

Multi-resolution Vessel Segmentation Using Normalized Cuts in Retinal Images

Wenchao Cai and Albert C.S. Chung

Lo Kwee-Seong Medical Image Analysis Laboratory,
Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Hong Kong
{`wenchao`, `achung`}@`cse.ust.hk`

Abstract. Retinal vessel segmentation is an essential step of the diagnoses of various eye diseases. In this paper, we propose an automatic, efficient and unsupervised method based on gradient matrix, the normalized cut criterion and tracking strategy. Making use of the gradient matrix of the Lucas-Kanade equation, which consists of only the first order derivatives, the proposed method can detect a candidate window where a vessel possibly exists. The normalized cut criterion, which measures both the similarity within groups and the dissimilarity between groups, is used to search a local intensity threshold to segment the vessel in a candidate window. The tracking strategy makes it possible to extract thin vessels without being corrupted by noise. Using a multi-resolution segmentation scheme, vessels with different widths can be segmented at different resolutions, although the window size is fixed. Our method is tested on a public database. It is demonstrated to be efficient and insensitive to initial parameters.

1 Introduction

The analysis of retinal images, also called fundus images, plays an important role in the diagnoses of various eye diseases. Useful features of color retinal images are extracted and analyzed to help the ophthalmologists, e.g. optic disk, exudates, the structure and widths of vessels. In this paper, we focus on the vessel segmentation in retinal images. Vessel segmentation is very important in various medical diagnoses. Many algorithms for vessel segmentation in 2-D and 3-D data have been proposed. In a broad term, these algorithms can be divided into two categories: those utilize some filters to extract boundary or ridge of the vessels, followed by further refinement (e.g. [1,2,3,4]); and those employ tracking strategy by some given or detected seeds in the vessels (e.g. [5,6,7]).

The algorithm presented by Chaudhuri *et al.* [2] was based on directional 2-D matched filter that assumed a Gaussian vessel cross-sectional profile and small vessel radius variations. Hoover *et al.* [1] improved this method by a region-based threshold probing of the matched filter response. Jiang and Mojon [8] directly applied multi-threshold probing to the image through a verification procedure which made use of a curvilinear structure model. Koller *et al.* [9] introduced a nonlinear multiscale line filter based on the second derivative of Gaussian

function. This filter could be used to detect lines of various widths in 2-D and 3-D image data. Similarly, some algorithms ([3,10,11,12]) used the Hessian matrix to extract the local line features on different scales of 3-D volume. Different from these algorithms based on filters, Tolia and Panas [5] developed a fuzzy C-means clustering algorithm. They started the fuzzy tracking algorithm from the optic located by the algorithm. Lalonde *et al.* [6] tracked a pair of points on the boundary of vessel in the edge map and Cree *et al.* [7] tracked the vessel by fitting a 2-D physical model.

The methods above for vessel segmentation can work well to extract the major parts of the vasculature. For the thinner parts, however, it is still an open problem because the image contrast around thin vessels is generally low. Recently, focusing on 2-D retinal images, several supervised methods [13,14,15] have been explored to get better results. At every pixel, they extracted a feature vector in the neighborhood. The vector was classified by a kNN-classifier [13,14] or a Bayesian classifier with class-conditional probability density function [15].

It can be labor-intensive to segment the training images manually. The process of training needs to be repeated in different situations. Thus we propose an unsupervised method in this paper. The proposed algorithm employs the gradient matrix of the Lucas-Kanade equation [16] to evaluate a local window, and determines whether there is a vessel in the window by eigen-decomposing the gradient matrix. Unlike the methods using the Hessian matrix, since the proposed algorithm avoids computing the second derivative, it can detect thinner vessels in a window. For every candidate window, which is a region containing a vessel, the algorithm searches a intensity threshold by minimizing the normalized cut criterion [17]. In order to segment thinner vessels and avoid selecting noisy candidate windows, the algorithm utilizes a tracking strategy. This makes it possible to set strict parameters for pre-selection and relax them dynamically. For vessels with different widths, the algorithm generates the Gaussian pyramid for an input image and segment vessels with different widths at different resolution levels. It is shown that the experimental results on a public database, DRIVE [13], are more accurate than one of the widely used unsupervised methods [8], and comparable with two supervised methods [13,15].

2 Proposed Method

The overview of the proposed method is described as follows:

1. Compute the Gaussian pyramid of the input image;
2. For images at different resolution levels, compute the gradient vector for every pixel by the Sobel operator;
3. Scan the image by a sliding window of fixed size, compute the gradient matrix of the window and evaluate it by eigen-decomposing the matrix (Section 2.1);
4. Select the window whose eigenvalues satisfy some conditions as an candidate window, and search a local intensity threshold in the candidate window by which the binary segmentation with the smallest $NCut$ in Eq. 3 produces the blood vessel (Section 2.2);

5. If a vessel is detected, track the vessel along the direction generated from the eigenvector of gradient matrix and relax parameters to select the next candidate window (Section 2.3); otherwise, continue scanning;
6. Combine all the results obtained from different resolution levels and refine the final result.

The details are explained in the following subsections.

2.1 Gradient Matrix

The gradient matrix originates from the problem of optical flow. Eq. 1 is the basic optical flow equation

$$\nabla I \cdot [u \ v]^T + I_t = 0. \tag{1}$$

Considering every pixel p in a small window under the assumption that the flow field is smooth locally, Lucas and Kanade [16] proposed a least square method to obtain the solution $[u, v]$ as follows:

$$\begin{bmatrix} \sum_p I_x I_x & \sum_p I_x I_y \\ \sum_p I_y I_x & \sum_p I_y I_y \end{bmatrix} \cdot [u \ v]^T = - \begin{bmatrix} \sum_p I_x I_t \\ \sum_p I_y I_t \end{bmatrix}. \tag{2}$$

The left hand side of Eq. 2 is called gradient matrix in this paper, denoted by G . The gradient matrix G contains the texture information of the window. Table 1 shows the relationship between local features and two eigenvalues of G , λ_1 and λ_2 ($\lambda_1 \geq \lambda_2$).

Table 1. Relationship between local features and G

| Local Feature | λ_1 | λ_2 | $ \Sigma \nabla I $ |
|---------------|-------------|-------------|---------------------|
| High textured | Large | Large | – |
| Low textured | Small | Small | Small |
| Edge | Large | Small | Large |
| Line | Large | Small | Small |

In Table 1, we see that the type of local feature can be determined by the values of λ_1 and λ_1/λ_2 . However, from the two eigenvalues, we cannot distinguish between an line and an edge. It is observed that, although both of them produce the gradient matrices with large λ_1 and small λ_2 , all the gradients along an edge are in the same orientation, and the gradients along a line are in two inverse orientations. For a line structure in the window, the magnitude of the sum of gradients ($|\Sigma \nabla I|$) will be small. Therefore, the line structure can be detected by looking at the value of ($|\Sigma \nabla I|/\max(|\nabla I|)$), where $\max(|\nabla I|)$ is the maximal magnitude of the gradient in the window. It is used to normalize $|\Sigma \nabla I|$ in regions with different contrast. For example, in Figure 1, we consider three 7×7 windows. From top to bottom, there is a junction in the first window, so it is a high texture region with a large $\lambda_1 = 64799.6$ and a small $\lambda_1/\lambda_2 = 2.4$. There is a vessel (a

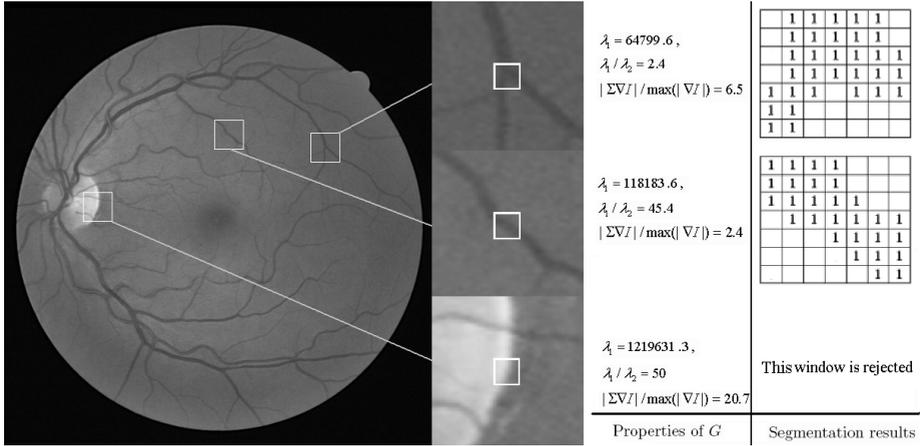


Fig. 1. Illustration of selecting candidate windows and segmentation in candidate windows. The left image is the green channel of an original image; the middle images show the corresponding zoom-in white windows in left image; the right column shows the eigenvalues of the gradient matrices of the corresponding 7×7 white windows and the segmentation results. The third window with an edge is rejected due to a large value of $|\Sigma \nabla I| / \max(|\nabla I|)$.

line) crossing the second window, so $\lambda_1 = 118183.6$ is large and $\lambda_1 / \lambda_2 = 45.4$ means that λ_2 is relatively small. There is an edge in the third window, $\lambda_1 = 1219631.3$ is also very large and $\lambda_1 / \lambda_2 = 50$ also indicates a relatively small λ_2 . However, the values of $|\Sigma \nabla I| / \max(|\nabla I|)$ distinguish the second (with 2.4) and the third (with 20.7) windows. Therefore, according to these three values, the first and second windows are selected as candidate windows, while the third window is rejected.

2.2 Normalized Cut

After selecting the candidate windows by the gradient matrix, the algorithm needs to find an efficient way to segment the vessel. In our work, we introduce normalized cut criterion to search the best intensity threshold for segmentation.

The normalized cut (NCut) was proposed by Shi and Malik [17] based on the graph theory and spectral clustering. For *two-way cut* of a graph, the normalized cut criterion can be written as:

$$NCut(A, B) = \sum_{p \in A, q \in B} w_{pq} \left(\frac{1}{\sum_{p \in A} D_p} + \frac{1}{\sum_{p \in B} D_p} \right), \tag{3}$$

where A, B are two segments of the original set V , w_{pq} is the similarity between two vertexes p and q according to intensity and distance, $D_p = \sum_{q \in V} w_{pq}$ is defined as the degree of the vertex p . From Eq. 3, we can see that the normalized cut measures the total dissimilarity between the different groups, as well as the

total similarity within groups, which can help to search the best threshold to segment the vessel in a given window.

Although the problem of minimizing $NCut$ is NP hard [17], ignoring the discrete constraint, the authors presented an approximation solution by eigen-decomposing. Because computation of eigen-decomposing is inefficient, we only use Eq. 3 as a criterion to search an intensity threshold for every candidate window which produces the lowest $NCut$. Although it is an approximation of the standard normalized cut, it works more efficiently and our experiments show that most of the vessels can be well segmented by a local intensity threshold in candidate windows.

Based on Eq. 3, the $NCut$ tends to partition the window into two parts with similar size. Therefore, we assume the width of a vessel is around 2 – 3 pixels, and set the size of the sliding window 7×7 to suit the property of normalized cut criterion. Because the size of candidate windows (7×7) is too small to show the segmentation image, the segmentation result of the first and second window in Figure 1 are shown by a binary 7×7 matrix in the right column of Figure 1.

2.3 Vessel Tracking and Multi-resolution

Based on the two steps described above, the algorithm can segment the vessels from an image. In order to obtain more accurate segmentation result, we have to set appropriate parameters to select candidate windows. Although loose parameters can help to select the windows with thinner vessels, they also introduce more noisy windows, which greatly increase the computation load and segmentation error. To be insensitive to initial parameters, our method utilizes tracking strategy to solve this problem in our work. Since the first eigenvector of the gradient matrix indicates the normal direction of a vessel in the corresponding window, the method can track along the tangent direction of the vessel. Meanwhile, the parameters are relaxed to make it easy to select the next tracked window.

As mentioned in Section 2.2, we fix the size of sliding window to make full use of the advantage of the $NCut$. When the width of vessel is around 2 – 3 pixels, the proposed method will work best. Considering the variation of widths, we construct the Gaussian pyramid for the input image, and thus vessels with different widths will be segmented in different resolution levels. Finally, the algorithm recovers the results from images of different resolutions to the original resolution. The pixels segmented out in any resolution level are combined as the final result.

3 Experiments and Results

3.1 Data and Evaluation

Our method was tested on a public database of retinal images, called DRIVE [13]. In total, there are 40 color images with 565×584 pixels. All the images are divided into two groups: 20 images segmented by only one observer for training and the other 20 images segmented by two observers for testing. More details of

the database can be found in [13]. In our experiments, we took the segmentation of the first observer as the ground truth, and the second is for reference. We only showed the results of the testing set to compare with other methods. As mentioned in [6], since the green channel provides the highest contrast of the image, we only used the green channel of the color images for testing.

To evaluate the performance, we employed the receiver operating characteristic (ROC) curves, which show the plots of true positive fraction T_p versus false positive fraction F_p . The true positive and false positive fractions are defined as follows:

$$T_p = \frac{N_{tp}}{N_{vessel}}, F_p = \frac{N_{fp}}{N_{non-vessel}}, \quad (4)$$

where N_{tp} , N_{fp} are the numbers of true positives and false positives respectively; N_{vessel} , $N_{non-vessel}$ are the total numbers of vessel and non-vessel pixels in the ground truth. The DRIVE database provides mask images for all the images. So that only pixels inside FOV were considered in the testing.

3.2 Parameters Setting

In our experiments, the parameters of the algorithm were set as follows. For normalized cut, we compute the similarity w_{ij} between two pixels i and j :

$$w_{ij} = e^{-\frac{\Delta I(i,j)^2}{\sigma_1^2}} e^{-\frac{D(i,j)^2}{\sigma_2^2}}, \quad (5)$$

where $\Delta I(i, j)$ is the difference of intensity; $D(i, j)$ is the distance between i and j ; $\sigma_1^2 = \delta_I^2 * 0.3$ (δ_I is the maximal variation of intensity in current window), and $\sigma_2^2 = 49$.

A window will be selected as a candidate if the conditions are satisfied: $\lambda_1 > T_1, \lambda_1/\lambda_2 > T_2, |\Sigma \nabla I| / \max(|\nabla I|) < T_3$. where T_1 , T_2 and T_3 are parameters for selecting candidate windows. Based on our experiments, we made a tradeoff between all the three parameters by using only one variable k : $T_1 = 12000 - 1000k$, $T_2 = 2.5 - 0.1k$, $T_3 = 2.0 + 0.7k$. We changed k from 0 to 10 to generate the ROC curves. For different resolution levels, T_1 and T_2 are multiplied by a factor $n + 2$, n is the level of resolution. In the experiments, we use three resolution levels, a higher resolution ($n = 0$), the original resolution ($n = 1$) and a lower resolution ($n = 2$). In lower resolution, the vessels to be segmented corresponds to the thick vessels in the original image. In general, the image contrast around these thick vessels is high, and it is easy to segment them. Therefore, the parameters were directly fixed in the lower resolution with $T_1 = 60000$, $T_2 = 2.5$, $T_3 = 4$ in all situations. In the tracking process, T_1 , T_2 and T_3 are multiplied by factors of 0.7, 0.7 and 1.5 respectively.

3.3 Results

Because the performances of some existing methods on the database have been reported in [14] and [15], we directly compared our results with the reported results. The ROC curves of all the methods are plotted in Figure 2. In [13]

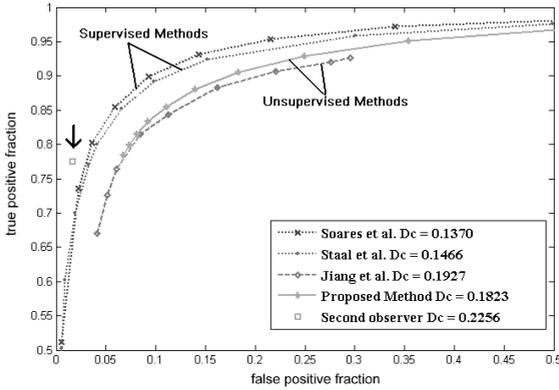


Fig. 2. The ROC curves and D_c of four methods on the DRIVE

and [15], they computed the area A_z under the ROC curve to evaluate the performance of the algorithms. However, we only plotted part of ROC with $F_p < 0.5$, and compute the distance D_c between the ROC and the ideal point $(0, 1)$ instead of A_z . The closer to the ideal point, the better the algorithm is. There are two reasons. First, unlike the supervised method, our method cannot control only one parameter to threshold all the pixels. This means our curves hardly reach left bottom and upper right corners. Second, we also think it is no need to compare A_z , because the result is meaningless when F_p is too large (there will be more false positives than true positives). We pay more attention to the area around the point of segmentation by the second observer, as pointed out by the arrows in Figure 2. Although we don't have the accurate testing results of other methods, the values of D_c still can distinguish the performances of all the methods (see the legend of Figure 2).

First of all, we have to point out that the methods of Staal *et al.* and Soares *et al.* are supervised. Especially, they did leave-one-out experiments on STARE database. It is natural that their results are better than other unsupervised methods. In Figure 2, although the curves of the proposed method are below the curves of two training methods, they are closer to the ideal point $(0, 1)$ and higher than the curves of Jiang *et al.*

Besides that, the ROC curves show another feature of the method. Every marker on the ROC of the proposed method stands for an average evaluation of a set of parameters (the markers on the other curves are the points we sample from the reported results). We notice that, although we linearly adjust the parameters by the same step, they are very compact on the left part of the curve which is in the region close to the second observer and the ideal point $(0, 1)$. That is because the algorithm uses the tracking strategy which makes the algorithm not so sensitive to initial parameters.

The experiments also show the method is efficient. Although our implementation is experimental, the proposed method can segment an image of DRIVE database in about 30s (Pentium-IV 1.5GHz), which is similar with the method of

Jiang *et al.*, but much faster than the supervised method in [13] (about 15min, Pentium-III 1.0GHz). There is a large space to speed up our method, because most of the major parts of vessels are segmented in all three resolution levels. We can combine the process of segmentation in different resolution levels together to avoid the redundant computation.

4 Conclusion

In this paper, we have proposed an unsupervised method based on normalized cut criterion and the gradient matrix to segment vessels in retinal images. The performances of the proposed method and some existing methods have been evaluated using the ROC curves. The evaluation demonstrates our method can improve the segmentation accuracy, as compared with the other widely used unsupervised method. The proposed method is computationally efficient.

References

1. Hoover, A., Kouznetsova, V., Goldbaum, M.H.: Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response. *TMI* **19** (2000) 203–210
2. Chaudhuri, S., Chatterjee, S., *et al.*: Detection of blood vessels in retinal images using two-dimensional matched filters. *TMI* (1989) 263–269
3. Krissian, K., Malandain, G., *et al.*: Model-based multiscale detection of 3d vessels. In: *CVPR*. (1998) 722–727
4. Lowell, J., Hunter, A., *et al.*: Measurement of retinal vessel widths from fundus images based on 2-d modeling. *TMI* **23** (2004) 1196–1204
5. Tolia, Y.A., Panas, S.M.: A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *TMI* **17** (1998) 263–273
6. Lalonde, M., Gagnon, L., Boucher, M.C.: Non-recursive paired tracking for vessel extraction from retinal images. In: *CVI 2000*. (2000) 61–68
7. Cree, M., Cornforth, D.J., H.F., J.: Vessel segmentation and tracking using a two-dimensional model. In: *IVC New Zealand*. (2005) 345–350
8. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multi-threshold probing with application to vessel detection in retinal images. *PAMI* **25** (2003) 131–137
9. Koller, T., Gerig, G., *et al.*: Multiscale detection of curvilinear structures in 2d and 3d image data. In: *ICCV*. (1995) 864–869
10. Lorenz, C., Carlsen, I.C., *et al.*: Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2d and 3d medical images. In: *CVRMed*. (1997) 233–242
11. Sato, Y., Nakajima, S., *et al.*: 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In: *CVRMed*. (1997) 213–222
12. Chen, J., Amini, A.A.: Quantifying 3-d vascular structures in mra images using hybrid pde and geometric deformable models. *TMI* **23** (2004) 1251–1262
13. Staal, J., Abràmoff, M.D., *et al.*: Ridge-based vessel segmentation in color images of the retina. *TMI* **23** (2004) 501–509

14. Niemeijer, M., Staal, J.J., *et al.*: Comparative study of retinal vessel segmentation methods on a new publicly available database. Volume 5370., SPIE (2004) 648–656
15. Soares, J.V.B., Leandro, J.J.G., *et al.*: Using the 2-D Morlet wavelet with supervised classification for retinal vessel segmentation. In: 18th Brazilian Symposium on Computer Graphics and Image Processing, Natal, RN (2005)
16. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI. (1981) 674–679
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22** (2000) 888–905