2010 11th Int. Conf. Control, Automation, Robotics and Vision
Singapore, 7-10th December 2010

Retinal Blood Vessel Segmentation via Graph Cut

Ana G. Salazar-Gonzalez, Yongmin Li and Xiaohui Liu
Department of Information Systems and Computing.
Brunel University, United Kingdom

***Abstract*—Image analysis is becoming increasingly prominent as a non intrusive diagnosis in modern ophthalmology. Blood vessel morphology is an important indicator for diseases like diabetes, hypertension and retinopathy. This paper presents an automated and unsupervised method for retinal blood vessels segmentation using the graph cut technique. The graph is constructed using a rough segmentation from a pre-processed image together with spatial pixel connection. The proposed method was tested on two public datasets and compared with other methods. Experimental results show that this method outperforms other unsupervised methods and demonstrate the competitiveness with supervised methods.**

***Index Terms*—Retinal images, vessel segmentation, graph cut.**

I. Introduction

The retinal blood vessel morphology is an important indicator for diseases like diabetes, hypertension and retinopathy. An inspection of the retinal vascular system represents an effective approach to detect the symptoms of the above diseases. Due to the non invasive characteristics, fundus image analysis has been of great interest [1].

Blood vessels can be seen as thin elongated structures in the retina, with variation in width and length. Different retinal images datasets have been collected under different conditions of illumination, resolution, field of view (FOV), etc.

Frequently vascular retinal segmentation methods are designed for a specific dataset, or parameters are adjusted to adapt them for different imaging conditions. It is important to have a segmentation algorithm that does not depend on parameter configuration, so the analysis can be unified for images of different datasets.

Structured filtering is a common approach for the retinal image analysis. The method presented in [2] uses an adaptive local thresholding based on a multithreshold probing scheme. The method presented in [3] utilizes a matched filter response, followed by a threshold probing. Wu [4] proposed an adaptive detection method which consists of three stages, enhancement using an extended adaptive histogram equalization, feature extraction through Gabor filter responses with different orientations, and tracing of the vascular networks for postprocessing. Mendonca [5] addressed the segmentation problem by detecting the vessel centerline first, followed by a vessel filling using global intensity characteristics and local vessel width information. Usman presents in [6] an enhancing preprocessing stage applying a Gabor Wavelet transform, follow by a histogram thresholding.

Supervised methods are effective models but their need for training with manual labeling makes them labor intensive. In [7] image ridges are used to form line elements, which are used to divide the image into patches. Pixel features are extracted using this representation. Various features are presented, and those that offer the best classification are selected. A segmentation process based on feature vector classification is presented in [8]. The feature vector is composed of pixel intensity, and two dimensional Gabor wavelet transform responses at multiple scales, followed by a Bayessian classifier with class conditional probability. In [9] the blood vessel characteristics are obtained at different scale, using image derivatives. These features are used in a growing procedure thereafter.

The normalized cut technique is used for the blood vessel segmentation in [10]. It presents an unsupervised method, using a gradient matrix to select a candidate window which may contain blood vessels. The normalized cut is utilized to segment the blood vessels on the selected window. Vessel tracking is used for postprocessing to improve results.

In this paper, we present a method for automatic retinal vascular network segmentation using the graph cut technique. Our approach takes as a first step the enhancement of blood vessels, followed by a rough segmentation, where prior information together with spatial pixel connection is used to construct the graph to obtain the final segmentation. The parameters of our method do not need to be adjusted for different image characteristics.

The rest of the paper is organized as follows. In section 2 we provide a brief review on graph cut technique and the concept of flux. The implementation of our algorithm is detailed in section 3. Section 4 presents the experimental results on two public datasets and performance comparison to other methods.

II. Graph Cut

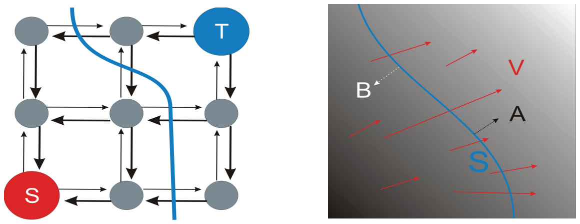
Graph cut is a widely used technique for interactive image segmentation in computer vision and medical image analysis [11]. It minimizes the energy function consisting of regional (computing likelihoods of foreground and background) and boundary terms (calculated by pixel intensity, texture, color, etc). Discrete graph cut is easy to implement, and is flexible to include various forms of regional, boundary, or geometric constraints.

A graph $G(\nu, \epsilon)$ is defined as a set of nodes ν and edges ϵ connecting neighboring nodes. An example of graph is shown in Figure 1(a). There are two special nodes called terminals, S source (foreground) and T sink (background) [12]. Edges between pixels are called n-links, while t-links are referred to the edges connecting pixels to terminals. All graph edges $e \in \epsilon$ including n-links and t-links are assigned some non negative weight (cost).

A cut is a subset of edges $C \in \epsilon$, that separate the graph into two regions: foreground and background. $G(c) = \langle \nu, \epsilon \backslash C \rangle$. Each cut has a cost which is defined as the sum of the costs of the edges that it severs. A globally minimum cut on a graph with two terminals can be computed efficiently in low order polynomial time via standard maxflow or push-relabel algorithms from combinatorial optimization [13].

The minimization energy using the boundary term has a tendency to follow short edges, frequently referred to as "the shrinking bias" problem [14]. This problem creates particular difficulties to segment thin elongated structures like the blood vessels.

One method to overcome the shrinking bias problem is to impose an additional connectivity prior, where the user marks the constrain connectivities [14]. We follow the approach presented in [15], which addresses the shrinking problem by adding the mechanism of flux into the construction of the graph.



a) s\t cut on directed graph   b) oriented surface

Fig. 1. Graph cut and flux. (a) Example of a graph, terminals S (foreground) and T (background). (b) Field of vectors **v** passing through surface S.

If any field of vectors **v** (e.g. image gradients), see Figure 1(b), are seen as speed in a water stream, the total volume of water passing through the hypersurface $\boldsymbol{S}$ can be compared to the absolute value of a flux. The orientation of the surface $\boldsymbol{S}$ determines the sign of the flux(**A** or **B**). Finsler metric, a class of continuous metric, can be approximated by discrete cut metric on regular grids. Finsler length can be represented by the sum of two terms, which represent the symmetric and antisymmetric parts of the cut metric. The symmetric part of the cut provide the standard length of contour independently of its orientation, while the antisymmetric part of the cut metric is considered flux of a given vector field through the contour [15].

III. Method

Our implementation can be divided in two main stages: preprocessing and graph construction. Figure 2 gives an overview of the method, which is described with detail next.

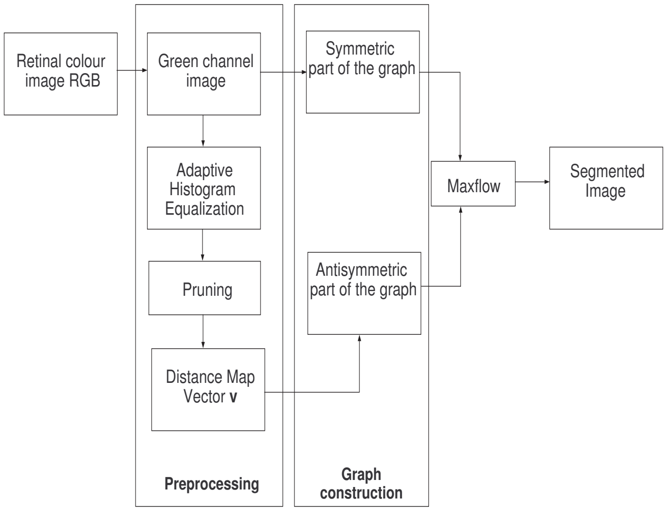

Retinal colour image RGB
Green channel image
Adaptive Histogram Equalization
Pruning
Distance Map Vector **v**
Preprocessing
Symmetric part of the graph
Antisymmetric part of the graph
Graph construction
Maxflow
Segmented Image

Fig. 2. General algorithm scheme

A. Preprocessing

The use of green channel for retinal image analysis has been considered in previous works [7], [3], [2] due to its high contrast between blood vessel and retinal background. In the preprocessing stage the green channel is separated from the RGB retinal color image, this image will be used to construct the symmetric part of the graph.

Similar to [4], we apply a contrast enhancement process to the green channel image. The intensity of the image is inverted, and the illumination is equalizated. The resulting image is enhanced using an adaptive histogram equalization process:

$$I_{Enhanced} = \left(\sum_{p' \in R(p)} \frac{s\left(I_2\left(p\right) - I_2\left(p'\right)\right)}{h^2} \right)^r \cdot M \qquad (1)$$

where $s(d) = 1 \; if \; d > 0$ and $s(d) = 0$ otherwise, $M = 255$, p denotes the pixel and $p's$ is the pixel neighborhood specified by a square window with width h. Increasing r would also increase the contrast between vessel pixels and the background. We have set up a window with $h = 81$ and $r = 6$. Figure 3(b) shows the result image after the enhancement process.

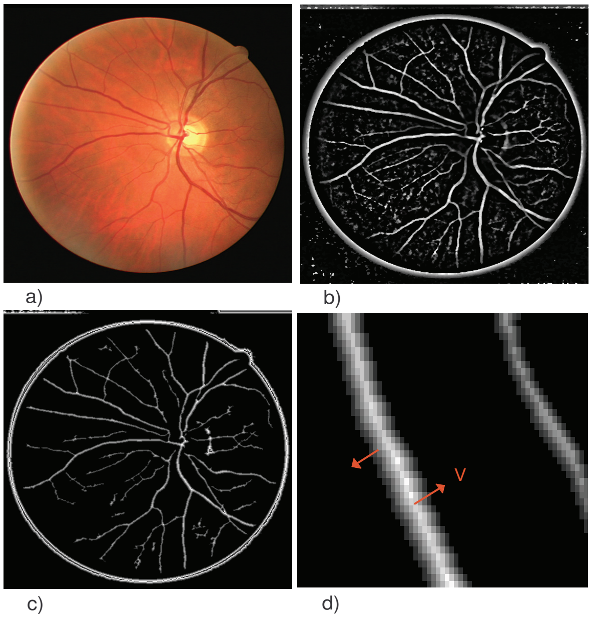


Fig. 3. Preprocessing. (a) input image, (b) enhanced image, (c) distance map, (d) a sample of a vessel into the distance map

The pruning stage is designed to eliminate the group of pixels with insufficient number of elements to conform a vessel. We apply a binary morphological open process, which eliminates the groups of pixel with less than a certain number of elements *E*. Based upon experimental trials *E* has been set to 200 pixels. At this stage it is important to reduce the false positives to minimum, because this information will be used to construct the graph connections.

From the pruned image, we create a distance map by applying the distance transform. This image is used to calculate the direction and magnitude of the vector **v** for the blood vessel pixels. Figures 3(c) and 3(d) show the distance map for the whole image and for a sample vessel, where we can see the center line as the brightest pixels and reduction in intensity to the direction of the edges (image gradients). This vector field will be used to construct the antisymmetric part of the graph.

B. Graph construction

Here we detail the construction of the graph. We adopt the method described in [15]. The graph is constructed for symmetric part g^+ and antisymmetric part g^- separately.

1) Symmetric part: The symmetric part corresponds to cut geometric length and is related directly with the n-link connections. The neighbor system can be described as a set of edges e_k, where $1 \leq k \leq N$, for N number of neighbors. e_k is the shortest vector connecting two pixels in the direction k. $W_k^+(p)$ is the weight of the edge e_k at pixel p. $\widetilde{W_k^+}(p)$ is the set of the edge weights at pixel p for all directions. The corresponding edge weights are defined according to:

$$\omega^+ = \frac{1}{2}Dg^+ \tag{2}$$

where D is a N x N matrix with entries

$$D_{ii} = -\frac{sin(\alpha_{i+1} - \alpha_{i-1})}{sin(\alpha_{i+1} - \alpha_i)sin(\alpha_i - \alpha_{i-1})} \tag{3}$$

$$\mathrm{D}_{ij} = \frac{1}{|sin(\alpha_j - \alpha_i)|} \quad if\ j-1 = \pm 1\ mod\ N$$

$$\mathrm{D}_{ij} = 0\ for\ all\ others\ entries$$

where α_k is the angle of the edge e_k with respect to the positive axis X.

We consider a grid map of 16 neighbors with edges $e_k, k = 1, 2, ..., 16$ (see Figure 4). For each pixel p in the green channel image, the edge weight $\widetilde{W_k^+}(p)$ is computed according to formula (2). g^+ is calculated using the pixel intensity difference between the two nodes.

$$g^+ = K \cdot exp\left(\frac{-(I_p - I_q)^2}{\sigma^2}\right) \tag{4}$$

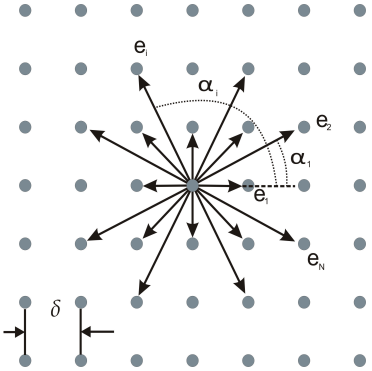


Fig. 4. Neighborhood system for a grid in the graph.

This function has a high value for pixels of similar intensities, when $I_p - I_q < \sigma$. But if the pixels are very different $I_p - I_q > \sigma$ the value is small, which represents a poor relation between the pixels, suggesting they belong to different terminals [12].

2) Antisymmetric part: The antisymmetric part corresponds to a flux of vector field **v** over a cut. Specific weights for t-links are based on the decomposition of vector **v**. Different decompositions may result in different t-links whose weights can be interpreted as an estimation of divergence. We consider a decomposition along grid edges using a natural choice of n-links oriented along the main axes, X and Y. This decomposition leads to t-link weights as

$$t_p = \frac{\delta^2}{2} \cdot (Vx + Vy) \quad (5)$$

where Vx and Vy are the components of vector $\mathbf{v}$, and δ is the size of the cell in the grid map. This is a particular scheme for computing divergence of vector $\mathbf{v}$. We add edge $(s \rightarrow p)$ with weight $C * (-tp)$ if $tp < 0$, or edge $(p \rightarrow t)$ with weight $C * tp$ otherwise. The parameter C is related to the magnitude of the vector $\mathbf{v}$, thus pixels in the center of the blood vessel have a higher connection to the source (foreground) than pixels in the edge of the blood vessels. Because the distance map is calculated on the pruned image, vector $\mathbf{v}$ is defined just for the pixels detected as blood vessels in the rough segmentation. The initialization of t-link weights for the rest of the pixel is set as $(p \rightarrow s)$ with weight $t = 0$ and $(p \rightarrow t)$ with weight $t = K$, where K is the maximum weight sum for a pixel in the symmetric construction.

The maxflow-v3.01 implemented by Komolgorov [1] is used in our implementation to compute the graph cut and find the final segmentation.

IV. Experiments and Results

Our algorithm was tested on two public datasets, the DRIVE[7] and the STARE[3], with a total of 60 images. The DRIVE consists of 40 digital images which were captured from a Canon CR5 non-mydriatic 3CCD camera at 45° field of view (FOV). The images have a size of 768×584 pixeles, eight bit per color channel. The dataset includes masks to separate the FOV from the rest of the image. Hand labeled images are available in two sets (set A and set B). The set A offers the manual labeling for all the images in the dataset, while the set B provides the manual labeling just for half of them.

The STARE dataset consists of 20 images captured by a TopCon TRV-50 fundus camera at 35° FOV. The size of the images is 700×605 pixeles, eight bit per color channel. We calculated the mask image for this dataset using a simple threshold for each channel of color, taking the matching results. The STARE dataset provides two sets of hand labeled images performed by two observers. The first observer labeled fewer vessel pixels than the second one. To test our method we adopt the first observer labeling as the ground truth for STARE, and set A for the DRIVE dataset.

The performance of segmentation is measured by true positive rate, false positive rate and accuracy rate. True positive rate (TPR) is defined as the total number of true positives, divided by the number of blood vessel pixel marked in the ground true image. False positive rate (FPR) is calculated as the total number of false positives divided by the number of pixels marked as non vessel in the ground true image. The accuracy rate is defined as the sum of true positives and true negatives, divided by the total number of pixel in the images. Figures 5 and 6 show the segmented images and the manually labeled images for the DRIVE and the STARE datasets respectively.

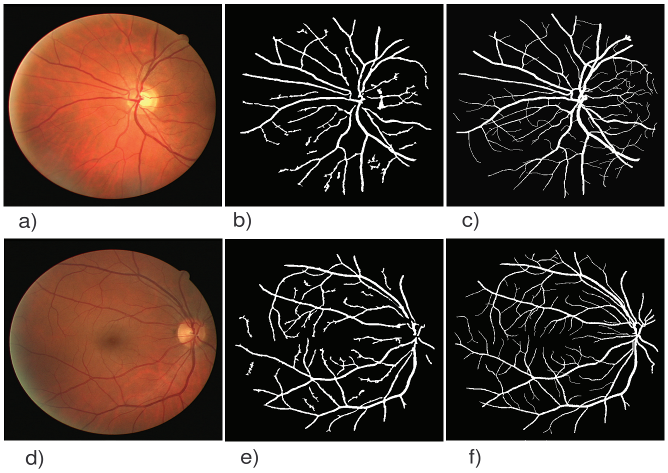


Fig. 5. The DRIVE dataset: a) and d) retinal images, b) and e) our segmentation results, and c) and f) manually labeled results.

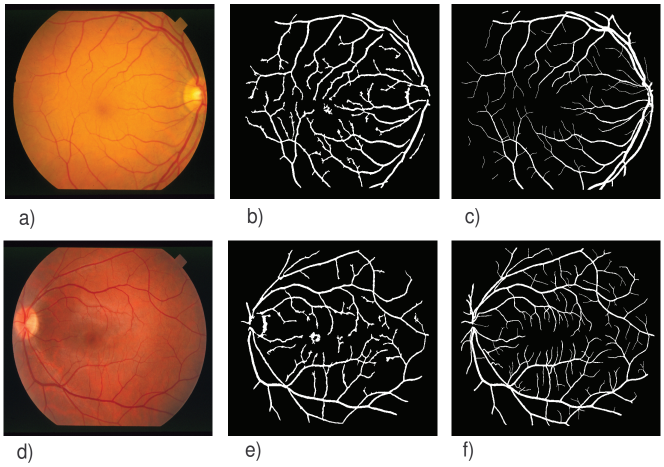


Fig. 6. The STARE dataset: a) and d) retinal images, b) and e) our segmentation results, and c) and f) manually labeled results.

The segmentation measure between two manual labellings for a same image establishes a reference to compare the average performance of other methods. For the DRIVE dataset set B is taken as a second observer. Segmentation in B is measured taking the manually labeled set A as a reference. At the same time the segmentation of the second observer in STARE is measured taking the first observer as a reference.

Tables 1 and 2 show the average performance of our method on the STARE and DRIVE datasets with a

[1] maxflow-v3.01 is available at
http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html.

comparison to the results in Staal [7],Mendonca [5], Martinez [9] and Usman [6]. We have also included the human observer rates. All the measured results used the same set as a reference, set A for DRIVE and first observer for STARE.

TABLE I
PERFORMANCE COMPARISON ON THE STARE DATASET.

Method	TPR	FPR	Accuracy
Second human observer [9]	0.8951	0.0438	0.9522
Hoover[5]	0.6751	0.0433	0.9267
Staal[7]	0.6970	0.0190	0.9541
Mendonca[5]	0.6996	0.0270	0.9440
Martinez[9]	0.7506	0.0431	0.9410
Graph Cut	0.7197	0.0335	0.9479

TABLE II
PERFORMANCE COMPARISON ON THE DRIVE DATASET.

Method	TPR	FPR	Accuracy
Human observer B[9]	0.7760	0.0275	0.9473
Staal[7]	0.6780	0.0170	0.9441
Mendonca[5]	0.7344	0.0236	0.9452
Martinez[9]	0.7246	0.0345	0.9344
Usman[6]	-	-	0.9469
Graph Cut	0.6782	0.0271	0.9478

The results show a TPR of 89.51% (STARE) and 77.6% (DRIVE) for the second observer segmentation. Most of the methods use the whole image to measure the performance. In [7] all the experiments are done on the FOV without considering the performance in the dark area outside the FOV. The method in [5] measures the performance on both the whole image and the FOV. The dark background outside the FOV in the retinal image is easy to segment. It is obvious an advantage in measuring the true negatives pixels when the whole image is considered. We have calculated the percentage of pixels outside the FOV in the images for the two datasets, which represents approximately the 25% of the pixels in the whole image. However, it does not affect all the measurement metrics, only where the true negative value is involve (e.g. Accuracy rate). On the other hand, most of the methods utilize the whole image for the performance measures, making the comparion fair. Considering the accuracy rate our method has the best performance on the DRIVE dataset. On STARE the TPR shows the graph cut outperformance over other approaches, including some of the supervised methods.

Any pathology presence in an image may obscure and confuse the blood vessel appearance, making it difficult to segment. Table 3 presents the results of normal cases (10 images) and abnormal cases (10 images) separately. Our graph cut method presents the best TPR values for normal and abnormal images.

TABLE III
PERFORMANCE COMPARISON ON THE STARE DATASET, NORMAL VERSUS ABNORMAL CASES

Normal cases			
Method	**TPR**	**FPR**	**Accuracy**
Mendonca[5]	0.7258	0.0209	0.9492
Hoover[5]	0.6766	0.0338	0.9324
Graph Cut	0.7417	0.0359	0.9471
Abnormal cases			
Method	**TPR**	**FPR**	**Accuracy**
Mendonca[5]	0.6733	0.0331	0.9388
Hoover[5]	0.6736	0.0528	0.9211
Graph Cut	0.6978	0.0311	0.9487

Unsupervised methods have also been reported in previous studies such as [10]. An experimental comparison between supervised and unsupervised methods is presented in [10]. These results, combined with ours, are show in Figure 7 in the form of receiver operating characteristic (ROC) curves. ROC curve is a graphical plot of the TPR versus FPR for a binary classifier, where its discrimination threshold is varied producing different TPR for different FPR. We compute different TPR by changing the initialization of the weight edges for the pixels with the terminals S (foreground) and T (background). The performance of the second observer **B** has been marked into the graphic as a reference.

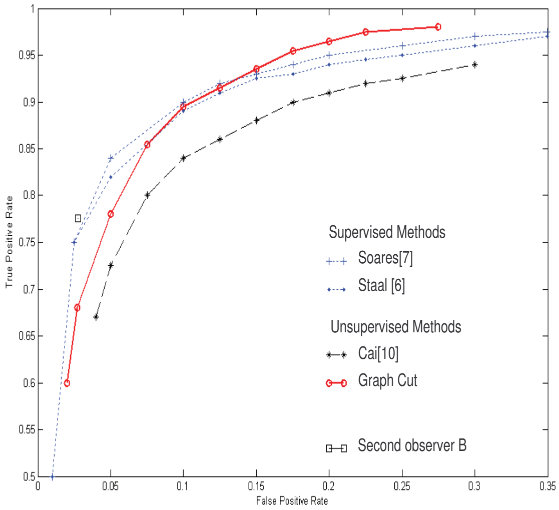


Fig. 7. ROC curve performance for supervised and unsupervised methods in DRIVE

These results show that our algorithm outperforms the unsupervised method presented in [10]. The supervised methods perform slightly better but at cost of an extra, and potentially intensive, training process.

V. CONCLUSIONS

Blood vessel segmentation is an important process in the analysis of retinal images. A method based on the construction of a graph has been presented in this paper. Integrating the mechanism of flux into the graph cut method provide a good balance between flux stretching and length shrinking along the boundary. Our method does not require ad hoc tuning of parameters on different datasets. Experimental results on the DRIVE and STARE datasets show that out method outperforms other unsupervised methods and is comparable to the supervised methods in the previous studies.

Currently, the initialization of the t-links for the pixels marked as non-vessel in the rough segmentation is defined by their probability to be background. Nevertheless we believe that combining the regional constraints in this process can improve the segmentation. Regional constrains can be calculated using the likelihood of the pixels on the foreground and background.

Acknowledgment

The authors would like to thank the Mexican National Council for Science and Technology (CONACYT) for financial support.

References

[1] J. K. Kamarainen L. Lensu I. Sorri A. Raninen R. Voitilainen H. Uusitalo H. Kalviainen T. Kauppi, V. Kalesnykiene and J. Pietila. Diaretdb1 diabetic retinopathy database and evaluation protocol. *In Proceedings of British Machine Vision Conference 2007.*

[2] X. Jiang and Mojon D. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):131–137, 2003.

[3] A. D. Hoover. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203–210, 2000.

[4] D. Wu. On the adaptive detection of blood vessels in retinal images. *IEEE Transactions on Biomedical Engineering*, 53(2):341–343, 2006.

[5] A. M. Mendonca and A. Campilho. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Transactions on Medical Imaging*, 25(9):1200–1213, 2006.

[6] M. Usman Akram, Anam Tariq, and Shoab A. Khan. Retinal image blood vessel segmentation. *Proceedings of the international conference on information and communication technology*, 978(1), 2009.

[7] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004.

[8] J. V. B. Soares. Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging*, 25(9):1214–1222, 2006.

[9] M. E. Martinez-Perez, A. D. Hughes, S. A. Thom, A. A. Bharath, and K. H. Parker. Segmentation of blood vessels from red-free and fluorescein retinal images. *Medical image analysis*, 11(1):47–61, 2007.

[10] W. Cai and A. C.S.Chung. Multi-resolution vessel segmentation using normalized cuts in retinal images. *In Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1:928–936, 2006.

[11] Y. Boykov and M-P. Jolly. Interactive organ segmentation using graph cuts. *In Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 276–286, 2000.

[12] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2), 2006.

[13] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[14] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008.*, 1:1–8.

[15] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. *In Proceedings of Tenth IEEE International Conference on Computer Vision, ICCV 2005.*, 1:564–571.