposium on Bioinformatics and BioEngineering*, pages 238–245, 2003. 1

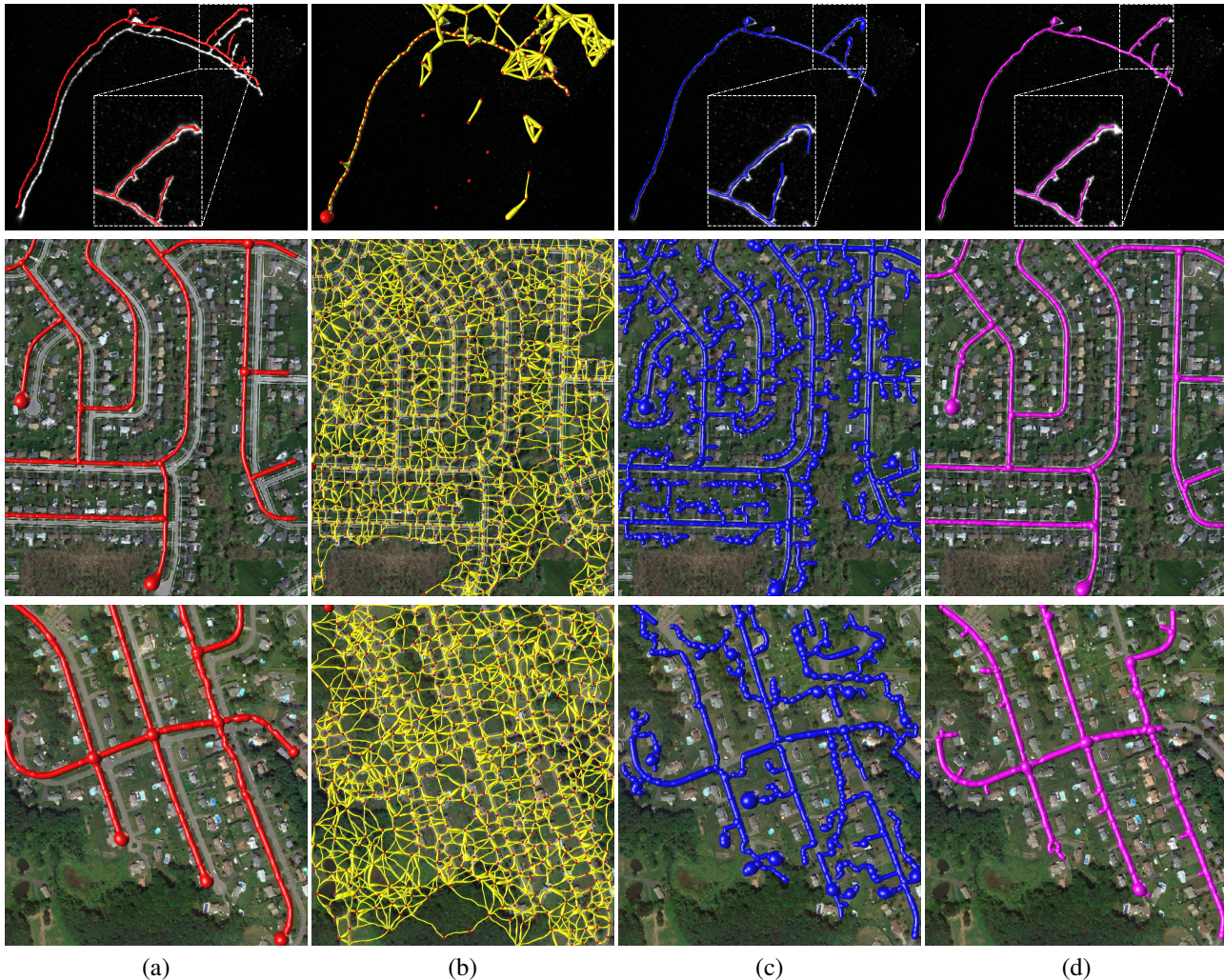


Figure 7: Tree reconstruction results on sample images from the evaluation datasets. (a) Ground truth overlaid in red. The paths are drawn as consecutive circles or spheres whose radius corresponds to the estimated width. They have been shifted from their true position to make the underlying linear structures visible. (b) Graphs obtained by the method of Section 4.1. The large red sphere denotes the root vertex of the tree structure. Path widths are downscaled to allow a better visualization. (c) Output of the k-MST algorithm [22] overlaid in blue. (d) Our results overlaid in magenta.

- [16] M. Law and A. Chung. Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux. In *ECCV*, pages 368–382, 2008. 1, 3
- [17] M. Law and A. Chung. An Oriented Flux Symmetry Based Active Contour Model for Three Dimensional Vessel Segmentation. In *ECCV*, pages 720–734, 2010. 2
- [18] H. Li and A. Yezzi. Vessels as 4-D curves: Global minimal 4-D paths to extract 3-D tubular surfaces and centerlines. *IEEE Transactions on Medical Imaging*, 26(9):1213–1223, 2007. 2, 3
- [19] E. Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77(7):693–704, 2010. 1
- [20] G. M. P. Hanchuan, F. Long. Automatic 3D neuron tracing using all-path pruning. *Bioinformatics*, 27(13):239–247, 2011. 2
- [21] Y. Sato, S. Nakajima, H. Atsumi, T. Koller, G. Gerig, S. Yoshida, and R. Kikinis. 3D Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. *Medical Image Analysis*, 2:143–168, June 1998. 1
- [22] E. Turetken, G. González, C. Blum, and P. Fua. Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9:279–302, May 2011. 2, 3, 4, 6, 7, 8
- [23] Y. Wang, A. Narayanaswamy, and B. Roysam. Novel 4-D Open-Curve Active Contour and curve completion approach for automated tree structure extraction. In *CVPR*, pages 1105–1112, 2011. 1, 2, 3
- [24] D. Weinland, M. Ozuyal, and P. Fua. Making Action Recognition Robust to Occlusions and Viewpoint Changes. In *ECCV*, September 2010. 5, 6
- [25] 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. 1, 2, 3