Automated Tracing of Retinal Blood Vessels Using Graphical Models

Jaydeep De[1,2], Tengfei Ma[5], Huiqi Li[4], Manoranjan Dash[2], and Cheng Li[1,3]

[1] Bioinformatics Institute (BII), A*STAR, Singapore
[2] School of Computer Engineering, Nanyang Technological University, Singapore
[3] School of Computing, National University of Singapore, Singapore
[4] Beijing Institute of Technology, China
[5] Univeristy of Tokyo, Japan**

Abstract. As an early indication of diseases including diabetes, hypertension, and retinopathy of prematurity, structural study of retinal vessels becomes increasingly important. These studies have driven the need toward accurate and consistent tracing of retinal blood vessel tree structures from fundus images in an automated manner. In this paper we propose a two-step pipeline: First, the retinal vessels are segmented with the preference of preserving the skeleton network, i.e., retinal segmentation with a high recall. Second, a novel tracing algorithm is developed where the tracing problem is uniquely mapped to an inference problem in probabilistic graphical models. This enables the exploitation of well-developed inference toolkit in graphical models. The competitive performance of our method is verified on publicly available datasets comparing to the state-of-the-arts.

1 Introduction

Structural study of retinal vessels provides valuable information for clinicians for the prognosis of diseases such as diabetes, hypertension, retinopathy of prematurity (ROP) [1,2]. For example, the change of length to diameter ratio of an artery tree is related to diseases like hypertension and diabetes [3,4]. However, commercial softwares still largely rely on manual efforts for tracing the artery trees. This is a tedious work due to the highly variable structure of these retinal vessels in images. Moreover, it is not sustainable for high-throughput analysis of fundus images from patients.

There are two types of studies conducted in retinal vessel analysis. The first is based on segmentation [5–12], while the second is vessel tracking based methods [13–17]. The first type classifies the pixels into vessel or non-vessel pixels and produces a binary image where foreground pixels correspond to vessel pixels and background pixels correspond to non-vessel pixels. The second type starts with a seed provided by the user and tracks the vessel edges based on local intensity values. Vessel tracking based methods are advantageous as they tend to produce

** The work was carried out when the author was with BII as an intern student.

J.-K. Kämäräinen and M. Koskela (Eds.): SCIA 2013, LNCS 7944, pp. 277–289, 2013.

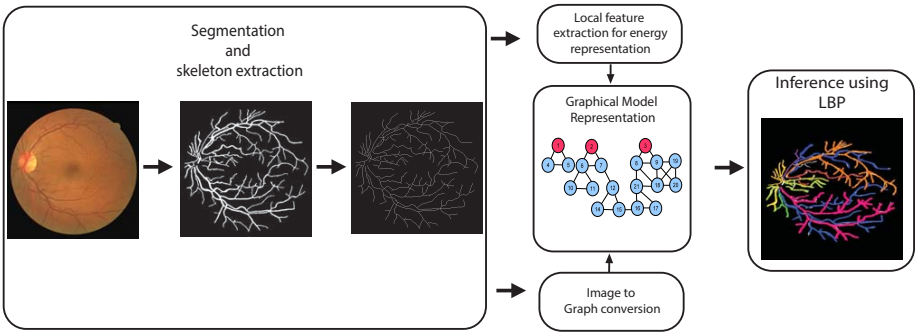

Fig. 1. Overview of the proposed tracing pipeline (Best view in color)

connected structures of vessel segments, whereas pixel classification methods often produce many disconnected or isolated segments. However, vessel tracking methods perform poorly in the crossover scenarios where they often get confused at the junction points and produce the wrong trees [17]. In this paper, a two-step automated pipeline is proposed to particularly address this problem, and promising results are obtained during evaluation on a standard benchmark. Our approach bridges segmentation algorithms and vessel tracking methods: Given an output from a segmentation algorithm our method can trace each tree from the others, to be used by the follow-up tree quantifications and analyses. The pipeline of our approach is illustrated in Fig. 1.

2 Segmentation

The goal of the segmentation step in our context is to extract the vessel skeleton while maintaining the structural connectivity, as well as the corresponding pointwise thickness along the skeleton, based on which the retinal vessels can be faithfully reconstructed. This differs notably from the usual aim of most existing segmentation work, where the emphasis is to achieve a high classification accuracy. As the number of vessel pixels are much fewer comparing to the number of background pixels, often a high accuracy is achieved by missing many vessel pixels – a situation we try to avoid. In fact, our goal can be better described as *segmentation with a high recall*. It is critical for us to retain the vessel pixels that keep the local vein and artery branches from being broken or entirely missing. To achieve this, based on two existing methods [9,11], our segmenter is formed by merging the results, while emphasizing on maintaining the network connectivity. Due to space concern, we discuss here only the quantitative analysis of the segmentation results.

To evaluate the segmentation performance, Table 1 compares to the state-of-the-art methods at the popular Drive dataset, while our segmenter excels in

picking up most vessel pixels (the recall column) – crucial for maintaining the connectivity of segmented parts, it also performs reasonably well in keeping only a small fraction of false alarms (the precision column) and thus a reasonable accuracy score – comparable with the leading methods.

Postprocess. The binary segmentation result is converted into a skeleton map (of one pixel thickness) by standard medial-axis transform. Meanwhile the optical disk region is identified and removed by applying a simple smoothing and thresholding step.

3 Tracing

As illustrated in Fig. 2, skeleton pixels are partitioned into three categories:

Body: Points (Pixels having two neighbors)
Terminal: Points (Pixels having one neighbors)
Branching: Points (Pixels having three or more neighbors)

In particular, the terminal points that are closer to the removed optical disk are called the *root* points, and the rest are *end* points.

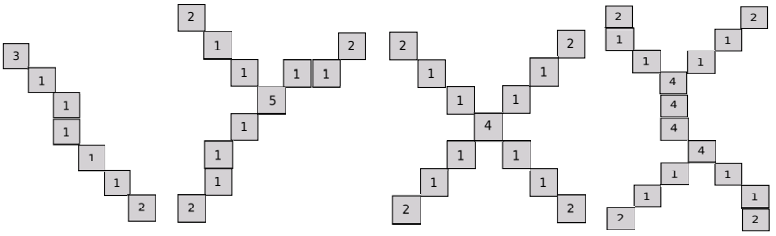

Fig. 2. Pixel labeling: (1)*Body points*, (2) & (3) *Terminal points*, and (4) & (5) *Branching points*

It is commonly assumed that for tracing, we always start from the optical disk where the root points of the retinal vessel skeleton are present. As a result, the retinal vessels can always be partitioned into a *disjoint* set of rooted trees, with each tree possessing a unique label stemming from its root point (and thus *the* segment it resides in). This is illustrated as the red-colored segments of the vessel skeleton in Fig. 3 top. Clearly the number of trees is always known at this stage. A *segment* is a continuous subset of skeleton pixels starts with a root or a branching point, and ends with a branching or an end point. We have counted the number of vessel segments in an image and label them with separate numbers(Fig. 3).The vessel segments which are connected to the root pixels are called *Root Segments*.

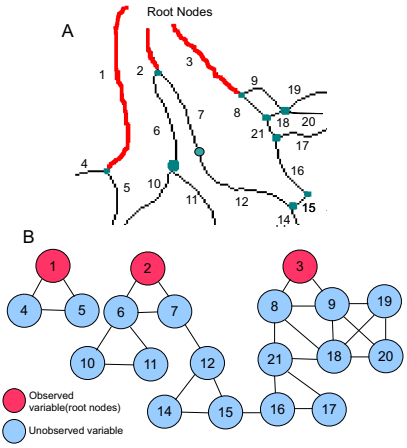

Fig. 3. Vessel skeleton to graph: (A) The vessel skeleton(Red one the root segment). (B) The Undirected Graph(Red ones are root/observed nodes, others are unobserved nodes).

3.1 From Image Skeleton to Graph Structure

The skeleton of the segmented image is converted into an undirected graph, $G = (V, E)$, where the nodes V are the vessel segments of the image and there is an edge between two nodes if the corresponding vessel segments are connected in the image. Fig. 3A-B provides an illustrative example on a small fraction of a retinal skeleton. Following the common assumption, the branch segments connected with the optical disk are regarded as the root nodes. If we consider each node in the graph as a random variable then the number of possible states for those random variable is equal to the number of root nodes in an image. As the root nodes are known, they are regarded as observed nodes (red nodes). So the problem is to label the remaining nodes (blue nodes) of the graph. Clearly it is an easy problem when the substructures are isolated from each other. It becomes increasingly difficult as there are more and more crossovers of these substructures in the original image skeleton.

3.2 Tracing as an Inference Problem

Once we have the graph structure, the tracing problem becomes that of an MAP inference problem in undirected graphical models, where we want to find the maximum probable label assignment for each of the node variables.

Formally, assume there are n number of nodes in the graph. $X = \{x_1, x_2, \ldots, x_k\}$ denotes the k root nodes which are observed, while $Y = \{y_1, y_2, \ldots, y_{n-k}\}$ denotes the unobserved nodes. The inference problem is formulated as

$$Y^* = \arg\max_{Y} \; p(Y|X), \tag{1}$$

where Y^* is the optimal assignment for the unobserved nodes. For undirected graphical models, following the celebrated Hammersley-Clifford theorem [18] and assuming binary interactions, the probability function possesses a factorized form, as

$$p(Y|X) = \frac{1}{Z} \prod_{(i,j)\in E} \psi_{ij}(y_i, y_j), \tag{2}$$

where $\psi_{ij}(y_i, y_j)$ is the local potential function defined over a pair of nodes connected by an edge, Z is the normalization constant. Consider $\psi_{ij}(y_i, y_j) = \exp\big(-E_{ij}(y_i, y_j)\big)$, where $E_{ij}(y_i, y_j)$ refers to the local pairwise energy function. By combining (1) (2), and by the fact that log is a monotone function, the problem can be reformulated as

$$Y^* = \arg\min_{Y} \sum_{(i,j)\in E} E_{ij}(y_i, y_j). \tag{3}$$

To solve it, the loopy belief propagation algorithm [18] is employed to infer the approximately optimal solution of (3).

3.3 Pairwise Energy E_{ij}

The pairwise energy $E_{ij}(y_i, y_j)$ can be equivalently described as a $k \times k$ lookup table, with an entry for the scenario where the random variables y_i and y_j take corresponding values. As an example and without loss of generality, in what follows we consider a Potts model [18] for $k = 3$, which gives a 3×3 lookup table as:

$$E_{ij}(y_i, y_j) \ \in \ e \times \begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \end{bmatrix}$$

In other words, we only look at whether the two nodes taking the same label or not, and encourage with $E = +e$ the case where they do take the same label and penalize with $E = -e$ when they take different labels.

The scalar e is defined as a function over $\theta \in [0, \pi)$, the angle or absolute difference between the local orientation for the two segments, i and j. To calculate the local orientation we take the skeleton of a particular segment and smooth it with a Gaussian filter, then we find the hessian matrix (2×2) of all the points on those skeleton pixels. The eigenvector corresponding to the largest eigenvalue gives me the orientation for each pixel. Then we have changed the orientation in such a way that it always points to the inward direction. After that we find the average orientation around the junction for few pixels. The functions are defined as follows

$$e = \begin{cases} f_1(\theta) & \text{if (i,j) is a part of a clique of 3 nodes.} \\ f_2(\theta) & \text{if (i,j) is a part of a clique of 4 or more nodes.} \end{cases} \tag{4}$$

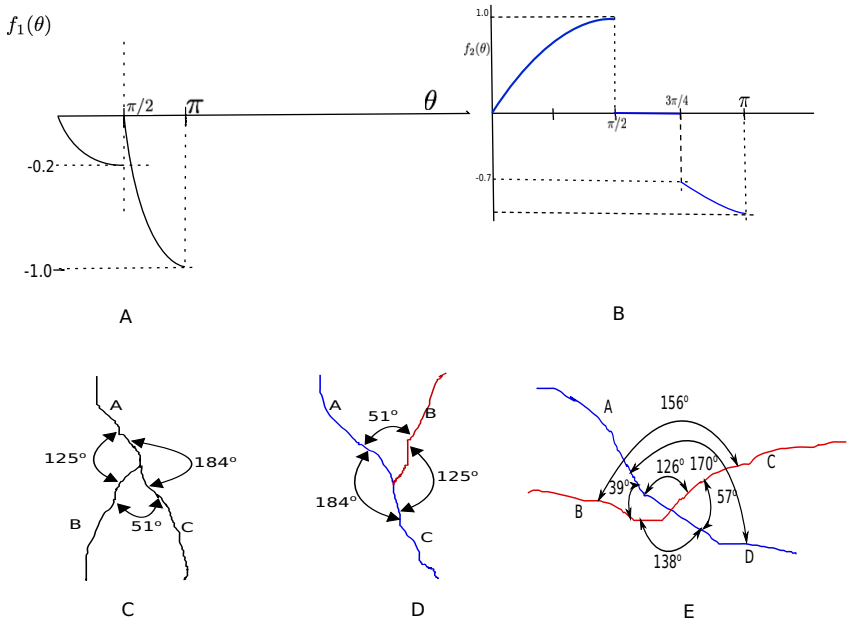

Fig. 4. Energy configuration for different scenarios

with

$$f_1(\theta) = \begin{cases} \cos(\theta) & \text{if } \pi > \theta > \pi/2 \\ \frac{-\sin(\theta)}{5} & \text{if } 0 < \theta < \pi/2 \end{cases} \tag{5}$$

$$f_2(\theta) = \begin{cases} \cos(\theta) & \text{if } \pi > \theta > 3\pi/4 \\ 0 & \text{if } \pi/2 < \theta < 3\pi/4 \\ \sin(\theta) & \text{if } 0 < \theta < \pi/2 \end{cases} \tag{6}$$

3 node occurs in branching(Fig. 4C) or when the end point of one branch connects on another branch(Fig. 4D). In both the scenarios if $\pi > \theta > \pi/2$ lower energy should be assigned to e. If $0 < \theta < \pi/2$ then there is a less chance that they belong to the same root so they should be assigned high energy compared to the previous case so we have taken the function $-sin(\theta)/5$. 4 node occurs in branching situation(Fig. 4E), where lower energy is assigned to higher angles and higher energy is assigned to lower angles(Fig. 4B).

3.4 Modified Loopy Belief Propagation

Our energy function is defined in such a manner that the diagonal elements are assigned one value and off diagonal elements are assigned another value. That means for a two state system (1,1) and (2,2) configuration will always get same energy and similarly (1,2) and (2,1) will also get same energy. This tie is broken by the observed nodes and their influence assigns higher probability to one of

the state. But in practical scenario very few nodes are known and some nodes are far away from the root nodes. So the influence of the root node gets decayed when they reach those far away nodes and they fail to break the tie and we get marginals for those nodes which are equiprobable for all the states. In literature this phenomena is called *Correlation Decay* problem [19, 20]. We have taken a practical approach for solving this problem.

At first we run the loopy belief propagation over the entire graph and calculate the marginals for all the nodes. Then we find those far away nodes which have equiprobable marginals and also find out those are non-equiprobable. Then we find out those non-equiprobable nodes which are directly connected with the equiprobable nodes. We re-initiate the belief propagation algorithm with non-equiprobable nodes as observed nodes and equiprobable nodes as unobserved nodes. We repeat this until we get non-equiprobable marginal over all nodes.

4 Experiments

Experiments are conducted on the publicly available DRIVE [21] dataset. It contains 40 color retinal fundus images divided into training and testing set. All images come with manual annotations from domain experts. The performance evaluation is undertaken by two scores, namely segmentation accuracy and tracing accuracy, as described below.

4.1 Evaluation of the Segmentation Step

Quantitatively our algorithm is clearly the *best* performer in terms of *recall* and *F1* score and is the *second best* in terms of precision. Our accuracy is also on par with the state-of-the-art. Visually our algorithm performs significantly better than the existing ones in term of preserving the connectivity among segments, which is crucial for tracing, as displayed in e.g. Fig. 5.

Table 1. Comparison of Vessel Segmentation performance in DRIVE (Red – Best Performer, Blue – Second Best Performer)

Method	Precision	Recall	F-1	Accuracy
Ricci [22]	-	-	-	.9563
Mendonca [5]	.7315	-	-	.9463
Peter Bankhead [11]	.7027	.7177	.7101	.9371
Garg [6]	-	-	-	.9361
Espona [7]	.7436	-	-	.9352
Martinez-Perez [8]	.7246	-	-	.9344
Diego Marin [10]	.8433	.7067	.7690	.9452
Soares et. al. [9]	.6943	.7425	.7176	.9466
Our method	.7602	.8336	.7952	.9429

4.2 Evaluation of the Tracing Step

To evaluate the performance of our tracing step, we manually label each segments and assigned them to one of the trees. Then we compare the labels assigned by our tracing algorithm to the manual labeling: If they are same then we take that segments as correctly traced otherwise wrongly traced. Table 2 presents the performance of our tracing algorithm. Note that the accuracy of 66.86% is not trivial, as for a typical network with 10 trees (i.e. 10 classes), a trivial classifier that randomly assigns labels will achieve merely 10%.

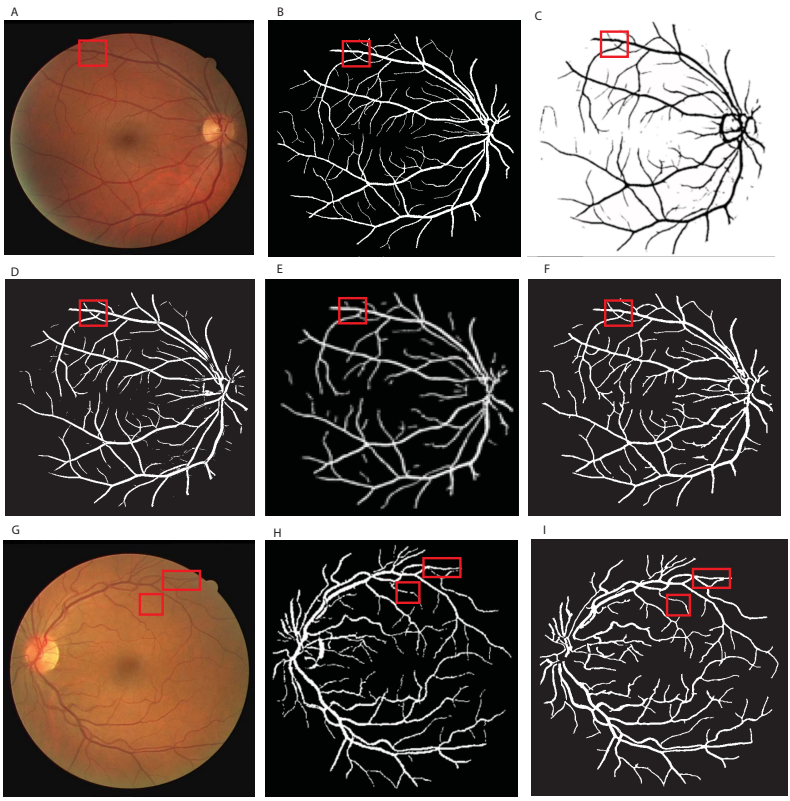

Fig. 5. Visual comparison of segmentation result (A) Original image (DRIVE-Test-Image-19), and its ground-truth segmentation in (B). (C) Segmentation result of [22]. (D) Segmentation result of [9] (E) Segmentation result of [10]. (F) Segmentation result of our method. (G) Original image (DRIVE-Test-Image-01). (H) Segmentation result of [8]. (I) Segmentation result of our method. If we pay close attention to the boxed regions, we can see that our segmentation step does a much better job at retaining the connectivity of the skeleton.

Table 2. Performance of the Tracing Step on DRIVE

No. of Segments	Correctly traced segments	Accuracy (%)
3529	2369	67.12

The visual tracing results are presented in Fig. 7 for an example with detailed discussion, as well as in Fig. 8 for two additional examples. As shown by the green boxes, the crossover cases are quite nicely resolved by the proposed algorithm. The tracing errors (red circles) can be roughly categorized to *four* main types: (1) Those due to issues in skeleton extraction, where e.g. 4-cliques are wrongly encoded into two 3-cliques (Type-A error) as illustrated in Fig 6; (2) Errors carried over from segmentation, which results in wrong local topology in the graph representation (Type-B error); (3) Errors due to undirected message-propagation (Type-C error); And (4) errors due to the limitation of the energy functions adopted here (Type-D error). Fig 7 provides more details of the suc-

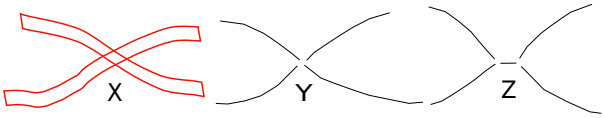

Fig. 6. (X) Crossover situation (Y) Correct skeleton (Z) Wrong skeleton

cess and failure scenarios of our approach: The left image presents the original image overlapped with predicted skeleton, while the right image delineates our tracing result. Therefore, a correct prediction (only one is shown) consists of a *pair* of green boxes – one from input and one from tracing result; A tracing mistake (E1-E5), on the other hand, will be denoted as a *triplet* of two red circles and a blue circle – the first two are from the input and the wrong tracing result, respectively, and the third brings its corresponding tracing ground-truth. The five errors (E1-E5) is discussed in details below:

E1. This is a type-D error. The angle between segments a and d is less than the angle between b and d. Our energy function thus assigns less value to pair (b,d) than (a,d). As a consequence, d is wrongly assigned the same label as of b.

E2. This is due to segmentation error (Type-B), which affects its skeleton map and follow-up graph structure by a local connection (topology) error. and leads to wrong tracing result.

E3. The tracing error is due to a combination of two reasons: type-C and type-D. The angles between pair (f,d),(g,i),(k,i) and (g,h) are all more than $\frac{\pi}{2}$. so they are all assigned to low energy values, which in turn encourages them to have the same class label. The incorrect predictions of nodes (d,h,e,k,j)

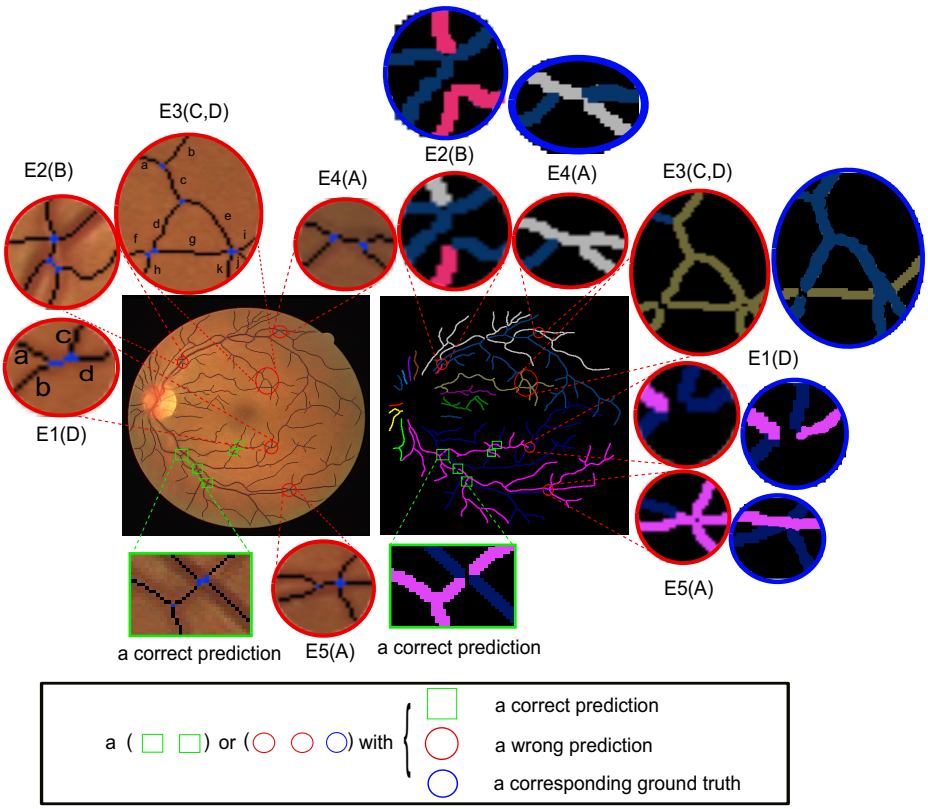

Fig. 7. An example of an input image (left) and its tracing result (right). A green box denotes correct tracing result, a red circle points out an error (denoted as E1-E5 for five separate errors. Meanwhile, A-D in bracket stand for four major error types), and a blue circle is for the corresponding ground truth. For each box or circle, a zoom-in view is provided. In particular, for each red circle error, its corresponding ground-truth is also provided in blue circle. See text for details.

influence other nodes like c and b. So they are also assigned to wrong labels. In contrast, as shown in blue circle for the corresponding ground-truth, the segments actually belong to two different classes.

E4. This is a type-A error where one 4-clique is wrongly treated as two 3-cliques.

E5. This is also a type-A error where one 4-clique is treated as two 3-cliques.

One observation from analyzing the examples is that our algorithm works best when the errors from segmentation step stay low, thus the topological connections remain mostly correct. We would like to address this over-sensitivity issue for future work.

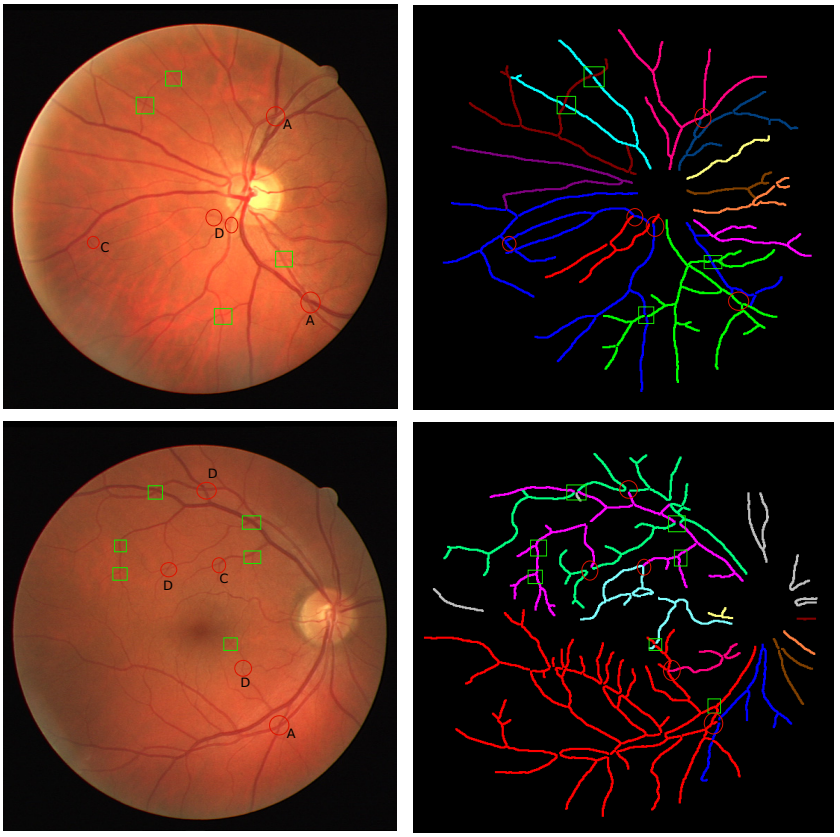

Fig. 8. Two more exemplar tracing results: Left column: Original retinal images; Right column: Our tracing results. Green boxes denote the correct tracing results, while red circles point out the errors.

5 Conclusion and Outlook

In this paper we have taken a new approach for tracing vessel like structures. Instead of tracing by looking at the image intensity level, we transform the tracing problem into an equivalent global inference problem in Graphical models. Our algorithm is shown to achieve good performance in crossover scenarios. As future directions we are working on learning the energy functions which would help to resolve more complex crossover situations, as well as exploiting ways to make the tracing step more robust to small perturbation from the preceding segmentation step.

Acknowledgements. This research was supported by an A*STAR JCO Project grant (1231BFG040), and an A*STAR JCO Career Development Award grant (12302FG010). We would like to thank Dr. Weimiao Yu for discussions, and thank Dr. Sohail Ahmed and Dr. Hwee Kuan Lee for their support.

References

1. Sun, C., Wang, J., Mackey, D., Wong, T.: Retinal vascular caliber: Systemic, environmental, and genetic associations. Survey of Ophthalmology 54(1), 74–95 (2009)
2. Wang, J., Liew, G., Klein, R., Rochtchina, E., Knudtson, M., Klein, B., Wong, T., Burlutsky, G., Mitchell, P.: Retinal vessel diameter and cardiovascular mortality: pooled data analysis from two older populations. European Heart Journal 28(16), 1984–1992 (2007)
3. Martinez-Perez, M., Hughes, A., Stanton, A.H., Thom, S., Chapman, N., Bharath, A., Parker, K.: Retinal vascular tree morphology: a semi-automatic quantification. IEEE Trans. Biomed. Eng. 49(8), 912–917 (2002)
4. King, L., Stanton, A., Sever, P., Thom, S., Hughes, A.: Arteriolar length-diameter (l:d) ratio: a geometric parameter of the retinal vasculature diagnostic of hypertension. J. Hum. Hypertens. 10(6), 417–424 (1996)
5. Mendonca, A., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Trans. Med. Imag. 25(9), 1200–1213 (2006)
6. Garg, S., Sivaswamy, J., Chandra, S.: Unsupervised curvature-based retinal vessel segmentation. In: International Symposium on Biomedical Imaging, pp. 1200–1213 (2007)
7. Espona, L., Carreira, M., Penedo, M., Ortega, M.: Retinal vessel tree segmentation using a deformable contour model. In: International Conference on Pattern Recognition (2008)
8. Martinez-Perez, M., Hughes, A., Thom, S., Bharath, A., Parker, K.: Segmentation of blood vessels from red-free and uorescein retinal images. Med. Image Anal. 11(1), 47–61 (2007)
9. Soares, J., Leandro, J., Cesar, R., Jelinek, H., Cree, M.: Retinal vessel segmentation using the 2d gabor wavelet and supervised classification. IEEE Trans. Med. Imag. 25(9), 1214–1222 (2007)
10. Marin, D., Aquino, A., Gegundez-Arias, M., Brav, J.: A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. IEEE Trans. Med. Imag. 30(1), 146–158 (2011)
11. Bankhead, P., Scholfield, C., McGeown, J., Curtis, T.: Fast retinal vessel detection and measurement using wavelets and edge location refinement. PLoS ONE 7(3), e32435 (2012)
12. Wang, L., Bhalerao, A.: Model based segmentation for retinal fundus images. In: Scandinavian Conference on Image Analysis, pp. 422–429 (2003)
13. Xu, X., Niemeijer, M., Song, Q., Sonka, M., Garvin, M., Reinhardt, J., Abrãmoff, M.: Vessel boundary delineation on fundus images using graph-based approach, pp. 1184–1191 (2011)
14. Can, A., Shen, H., Turner, J., Tanenbaum, H., Roysam, B., Roysam, B.: Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms, pp. 125–138 (1999)
15. Grisan, E., Pesce, A., Giani, A., Foracchia, M., Ruggeri, A.: A new tracking system for the robust extraction of retinal vessel structure. IEEE Engineering in Medicine and Biology Society 1, 1620–1623 (2004)
16. Bekkers, E., Duits, R., Romeny, B., Berendschot, T.: A new retinal vessel tracking method based on orientation scores. CoRR abs/1212.3530 (2012)
17. Al-Diri, B., Hunter, A., Steel, D.: An active contour model for segmenting and measuring retinal vessels. IEEE Transactions on Medical Imaging 28(9), 1488–1497 (2009)

18. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
19. Gamarnik, D., Katz, D.: Correlation decay and deterministic fptas for counting colorings of a graph. J. Discrete Algorithms 12, 29–47 (2012)
20. Gamarnik, D., Goldberg, D., Weber, T.: Correlation decay in random decision networks. CoRR abs/0912.0338 (2009)
21. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Trans. Med. Imag. 23(4), 501–509 (2004)
22. Ricci, E., Perfetti, R.: Retinal blood vessel segmentation using line operators and support vector classification. IEEE Trans. Med. Imag. 26(10), 1357–1365 (2007)