Automated Separation of Binary Overlapping Trees in Low-Contrast Color Retinal Images

Qiao Hu[1], Michael D. Abràmoff[2,1,3], and Mona K. Garvin[3,1]

[1] Departments of Electrical and Computer Engineering
[2] Ophthalmology and Visual Sciences, The University of Iowa, Iowa City, IA, USA
[3] Department of Veterans Affairs Health Care System, Iowa City, IA, USA
{qiao-hu, michael-abramoff, mona-garvin}@uiowa.edu

Abstract. While many approaches exist for the automated segmentation of retinal vessels in fundus photographs, limited work has focused on the problem of separating the arterial from the venous trees. The few existing approaches that do exist for separating arteries from veins are local and/or greedy in nature, making them susceptible to errors or limiting their applicability to only the very largest vessels. In this work, we propose a new, more global, optimization framework for separating two overlapping trees within medical images and apply this approach for the separation of arteriovenous trees in low-contrast color fundus images. In particular, our approach has two stages. The first stage is to generate a vessel potential connectivity map (VPCM) consisting of vessel segments and the potential connectivity between them. The second stage is to separate the VPCM into multiple anatomical trees using a graph-based meta-heuristic algorithm. Based on a graph model, the algorithm first uses local knowledge and global constraints of the vasculature to generate near-optimal candidate solutions, and then obtains the final solution based on global costs. We test the algorithm on 48 low-contrast fundus images and the promising results suggest its applicability and robustness.

1 Introduction

The retinal vasculature is an important structure for indicating ophthalmic disease [1]. However, while many approaches exist for the segmentation of retinal vessels [2], a much smaller number of approaches actually focus on separating the retinal vessels into arterial and venous trees. Martinez-Perez *et al.* [3] propose an approach which segments and thins vessels first, and then recognizes bifurcations and crossovers using local neighbor information to build trees. Lin *et al.* [4] propose a grouping algorithm to iteratively connect un-grouped segments to grouped segments by maximizing the continuity of the vessel using an extended Kalman filter. Joshi *et al.* [5] propose a structural mapping method which first detects landmarks and then uses path-based graph approaches to solve the problem. Rothaus *et al.* [6] propose a new grouping approach to separate arterial and venous trees using a graph modeled as a SAT problem. The graph structure can be changed dynamically to resolve some conflicts, but manual inputs are required to initialize the labeling and if some conflicts cannot be resolved.

These existing approaches usually depend on local and/or greedy decisions and are correspondingly susceptible to local errors, especially with ambiguous local image information and/or inaccuracies in the automated vessel segmentations. Some common errors are: (a) the misclassification of a crossing point as a bifurcation when one vessel is missing or disconnected; (b) the disconnection of the tree due to a missing part of a vessel; and (c) the identification of false bifurcations and crossings due to spurious vessels. In addition, complex landmarks are hard to recognize with local knowledge (e.g., two overlapping vessels which would be misclassified as one vessel, and two overlapping landmarks).

In this paper, we propose a novel and more global framework to build the vasculature which tolerates errors introduced during imaging and pre-processing. In general, our framework first extracts partitioned vessel segments and repairs lost connectivities by over-connecting them to generate a vessel potential connectivity map (VPCM). The problem of separating the segments into underlying trees is modeled as a graph-based optimization problem that allows for global costs, which is a key novelty of our approach. Because of its NP-hard nature, a meta-heuristic algorithm is designed to find a near-optimal solution.

2   Method

2.1   The Vessel Potential Connectivity Map

The VPCM consists of a set of partitioned vessel segments and their potential anatomical neighbors. For a VPCM, each segment v_i has two ends ep_{2i} and ep_{2i+1}, which are connected to the ends of neighboring vessels. Each set of end points in a close proximity is defined as a neighborhood, denoted as $\mathcal{N}_k$. Thus ep_i and ep_j are neighbors if $ep_i, ep_j \in \mathcal{N}_k$, and segments v_i and v_j are neighbors if either of their ends are neighbors. $|\mathcal{N}_k|$ is the number of vessel ends within $\mathcal{N}_k$, and neighborhoods are only established when $|\mathcal{N}_k| > 1$.

To obtain a VPCM, we first obtain the vessel skeleton by thresholding a vessel classification probability map [2], and skeletonizing it. Then vessel segments are obtained by cutting the vessel skeleton at critical pixels which have multiple neighbors, and when the vessel direction changes dramatically. A two-step algorithm is developed to find potential neighbors for each segment. The first step is to artificially extend the vessel ends without neighbors until they find a neighbor, or reach the maximum extension. The maximum extension is controlled by a circular sector whose radius is r and central angle is θ. Here r is proportional to the vessel length and inversely proportional to vessel tortuosity [7], and θ is set as $30°$ (with use of other values of θ not affecting the results). If the extension from ep_i meets another vessel end ep_j, then ep_i and ep_j are connected; if the extension meets another vessel pixel of v_m, then v_m is cut into two segments and ep_i is connected to the two end points generated by the cutting; if the extension meets another extension from ep_k, ep_i and ep_k are also connected.

The second step is to merge close neighborhoods and to re-connect falsely separated segments. A polygonal boundary is automatically constructed for each
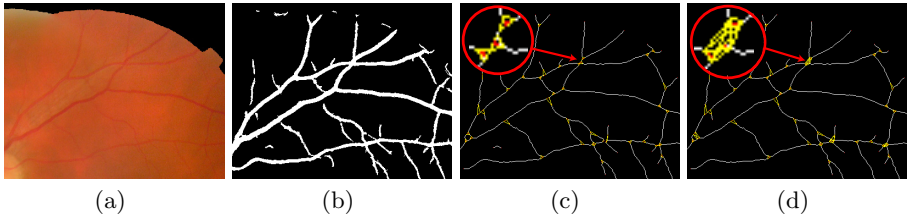
(a)	(b)	(c)	(d)

Fig. 1. The construction of VPCM. (a) Portion of fundus image. (b) Vessel segmentation (using pixel classification approach [2]). (c) The VPCM before the merge of neighborhoods. The neighborhoods are indicated with yellow boundaries. The enlarged portion within the red circle shows three neighborhoods. (d) The final VPCM. The enlarged portion within the red circle shows one neighborhood.

$\mathcal{N}_k$ and two neighborhoods merge if their boundaries meet. An example resulting VPCM is shown in Fig. 1.

2.2   The Optimization Problem

Effectively, the resulting VPCM from the prior step provides us with a set of neighborhoods (and the vessel segments belonging to each neighborhood). Intuitively, neighborhoods correspond to potential anatomical structures such as branching points and crossing points and we desire to make this determination based on both local and global information. Because arteries do not cross with arteries and veins do not cross with veins, we can assume that vessels of the same type within a neighborhood should belong to the same tree. Thus, determining the anatomical connectivity between segments is equivalent to the problem of determining their types. Bearing this idea, the problem of constructing the anatomical trees on the VPCM is viewed as to separate a graph into multiple trees with binary labelings. Let l_i reflect the label of segment v_i ($l_i \in [0, 1]$, $l_i = l(v_i) = l(ep_{2i}) = l(ep_{2i+1})$). Also denote a vector of neighborhood labelings, $\boldsymbol{l}_{\mathcal{N}_k}$, such that $l_i \in \boldsymbol{l}_{\mathcal{N}_k}$ if ep_{2i} or $ep_{2i+1} \in \mathcal{N}_k$. Our optimization problem is to find the minimum-cost vector of labelings $\boldsymbol{L}$ for all segments with the constraint that the generated anatomical trees have no cycles. The overall cost, $E(\boldsymbol{L})$ (eq. 1), contains a weighted combination of a global cost term to evaluate topological properties, $F(G)$, and a summation of local neighborhood cost terms, with each neighborhood cost computed as the reciprocal of the probability of $\boldsymbol{l}_{\mathcal{N}_k}$:

$$E(\boldsymbol{L}) = \beta F(G) + \sum_{k=0}^{K-1} E_{\mathcal{N}}(\boldsymbol{l}_{\mathcal{N}_k}) \ , \tag{1}$$

$$\text{where } E_{\mathcal{N}}(\boldsymbol{l}_{\mathcal{N}_k}) = \frac{1}{P(\boldsymbol{l}_{\mathcal{N}_k} | \mathcal{N}_k, A_{\mathcal{N}_k})} \ . \tag{2}$$

Intuitively, $P(\boldsymbol{l}_{\mathcal{N}_k} | \mathcal{N}_k, A_{\mathcal{N}_k})$ reflects, for a neighborhood $\mathcal{N}_k$, the probability of its landmark type (e.g. a branching point or a crossover) represented by its

labeling $\boldsymbol{l}_{\mathcal{N}_k}$ given its property matrix $A_{\mathcal{N}_k}$ (eq. 2). While we keep the general form of the terms here for describing the overall algorithm, Section 2.5 provides more details about how $P(\boldsymbol{l}_{\mathcal{N}_k}|\mathcal{N}_k, A_{\mathcal{N}_k})$ was computed in this work and Section 2.6 provides more details about $F(G)$. Because of the NP-hard nature of this optimization problem, we design a special graph model to incorporate the global constraint of the anatomical trees and use the local term of eq. 1 to generate candidate solutions. These candidate solutions are then transferred back to the image domain to obtain the final solution with the global cost.

2.3   The Graph Model

In the graph model, each vertex n_i represents a segment v_i and each edge e_{ij} represents a relation between segments v_i and v_j within a neighborhood. For $|\mathcal{N}_k| = n$, $n-1$ edges are applied to connect the n vertices to form a simple cycle-free graph. Here a concept of cluster $\mathcal{C}_k$ is introduced in the graph corresponding to the neighborhood $\mathcal{N}_k$ in the image domain, which is a set of edges generated by the vessel ends within one neighborhood. If in the image domain, $ep_i, ep_j \in \mathcal{N}_k$, and corresponding edge $e_{\lfloor i/2 \rfloor \lfloor j/2 \rfloor}$ is one of the $|\mathcal{N}_k| - 1$ edges, then $e_{\lfloor i/2 \rfloor \lfloor j/2 \rfloor} \in \mathcal{C}_k$. Fig. 2(a) shows examples of how a 3-p and a 4-p $\mathcal{N}$ are transferred to clusters in the graph domain. Fig. 2(b) shows a virtual VPCM with four neighborhoods transferred to a graph. Four neighborhoods in the image domain corresponds to the four clusters in the graph, whose edges are labeled in different colors.

   In the graph, each vertex needs to be labeled in one of two colors, corresponding to vessels being assigned one of two types. Each edge is associated with one of two constraints: equality or inequality. The equality constraint dictates the two vertices connected by the edge must be in the same color; the inequality constraint dictates they must be in different colors. Within each cluster, a combination of edge constraints is equivalent to a two-color scheme on vertices, which reflects the equivalence of the problem of determining anatomical connectivity and the problem of determining segment types within each neighborhood.

   Corresponding to the VPCM, vertex labelings and edge constraints for one cluster are mutually inferable. Fig. 2(c) shows an example of a 4-p neighborhood, three potential landmark types, and its corresponding cluster in the graph, with their edge constraints and vertex labelings. In this example, case (i) indicates v_0, v_1 and v_2 are the same type and connected, v_3 is another type and disconnected with them, meaning v_0, v_1 and v_2 form a bifurcation, and v_3 is falsely connected to it; case (ii) shows another possible bifurcation case; case (iii) indicates v_0 and v_2 are the same type thus connected, v_1 and v_3 are the other type, meaning the neighborhood represents a crossing point.

   By this transformation, the optimization problem in the VPCM is transferred on the graph so as to find the proper edge combinations for each cluster, such that there is a feasible two-color solution for the graph, meanwhile having a minimum sum of their associated costs. If there is no cycle in the graph, this is a trivial problem. The solution is simply to choose the combination of edge constraints with lowest costs for each cluster. However, if there is a cycle in the graph with an odd number of inequality edges, there is no feasible solution to
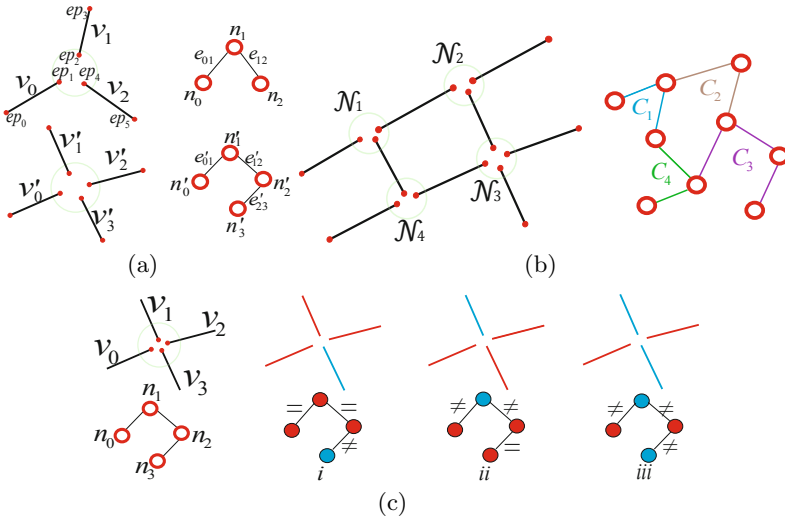
Fig. 2. Graph examples. (a) Neighborhoods transferred into clusters. (b) A virtual VPCM transferred into a graph. (c) First row: a 4-p neighborhood and three example landmark types. Second row: corresponding cluster with different cases of edge constraints and color labelings.

color vertices on the cycle with two colors. Thus, any cycle with an odd number of inequality edges is a conflict cycle and must be dealt with as described in the next section.

2.4   The Heuristic Algorithm

Here we define a solution s as the choices of edge constraints on a graph, and a feasible solution as a solution such that vertices can be colored in two colors (i.e., there are no conflict cycles). As is common in many meta-heuristic algorithms, the overall idea is to maintain a candidate solution pool, with candidate solutions generated as in Algorithm 1. One of the key components of the approach is the ability to handle and resolve conflict cycles. Starting from an initial solution, the algorithm checks for conflict cycles. This is done by detecting a cycle basis first and then checking each cycle of the basis. Given an undirected graph, a cycle basis is a minimal set of circuits such that any cycle can be written as a sum of the circuits in the basis [8].

If there are conflict cycles, the algorithm permutates the edge constraints to reduce the number of conflict cycles. The solution pool is updated during every conflict cycle reduction. The algorithm stops when enough number of feasible solutions are generated ($n_f = 100$ in this work), or the iteration number reaches the maximum limit ($I_{\max} = 300$ in this work).

Within Algorithm 1, another key component is *permutateEdgeConstraint*(s, C, m) which permutates the edge constraints of clusters to remove conflict cycles. Here m is the number of solutions generated by a single parent solution (in

Algorithm 1. Find candidate solutions given a graph G with cycle basis B, n_f (max number of feasible solutions to find), m (number of child solutions generated by a parent solution), initial solution t, and $I_{\max}$ (max iteration number)

Initialize solution pool $\mathcal{S}$ with t, initialize feasible solution pool $\mathcal{F} \leftarrow \varnothing$
while $|F| < n_f$ & $i < I_{max}$ **do**
  Initialize solution pool $\mathcal{S}' \leftarrow \varnothing$
  for each solution $s \in \mathcal{S}$ **do**
    $C \leftarrow findConflictCycle(s, B)$
    if $|C| = 0$ **then**
      put s into $\mathcal{F}$ if $s \notin \mathcal{F}$
    else
      solution pool $\mathcal{N} \leftarrow permutateEdgeConstraint(s, C, m)$
      for each solution n in $\mathcal{N}$ **do**
        put n in $\mathcal{F}$ if it is feasible ($|findConflictCycle|{=}0$); otherwise put n in $\mathcal{S}'$
      end for
    end if
  end for
  $\mathcal{S} \leftarrow \mathcal{S}'$, $i \leftarrow i + 1$
end while
return $\mathcal{F}$

---

this work, $m = \max(6, c/2)$, where c is the number of clusters on a conflict cycle route). This algorithm considers edges on the conflict cycle and finds their corresponding clusters. All the potential configurations of these clusters are sorted based on their costs. Random Gaussian noise is added during the sorting to enlarge the search region and to prevent being trapped in local minima.

2.5   Configurations and Costs of Local Neighborhoods

We calculate $P(l_{\mathcal{N}_k}|\mathcal{N}_k, A_{\mathcal{N}_k})$ by combining local knowledge and the global knowledge of the retinal vasculature using different algorithms according to $|\mathcal{N}_k|$ as the probabilities of landmark types for each neighborhood. Global knowledge is motivated from a concept of blood flow since blood flows for arteries from the optic disc (OD) to the periphery of the image, and in the reversed direction for veins. Defining the flow direction from the tail ep_{it} to the head ep_{ih}, we can distinguish two ends of a segment as a head and a tail. This enables the higher probabilities to be associated with configurations that properly connect heads with tails. Local properties include the angle θ_{ij} between directions of segments within one neighborhood, and the distances between end points.

   When $|\mathcal{N}_k| \geq 4$, we limit the possible landmark types with prior information. Specifically, when $|\mathcal{N}_k| = 5$, we only consider the case that $\sum_{ep_i \in \mathcal{N}_k} l_{\lfloor i/2 \rfloor} = 2$, which represents the case of the overlapping of a bifurcation and a crossing point. When $|\mathcal{N}_k| = 6$, we consider the cases that $\sum_{ep_i \in \mathcal{N}_k} l_{\lfloor i/2 \rfloor} = 2$ (overlapping of two close bifurcations in one vessel and a crossing point) and $\sum_{ep_i \in \mathcal{N}_k} l_{\lfloor i/2 \rfloor} = 3$ (overlapping of two bifurcations).

2.6   Global Costs

After generating candidate solutions, a global cost term (eq. 3) is added to find the lowest-cost overall solution. The first term of eq. 3 is the distance of the root of every connected tree t (with the same label) to the optic disc. The second term penalizes unlikely angles at each bifurcation b. θ_1 and θ_2 are the angles between two child vessels and the parent vessel; and l_1 and l_2 are their lengths.

$$F(G) = \sum_{t \in G} d(t) + \alpha \sum_{t \in G} \sum_{b \in t} h(b) \ , \qquad (3)$$

$$\text{where } h(b) = l_1 e^{\frac{\pi}{2} - \theta_1} + l_2 e^{\frac{\pi}{2} - \theta_2} \ . \qquad (4)$$

3   Experiments and Results

Our method is tested on 48 color low-contrast registered fundus images. A mask is manually applied to mask vessels within the optic disc. The whole VPCM is first separated into independent sub-VPCMs, each of which is transferred to a graph. The final solution is the vessel centerline labeled in two colors. Notice that the two-color labelings only indicate the connectivity, which is the key step for determining the arterial and venous trees, but the correspondence of these colors to arteries/veins still needs to be made. We categorize a sub-VPCM into a simple VPCM which has only one segment (not evaluated as trivial) and a complex VPCM which has more than one, which is further categorized as acyclic VPCM which has no cycles and a cyclic VPCM. In these 48 images, there are an average of 10.3 sub-VPCMs, 6.6 complex and 2.6 cyclic VPCMs per image.

To evaluate the performance, a reference standard is generated by manually labeling vessel types on each VPCM by an expert. Since the VPCM is generated from an automatic segmentation, four types of vessels are labeled: arteries (A), veins (V), overlapping of both (O), and the false positive (FP) or uncertain (U). This enabled us to specifically evaluate the algorithm's ability to correctly separate overlapping trees given an existing vessel segmentation (the focus of this work) rather than to also evaluate the vessel segmentation. Among the 48 images, on average, 42.98% vessel centerline pixels are labeled as A, 45.65% are V, 0.34% are O, and 11.03% are FP or U. The labeled vessel images generated by the algorithm however only indicate either A or V, thus FP or U are excluded from the evaluation. Fig. 3 shows an example of a whole VPCM with sub-VPCMs in different colors, the reference standard and the result of our approach.

The accuracy is calculated as the number of correct constructed centerline pixels versus all centerline pixels. The average and standard deviation for the cases including complex VPCMs are 0.84 and 0.09, and for only including cyclic VPCMs are 0.83 and 0.11. The running time per image (700×800) was within 2 minutes. The results indicate the high accuracy and reliable performance of our algorithm. Advantages of using our global optimization framework over a more local approach are particularly noticeable in cases with ambiguous local information (such as that indicated by the purple and green arrows in Fig. 3).
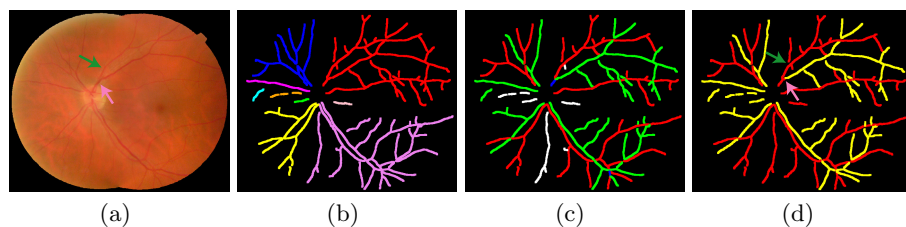
(a)   (b)   (c)   (d)

Fig. 3. Example result. (a) Fundus image. (b) Sub-VPCMs in different colors. (c) The reference standard (red for A, green for V, blue for O, white for FP or U). (d) The result indicated in two colors. Purple and green arrows = landmarks that may be mistakingly identified as a bifurcation and crossover, respectively, using a local approach.

4   Conclusion

In this paper we present a general framework to construct anatomical vasculature in retinal images combining local and global knowledge. Based on vessel segmentation, our framework repairs connectivity between segments and separates the vessel network into anatomical trees by using a graph-based algorithm. Our approach is tested on low-contrast fundus images and promising results indicate its reliable performance and potential applicability to similar problems.

References

1. Cheung, N., Donaghue, K.C., Liew, G., Rogers, S.L., Wang, J.J., Lim, S.W., Jenkins, A.J., Hsu, W., Lee, M.L., Wong, T.Y.: Quantitative assessment of early diabetic retinopathy using fractal analysis. Diabetes Care 32(1), 106–110 (2009)
2. Niemeijer, M., Staal, J., van Ginneken, B., Loog, M., Abramoff, M.D.: Comparative study of retinal vessel segmentation methods on a new publicly available database. In: Proc. SPIE Medical Imaging, vol. 5370, pp. 648–656 (2004)
3. Martinez-Perez, M.E., Highes, A.D., Stanton, A.V., Thorn, S.A., Chapman, N., Bharath, A.A., Parker, K.H.: Retinal vascular tree morphology: a semi-automatic quantification. IEEE Trans. Biomed. Eng. 49(8), 912–917 (2002)
4. Lin, K., Tsai, C., Sofka, M., Chen, S., Lin, W.: Retinal vascular tree reconstruction with anatomical realism. IEEE Trans. on Biomed. Eng. 59(12), 3337–3347 (2012)
5. Joshi, V.S., Garvin, M.K., Reinhardt, J.M., Abramoff, M.D.: Automated method for the identification and analysis of vascular tree structures in retinal vessel network. In: Proc. SPIE Medical Imaging, vol. 7963, p. 79630I (2011)
6. Rothaus, K., Jiang, X., Rhiem, P.: Separation of the retinal vascular graph in arteries and veins based upon structural knowledge. Image and Vision Computing 27(7), 864–875 (2009)
7. Joshi, V., Reinhardt, J.M., Abramoff, M.D.: Automated measurement of retinal blood vessel tortuosity. In: Proc. SPIE Medical Imaging, vol. 7624, p. 76243A (2010)
8. Kavitha, T., Liebchen, C., Mehlhorn, K., Michail, D., Rizzi, R., Ueckerdt, T., Zweig, K.A.: Cycle bases in graphs characterization, algorithms, complexity, and applications. Computer Science Review 3(4), 199–243 (2009)