

Reconstructing Geometrically Consistent Tree Structures from Noisy Images

Engin Türetken¹ *, Christian Blum², Germán González¹, and Pascal Fua¹

¹ Computer Vision Lab., Ecole Polytechnique Fédérale de Lausanne, Switzerland

² ALBCOM, Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract. We present a novel approach to fully automated reconstruction of tree structures in noisy 2D images. Unlike in earlier approaches, we explicitly handle crossovers and bifurcation points, and impose geometric constraints while optimizing a global cost function.

We use manually annotated retinal scans to evaluate our method and demonstrate that it brings about a very substantial improvement.

1 Introduction

Tree-like structures, such as vascular networks, dendritic trees, or bronchial networks, are pervasive in biological imagery. With the advent of modern acquisition techniques that produce endless streams of 2D and 3D imagery, there has been renewed interest in automated delineation as a means of exploiting this data. Of particular interest are topologically accurate delineations, which are critical for diagnosis and analysis purposes. However, despite many years of sustained effort, automated delineation techniques remain fragile and error-prone.

In earlier work [1], we showed that robustness could be improved by exploiting the global tree topology early in the algorithm. However, this method suffers from the fact that the tree-growing algorithm it uses makes all its decisions based on local image evidence without regard to tree shape. As a result, it still makes topological mistakes, such as those highlighted by circles in Fig. 1.

In this paper, we address this issue by incorporating into the tree reconstruction algorithm shape priors that enforce geometric consistency between pairs of graph nodes and edges. This involves solving a *k-cardinality arborescence problem*, which is known to be NP-hard [2]. Nevertheless, we will show that good approximate solutions can be obtained by extending the metaheuristic approach of [3]. We also explicitly model crossovers and bifurcations, which are particularly troublesome in 2D projections of 3D volumes and can result in spurious branches, gaps, and other topological errors if not handled properly.

Introducing geometry constraints and handling crossovers results in a robust fully automated delineation technique that we demonstrate on vascular trees of retinal fundus images. As shown in Fig. 1, it yields a significant improvement, which is also supported by quantitative evaluation.

* This work was supported in part by the Swiss National Science Foundation and in part by the MicroNano ERC project.

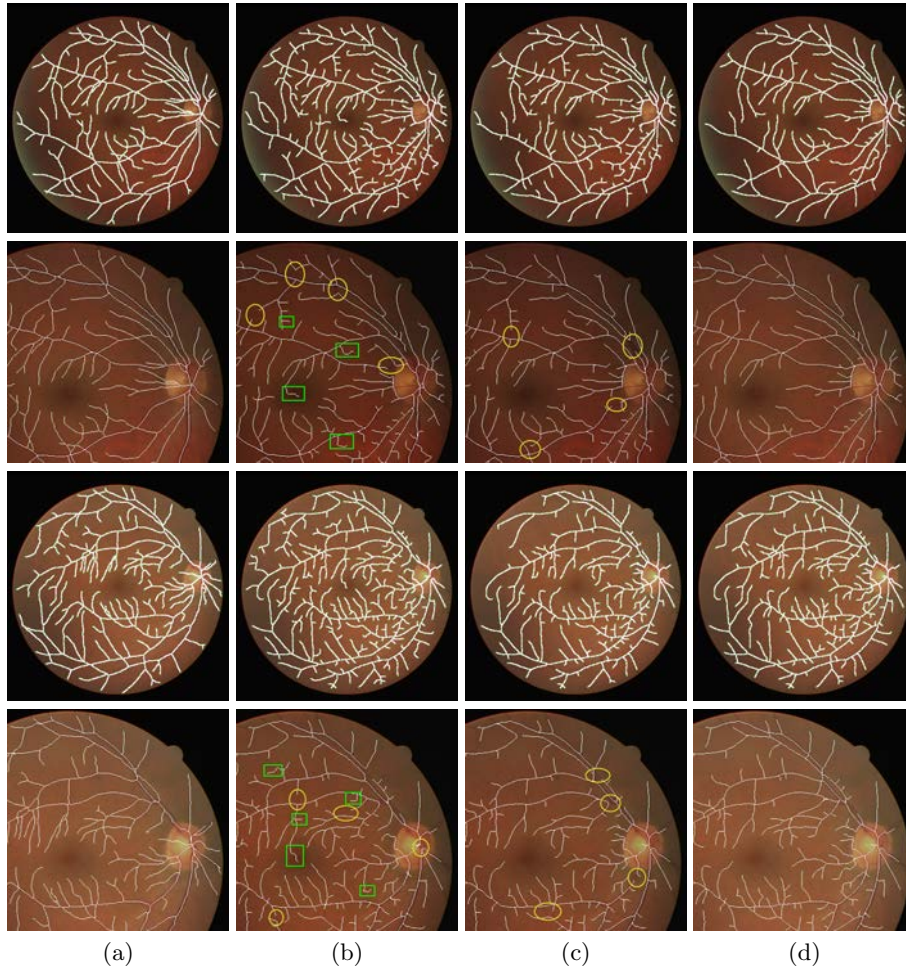


Fig. 1: The first and third rows depict vascular tree reconstructions from two different retinal scans. In the second and fourth rows, we show enlarged versions of the same images. (a) Original images with manually outlined blood vessels overlaid in white. (b) Trees reconstructed using a minimum spanning tree (MST) approach. (c) Trees reconstructed using our earlier method [1]. (d) Trees reconstructed using the proposed technique. Note the false positives and the false negatives, highlighted respectively by the green rectangles and the yellow circles in columns (b) and (c), have disappeared from column (d).

2 Related Work

Most automated delineation techniques start by computing a *tubularity* image in which pixels likely to belong to filaments have high scores. This image can then be thresholded and its skeleton computed [4]. This tends to produce disconnected components and artifacts on noisy data, which often require considerable post-processing and analysis for a correct tree to be produced.

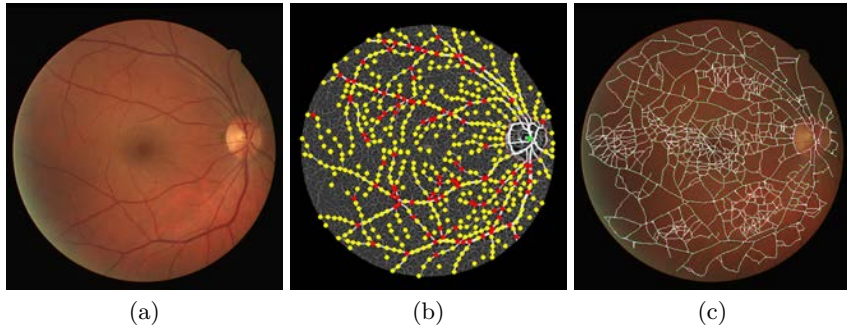


Fig. 2: Graph construction. (a) Original image. (b) Sampled points. The green point and the red circle represent the root node and optic disk region respectively. The red points are potential crossovers and the yellow ones are ridge points. (c) Graph built from the most probable paths.

Alternatively, the tubularity scores can be used to find seed points and recursively trace high-tubularity paths [5, 6]. Although computationally efficient, this technique lacks robustness since cumulative tracing errors can result in large topological ones. More global methods avoid this problem by using more of the image evidence and optimizing a global objective function [7, 8]. However, while such methods produce smooth tree components, they do not guarantee their spatial connectedness. Furthermore, they are computationally intensive, which limits their applicability to large datasets.

By contrast, methods that sample local maxima of the tubularity image and then connect these samples into a spanning tree [1, 9], guarantee connectivity. However, they do not take into account global tree geometry, such as smoothness along the edges or branching factors, which can play an important role in improving topological accuracy, avoiding over-fitting, and speeding up convergence. They also fail to explicitly account for bifurcations and crossovers, which can easily lead to mistakes. While post-processing pruning [9] can sometimes eliminate some spurious branches, it does not allow for recovery from other topological mistakes. This is the problem we address in this paper by introducing more global geometry constraints early in the algorithm to prevent such mistakes.

3 Method

Our method consists of the following steps:

1. We compute a tubularity value at each pixel [10], which encodes how likely it is to be on the centerline of an elongated linear structure.
2. To avoid having to compute a tree that spans individual image pixels, which would be prohibitively expensive, we select high-probability pixels such as those of Fig. 2(b) that are as evenly spaced as possible and treat them as the graph nodes to be linked.
3. We compute the most probable paths between pairs of nearby nodes, such as those of Fig. 2(c), and treat them as the edges of our directed graph. We assign them probabilistic costs that are lowest when all pixels along them are likely to lie in the middle of a filament.



Fig. 3: Two cases that can lead to reconstruction errors. (a) A crossover with a single sample point that can be used to build either the horizontal \overline{AXB} branch or the vertical \overline{YXZ} one, but not both. (b) If there is no sample point at the bifurcation D , incorporating both the \overline{AB} and \overline{AC} paths into the graph will result in counting twice the pixels in \overline{AD} . This is avoided by introducing a sample point at D .

4. We compute the lowest-cost arborescence among those that span K edges for a wide range of $K < N$, where N is the number of nodes. This is known as the K -Cardinality Arborescence Problem. Even though it is NP-Hard, approximate solutions can nevertheless be computed efficiently and fast.
5. Among these arborescences, we select the one that optimizes a global objective function.

The above workflow is the same as the one we introduced in our earlier work [1] with two key differences: First, in [1], we did not take into account the geometric properties of the trees when constructing them. Here, we incorporate into our objective function geometric terms that favor trees whose branches are smooth and along which the width remains consistent. Second, we explicitly model bifurcations and crossovers.

To solve the associated minimum arborescence problem, we had to substantially modify the tree building procedure [3] we used in our previous work. This produced very significant performance improvements, as shown in Fig. 1 and further discussed in Section 4.

3.1 Sampling

We first compute the response of the Rotational Features introduced in [10] at different scales and orientations at each image location x_i , and retain the maximum value as our tubularity score $f(x_i)$. We then map this score to a posterior probability that x_i belongs to a filament centerline, $P(x_i|f(x_i))$, by fitting a sigmoid function. We keep the orientation ϕ_i and the width estimates w_i of each point.

We then use a two-step approach to sampling local maxima of this probability image. First, we threshold and skeletonize it, and use the combined cross-point number method [11] to detect potential crossovers and bifurcations, which we will refer to as *landmarks* such as those depicted by the red dots in Fig. 2(b). For each landmark, we create two colocated nodes to prevent problems at tree reconstruction time, such as those depicted by Fig. 3(a). The sampling of bifurcations such as the one in Fig. 3(b) is useful to avoid overcounting pixels when scoring the trees that we reconstruct. In the second step, we sort the remaining pixels according to their probability of belonging to a filament, select the

most probable one, eliminate all those within a certain radius, and iterate. This produces the regularly spaced samples shown as yellow dots in Fig. 2(b).

In the specific case of retinal scans, we know *a priori* that the root of the tree we want to build is located in the optic disk, which is depicted by the red outline in Fig. 2(b). We use a variant of [12] to detect the optic disk and remove all samples that reside within it except the one nearest to its center, which will be assigned as the tree root.

3.2 Building the Graph

The procedure described above returns a set V of nodes. We construct a directed graph $G = (V, E)$ such as the one of Fig. 2(c) by linking all pairs of samples v_m and $v_n \in V$ that are within a certain distance of each other—except colocated ones—by a Dijkstra path $e_{mn} \in E$ that minimizes the integral of the negative log of the posterior probabilities $P(x_i|f(x_i))$ introduced in Section 3.1.

In [1], we showed that such paths are maximal probability paths between the vertices and that their total costs c_{mn}^{nl} can be treated as sum of negative log likelihoods along them. We also showed that if we assume that all geometric arrangements of edges are equally probable, a near-optimal tree can be obtained by minimizing

$$F^{nl}(T(k)) = \sum_{e_{mn} \in T(k)} c_{mn}^{nl} , \quad (1)$$

where $T(k)$ denotes a tree of cardinality k . However, in reality, not all trees are equally plausible. Those whose branches are smooth, conform to the underlying image orientation, and whose widths vary slowly and consistently, are much more likely to be correct than others.

To account for this, we exploit four geometric terms to capture the underlying relations between parts of the tree structure:

1. **Edge Direction Similarity** (Φ_e) We model the angular difference between pairs of adjacent directed edges by a von Mises distribution (circular normal distribution).
2. **Width Consistency** (Φ_w) We model the width differences of pair of vertices connected through directed edges by an asymmetric Gaussian distribution, since for most datasets a decrease in width is more probable than an increase along a directed path from the root vertex.
3. **Orientation Consistency** (Φ_o) For pairs of connected vertices, we measure the angular deviations of their orientations from the direction of the line between them. The deviations are modeled using a von Mises distribution.
4. **Tortuosity** (Φ_t) We compute tortuosity values of the paths corresponding to edges and represent them by a Gaussian distribution. For the vascular reconstructions obtained in this paper, the tortuosity measure that we used is the ratio of the path length to the linear distance between the endpoints.

We estimate the parameters of these distributions using maximum likelihood estimation (MLE). Given the individual terms, the combined pairwise potential

can be written as

$$\begin{aligned} \Phi(e_{mn}|e_{rm}, \mathbf{w}_{rmn}, \phi_{rmn}) = & \Phi_e(e_{mn} | e_{rm}) + \Phi_w(e_{mn} | w_m, w_n) + \\ & \Phi_o(e_{mn} | \phi_m, \phi_n) + \Phi_t(e_{mn} | l(v_m, v_n)) . \end{aligned} \quad (2)$$

where $\mathbf{w}_{rmn} = (w_r, w_m, w_n)$ and $\phi_{rmn} = (\phi_r, \phi_m, \phi_n)$ denote triplets of width and orientation estimates, and $l(v_m, v_n)$ the estimated path between vertices v_m and v_n . The criterion we optimize then becomes

$$F^{\text{nl}}(T(k)) = \sum_{e_{mn} \in T(k)} c_{mn}^{\text{nl}} + \sum_{\substack{e_{rm} \in T(k) \\ e_{mn} \in T(k)}} \Phi(e_{mn} | e_{rm}, \mathbf{w}_{rmn}, \phi_{rmn}) . \quad (3)$$

When k is given, finding the tree that minimizes $F^{\text{nl}}(k)$ yields very good results. In practice, however, k is unknown and the criterion of Eq. 3 suffers from one severe drawback: since the cost of additional edges is always positive, it systematically favors low values of k . To overcome this difficulty, when scoring the final tree, we count not only the cost of including some edges in the tree but also of discarding those that do not belong to it. This turns out to be equivalent to replacing the c_{mn}^{nl} , which are the sums of negative log likelihoods along the paths, by c_{mn}^{llr} computed by summing log likelihood ratios $-\log(P(x_i|f(x_i))/(1 - P(x_i|f(x_i))))$ along the paths. We then take the global score to be

$$F^{\text{llr}}(T(k)) = \sum_{e_{mn} \in T(k)} c_{mn}^{\text{llr}} + \sum_{\substack{e_{rm} \in T(k) \\ e_{mn} \in T(k)}} \Phi(e_{mn} | e_{rm}, \mathbf{w}_{rmn}, \phi_{rmn}) . \quad (4)$$

In effect, replacing the log likelihoods by the log likelihood ratios amounts to penalizing graph edges with high probabilities that are left out of the final tree and overcomes the bias for low values of k [1].

In summary, for each cardinality $k=2 \dots N$, we build the tree that minimizes Eq. 3, and assign to it the score of Eq. 4. The final result is the tree that minimizes such score among all cardinalities,

$$\hat{\mathbf{T}}^* = \underset{T(k) \in \{T(2), \dots, T(N)\}}{\operatorname{argmin}} F(T(k)). \quad (5)$$

3.3 Estimating the Optimal Tree

Both to force trees to grow from the estimated root and to take advantage of the pairwise terms to guide the reconstruction process, we extended the ant colony optimization based algorithm presented in [3]. Due to space limitations, here we only sketch the modifications we made.

First, to operate on directed graphs, we modify the neighborhood structure to only contain those edges that point away from the leaves of the current arborescence. Second, we fix the root node and initiate the search from it. At each growing step, we compute the effective cost of an edge in this neighborhood by taking into account the pairwise potentials introduced in Section 3.2. An edge

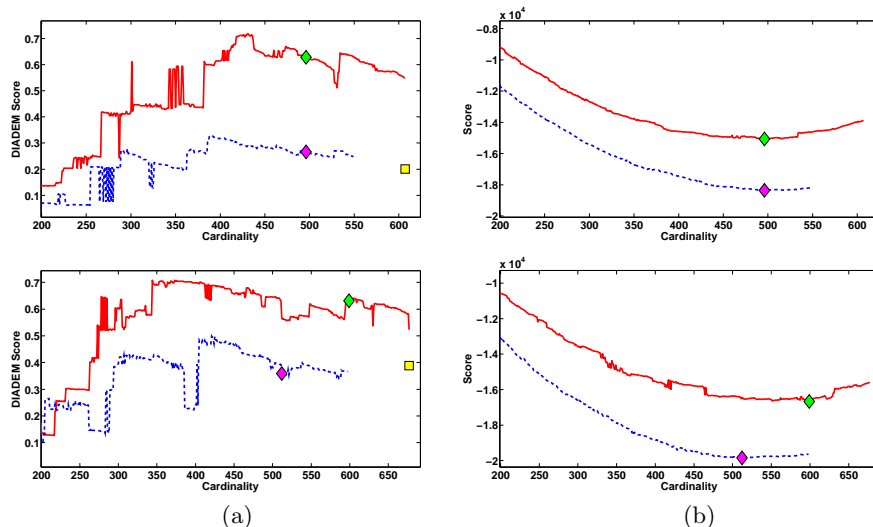


Fig. 4: Quantitative evaluation of the reconstructions in the two retinal scans of Fig. 1. (a) DIADeM scores as a function of the tree cardinality for our earlier approach [1] in blue dotted line, and for the one presented here in red solid line, which is substantially better. The yellow squares represent the scores of the standard MST. (b) Corresponding scores computed using the criterion of Eq. 4. In both cases, the diamonds denote the selected cardinality, taken to be the one that minimizes this criterion.

is then stochastically selected based on this cost and the number of times it has been previously selected in the search. Finally, we set a limit on the bifurcation factor of a node in order avoid false connections on crossovers.

The above procedure is run several times until the obtained minimum costs for all cardinalities stabilize. Finally, we pick among all these arborescences of different cardinalities the one that minimizes the criterion of Eq. 4. The algorithm is very efficient since it requires only a few minutes to converge to topologically sound reconstructions, such as the ones shown in Fig. 1.

4 Results

The retinal scans of Fig. 1 belong to the DRIVE database [13]. We manually outlined the vascular trees and treat them as ground truth. These ground truths are used to compare the quality of the reconstructions of the presented method as shown in Fig. 1(d), against a spanning tree as given in Fig. 1(b), and the work of [1], Fig. 1(c).

To compare the methods quantitatively, we use the metric introduced by the DIADeM challenge [14], which is specifically designed to compare topology of a reconstructed tree against ground truth. The metric returns a number between 0 and 1 that measures the topological distance between trees by matching their branching and end points, and then analyzing the connecting paths. Mistakes close to the tree root are penalized more heavily than the ones closer to the leaves since they produce more severe topological changes.

Fig. 4(a) shows that the method proposed here yields a very substantial improvement over the other methods with respect to the DIADEM metric, which is consistent with the qualitative results presented Fig. 1(d). The optimal tree cardinalities, illustrated by the diamonds in Fig. 4(b), are automatically obtained by minimizing the score of Eq. 4.

5 Conclusion

We have presented an algorithm for automatic tree reconstruction that enforces geometric constraints such as smoothness and width consistency along the branches, while explicitly handling crossovers and bifurcations. This yields a substantial qualitative and quantitative improvement in the reconstructions of retinal vascular trees at an acceptable computational cost.

In future work, we will extend our approach to other domains such as 3D dendrite delineation and generalize the type of constraints we can impose.

References

1. Gonzalez, G., Turetken, E., Fleuret, F., Fua, P.: Delineating Trees in Noisy 2D Images and 3D Image-Stacks. In: CVPR, San Francisco, CA (June 2010)
2. Chimani, M., Kandyba, M., Ljubić, I., Mutzel, P.: Obtaining optimal k-cardinality trees fast. *J. Exp. Algorithmics* **14** (2009) 2.5–2.23
3. Blum, C., Blesa, M.J.: New metaheuristic approaches for the edge-weighted k-cardinality tree problem. (2005) 1355–1377
4. Leandro, J., Soares, J., Cesar, R., Jelinek, H.: Blood vessels segmentation in non-mydratic images using wavelets and statistical classifiers. *Brazilian Symposium on Computer Graphics and Image Processing* **0** (2003) 262
5. Yedidya, T., Hartley, R.: Tracking of blood vessels in retinal images using kalman filter. In: DICTA, Washington, DC, USA, IEEE Computer Society (2008) 52–58
6. Can, A., Shen, H., Turner, J., Tanenbaum, H., Roysam, B.: Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. *TITB* **3**(2) (June 1999) 125–138
7. Fan, D.: Bayesian inference of vascular structure from retinal images. PhD thesis, Dept. of Computer Science, U. of Warwick, Coventry, UK (2006)
8. Sun, K., Sang, N., Zhang, T.: Marked point process for vascular tree extraction on angiogram. In: EMMCVPR. (2007) 467–478
9. Gonzalez, G., Fleuret, F., Fua, P.: Automated Delineation of Dendritic Networks in Noisy Image Stacks. In: ECCV, Marseille, France (October 2008)
10. Gonzalez, G., Fleuret, F., Fua, P.: Learning Rotational Features for Filament Detection. In: CVPR, Miami, FL (June 2009)
11. Aibinu, A., Iqbal, M., Shafie, A., Salami, M., Nilsson, M.: Vascular intersection detection in retina fundus images using a new hybrid approach. *Computers in Biology and Medicine* **40**(1) (2010) 81–89
12. Huang, K., Yan, M.: Robust Optic Disk Detection in Retinal Images using Vessel Structure and Radon Transform. In: SPIE. Volume 6144. (2006)
13. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging* **23** (2004) 501–509
14. HHMI: Diadem challenge (2010) <http://diademchallenge.org/>.