Blood Vessel Segmentation in Retinal Images

P. Echevarria T. Miller J. O'Meara

February 8, 2004

Abstract

Segmentation of blood vessels in retinal images allows early diagnosis of disease; automating this process provides several benefits including minimizing subjectivity and eliminating a painstaking, tedious task. Previous approaches, while satisfactory in some cases, still leave room for improvement, especially in abnormal retinal images. We propose to utilize a tracking based algorithm based on level sets and fast marching methods.

1 Introduction

Examination of blood vessels in the eye allows detection of eye diseases such as glaucoma and diabetic retinopathy. Traditionally, the the vascular network is mapped by hand in a time-consuming process that requires both training and skill. Automating the process allows consistency, and most importantly, frees up the time that a skilled technician or doctor would normally use for manual screening.

While success has been achieved on normal retinal images, on abnormal or diseased images - for which accuracy is more crucial than ever - the algorithms frequently fail. For instance, popular convolution approaches suffer from variable retinal background and low contrast between vessels and surrounding pixels. Tracking algorithms fail in special cases on abnormal images; they are often sidetracked by light objects and sometimes experience difficulty locating starting points. Current levels of success are still frequently inadequate for widescale implementation.

Our method utilizes the concept of level sets to remove noise, enhance the image, and track the edges of the vessels. Initally, to improve contrast in the image, we implement the filter designed by Chaudhiri et. al. [2]. Then we follow with a level set method to enhance the image and remove noise. After this, we look for starting points and finally finish with a tracking

method featuring an interface which propagates through the blood vessels. The level set and fast marching methods have been researched extensively for use on medical images from brain MRIs to arteriograms. Though there has been some limited application of this approach to retinal images, it is not fully autonomous, and a cohesive study on a large number of normal and abnormal images has never been conducted.

Thus far, each stage has been partially implemented. We have reimplemented the Chaudhiri Gaussian matched filter. We have applied the level set image enhancement algorithm to both our filtered images and to the publicly available images pre-filtered by Hoover et. al. [3]. In attempting to locate starting points for the tracking algorithm, we have experimented with a number of methods including thresholding, skeletonizing, and Matlab filters. We are currently in the process of fleshing out the final stage of propagating the interface to yield the final segmentation.

2 Background

The problem we are trying to solve is to segment images of retinas into portions that are blood vessel and not blood vessel.

There has been extensive previous work on this problem and related problems, such as artery mapping in angiograms. This work can be divided into several techniques.

One technique is a window based method. This involves looking at a small region of the image and extracting possible blood vessel pixels based on local image characteristics. This was done by [7] in the mapping of arteries in angiograms. A model was created for blood vessel densitometry and matched to regions of angiograms to determine blood vessel location as well as diameter and cross-sectional area. Work done by [2] attempts to match the gray-scale intensities of regions of a retinal image to a Gaussian profile and thereby locate the blood vessels. Edge detection was done by [8], and parallel edges were connected and identified as blood vessels.

Another methodology involves exploiting the known structure of veins and arteries to evaluate potential vessels. Pattern recognition is utilized to eliminate spurious blood vessels [11]. More recent methods typically utilize a cross-section of the three major categories of retinal image segmentation. Generally, some type of filtering and thresholding is used in the preprocessing stage. The image is then processed in one of several ways, and finally a post-processing step detects misclassified pixels and patches together labelled segments. Li et. al. [5] reject the traditional single Gaussian filter

for a double Gaussian filter which more accurately models the vessel profile. They use this piecewise Gaussian model to differentiate between arteries and veins. Lowell et. al. [6] use a localization method utilizing a correlation filter to tackle the problem of locating the optic disk, frequently a starting point for tracking algorithms. Lalonde et. al. [5] use the Canny operator in order to achieve optimal edge placement and to calculate an accurate estimate of the normal vector the the edge. They follow the edge detection with thinning and then a tracking algorithm. Their tracking algorithm tracks each edge of each vessel individually before combining found walls as vessels by exploiting knowledge of the parallel character of vessel borders. Post-processing displays the centerline of the vessels. Hunter et. al. [4] developed an algorithm relying primarily on image filtering with secondary reliance on exploiting global connectivity. Order statistic filtering at multiple orientations is used for its robustness against noise and insensitivity to distractors. The results are cleaned up for noise and thinned. Finally, a Gaussian model is fit locally to the image at each pixel in order to measure vascular diameter and estimate beading.

In addition, several tracking methods have been implemented in the past. Previous tracking methods proceeded by first determining start points and then tracking the vessels from those points. One principal strategy involves locating the fundus, usually the brightest part of the eye, and locating starting points from there. For instance, Tamura et. al. [10] use this strategy and then incrementally locate pixels within the center of vessels by incrementally probing a vessel with a Gaussian profile. Tolias and Panas [12] also began at the fundus but propagated their probe forward using a fuzzy pixel-classification profile rather than a Gaussian profile. Others chose alternative means to locate tracking starting points. Hoover et. al. [3] skeletonize a thresholded image and start from the endpoints of the line segments they obtained. Subsequently, they probe the image and test segments for region-based properties to determine whether they are vessels or not.

One of the landmark works in this field was done by [3] and uses a hybrid method. These authors use match filters (as in [2]) to extract possible vessels from the background. This filtered image is then segmented using iterative thresholding. An example of this work is shown in Figure 1. One of the reasons the work is considered so crucial is that they hand labelled the blood vessels on 20 images and used these as the truth data. This dataset has been referenced and used by many subsequent researchers.

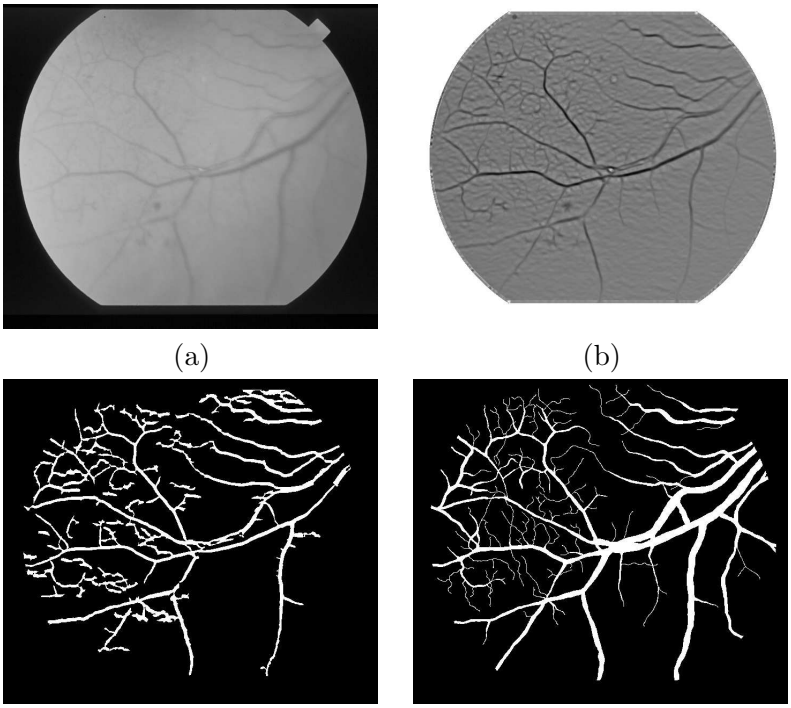


Figure 1: (a) An image of a retina, (b) its matched filter response, (c) the segmented image, and (d) the hand labelled "truth" image. [3]

3 Approach

3.1 Smoothing with Level Sets

Initially, the images are filtered using two-dimensional matched filters in the manner of Chaudhuri et. al. [2]. The filter employs three primary principles. First, pairs of blood vessel walls may be approximated as linear due to typically small curvature. Second, vessels appear darker than the background as a result of low reflectance in comparison to other retinal surfaces. The profile may be modelled as a Gaussian. Finally, the width of blood vessels is variable [2]. Precisely, the algorithm is as follows. A kernel is generated in the following fashion: $P = [x\ y]$ is a point in the kernel and θ_i is the orientation. The kernel is centered around the origin, $P = [0\ 0]$. The rotation matrix,

$$r_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$$

is used, and $p_i = Pr_i$. Twelve different kernels are generated for an angular resolution of 15 degrees. Each kernel is convolved with the image, and at each point within the image, the maximum response is considered [2].

The Gaussian is extended to three standard deviations. The neighborhood N is defined as follows: $N = \{(u,v)||u| = 3\sigma, |v| = L/2\}$. Then within N, $K_i(x,y) = -\exp(-u^2/2\sigma^2)$. The mean value of each kernel, m_i, is determined by averaging across N, and for all p_i in N, $K_i' = K_i(x,y) - m_i$. Finally, the kernel is scaled by a factor of 10 [2].

After the initial matched filtering stage, level sets are utilized to remove noise from the image while preserving the edges. The image is regarded as a compilation of iso-intensity contours. Spikes of noise corresponding to high curvature are removed and oscillations in boundaries smoothed out using variations of curvature flow [9]. The curvature evolution equation is as follows:

$$\kappa = \nabla \cdot \frac{\nabla I}{|\nabla I|}$$

Our goal is to generate a flow which will smooth out undesired oscillations yet maintain fundamental characteristics of the shape. The concept of min/max switch is utilized in the following function:

$$\overline{F}_{min/max} = \begin{cases} \max(\kappa, 0) & if\ Ave_{\phi(x,y)}^{R=kh} < T_{threshold} \\ \min(\kappa, 0) & otherwise \end{cases}$$

Finally, $I_t = -\overline{F}|\nabla I|$.

The Ave function is the average value of ϕ given a disk of radius $R = kh$ centered around a point (x,y). k is chosen as 1, a choice which allows some movement of the zero level corresponding to the boundary [9]. We experimented with various values of h. Thus a speed function dependent on the average value of f and the local level of curvature was obtained. Given $I(x,y)$, the original intensity profile, one iterates until I_t goes to zero and stable values are reached.

3.2 Fast Marching with Level Sets

We plan to segment the retinal pixels by propagating initial seed interfaces within major vessels until they encompass all the retinal pixels. The speed

F with which an interface is made to propagate is constructed so that it moves quickly in areas of low gradient and stops when the gradient is high. In addition to this, the speed also takes into account the curvature of the interface so as to provide some "surface tension" along the boundary and maintain its smoothness as it propagates.

We construct F so that it is always positive (that is, the interface always expands and never contracts). Given this, we can define the propagation of the interface along the image with function $T(x, y)$ denoting the time the interface passes through a point (x, y) in the image. We propagate the interface by solving the boundary formulation partial differential equation

$$|\nabla T|F = 1$$

and let the interface front at time t be $\{(x, y)|T(x, y) = t\}$.

The Fast Marching method gives us a way to do this efficiently. Given a point O whose T is known, we can compute the T of its neighbors by solving the partial differential equation:

$$\left[\max(D_{ij}^{-x}(T), -D_{ij}^{+x}(T), 0)^2 + \max(D_{ij}^{-y}(T), -D_{ij}^{+y}(T), 0)^2\right]^{1/2} = 1/F_{ij} \tag{1}$$

The Fast Marching algorithm is as follows (adopted from [9])

1. Divide all the points in the image into three sets:
   - Known = all points (x, y) s.t. $T(x, y)$ is known for all points within and on the interface, $T(x, y) = 0$
   - Trial = all points one pixel away from a Known point
   - Unknown = all points that are not in Known nor in Trial
2. Compute T for all points in Trial.
3. Let A be the Trial point with the smallest T value.
4. Add the point A to Known; remove it from Trial.
5. Tag as Trial all neighbors of A that are not Known. If the neighbor is in Unknown, remove, and add to the set Trial.
6. Recompute the values of T at all Trial neighbors of A according to 1 by solving the equation.

7. Return to 3.

The loop is repeated until the interface stops moving.

When the interface is near a boundary, we switch to the Narrow Band level set method. To propagate the interface with this method, we solve the initial value formulation PDE:

$$\phi_t + F|\nabla\phi| = 0 \tag{2}$$

and let the interface at time t be $\{(x, y)|\phi(x, y, t) = 0\}$

The Narrow Band method is as follows (adopted from [9]):

1. Create a narrow band of pixels surrounding the interface, and tag them as Alive.
2. Build Land Mines to indicate which pixels are near the edge.
3. Initialize Far Away points outside the narrow band with large positive values if values are outside the interface itself.
4. Solve the equation 2, until land mine hit.
5. Rebuild ϕ and return to step 1. Stop when the the interface stops moving.

3.3 Post Processing and Evaluation

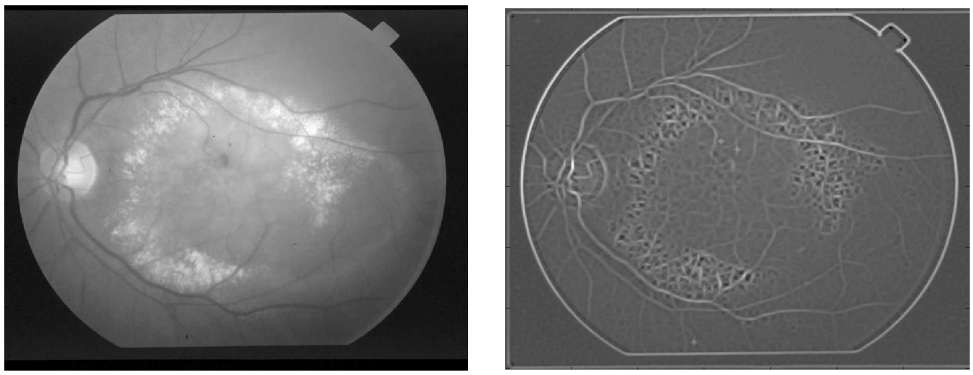
As with many other vessel detection algorithms, it is likely that some final processing will be necessary even after execution of the fast marching algorithm. The approach we will use is highly dependent on the results of the fast marching algorithm and therefore is as yet undeveloped.

Evaluation of our algorithm will be performed using the method of the well known Receiver Operating Characteristic (ROC) curve. This plots the false positive rate (1-specificity) versus the true positive rate (sensitivity) of a test. This evaluation method is well suited to blood vessel mapping algorithms since there is a standard dataset that includes accepted "truth" images, showing segmented vessels.

4 Current Results

4.1 Smoothing with Level Sets

Currently, Chaudhiri's filtering method is imitated, but the results are not yet precisely replicated. This is shown in Figure 2.



(a) (b)

Figure 2: (a) An unfiltered retinal photo. (b) Same photo filtered with a rotated Gaussian filter.

Starting with images obtained from the website of Hoover et. al. that result from a spatial probe filter, the level set noise filtering algorithm is applied. Figure 3 displays the original MSF image as well as the result of the level set algorithm after ten time-steps.

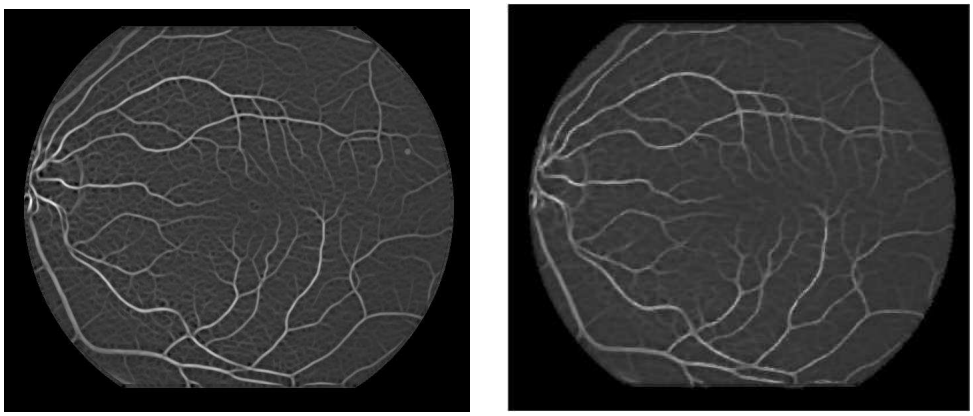
On closer inspection (Figure 4) the noise removal is more apparent. Once again, the image on the left is the original, and the image on the right has had the level set algorithm applied.

4.2 Fast Marching with Level Sets

Performing the fast marching algorithm requires initial seed points. These are calculated using a thresholding technique, followed by several transformations. The end result is an image of likely blood vessel locations from which the fast marching algorithm can begin.

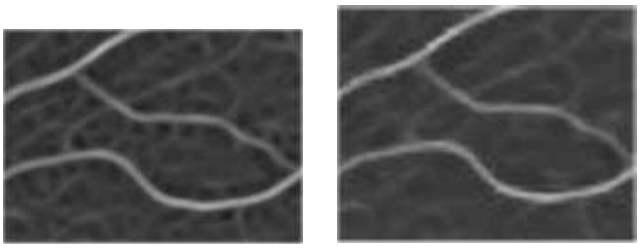
The thresholding is done using a double thresholding technique. The image is thresholded at two levels - such that 8 and 11 percent of all pixels are selected. Connected regions of the 11 percent image that did not also appear in the 8 percent image are then eliminated. The thresholded image then does not include the minor regions that only appeared in 8 percent image.

Having created a binary version of the image, it is then altered with a series of morphological operators. The image is skeletonized, spurs are removed, and isolated pixels are eliminated. Finally, any loops in the image are detected and subtracted from the skeleton of the image. In many blood



(a) (b)

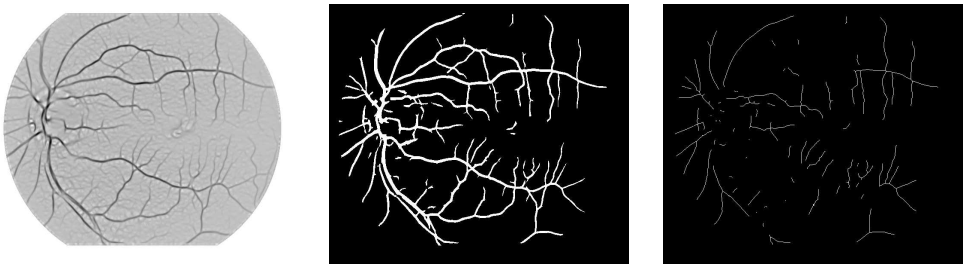
Figure 3: (a) The original MSF image from [3] (b) Result of level set algorithm after ten time steps.



(a) (b)

Figure 4: (a) The original MSF image from [3] (b) Result of level set algorithm after ten time steps.

vessel images unusual spots resulting from the imaging process form loops. Unfortunately many blood vessels cross other blood vessels, also forming loops, and these are eliminated as well. However, since the goal of the process is merely to find a few good starting points for the fast marching algorithm, it is of primary importance that there are few false positives in the process.



(a) (b) (c)

Figure 5: (a) An MSF filtered image (b) The same image double thresholded at 8 and 11 percent, and (c) the calculated seed points for the fast marching algorithm.

An example of a possible set of starting points is shown in Figure 5. This particular example has a 0.05% false positive rate, amounting to 182 false positives. The average false positive rate 0.19%. The average number of false positive pixels is 731 with a maximum of 2082 and a minimum of 68. A typical amount of time for a single run to complete on a 1.13 GHz AMD processor is 1 minute and 15 seconds.

Currently a framework for propagating a given seed interface is in the process of being implemented.

4.3 Post Processing and Evaluation

Currently no code has been written for post processing of images. This step is awaiting the results of the previous processing steps.

Code has been written for calculation of an ROC curve. This code takes as input a truth image and a series of segmented images at different thresholds. It outputs the number of false positives and true positives as well as a plot of the corresponding ROC curve.

5 Next Steps

5.1 Smoothing with Level Sets

The matched filter response currently yields results inferior to those obtained by the filters of Chaudhiri et. al. and of Hoover et. al. The kernel must be optimized to improve this result and obtain a viable image upon which to apply the level set and fast marching methods algorithms. In addition to simply imitating Chuadhiri's filter, there are several options to improve performance. Currently, a kernel of fixed width is being used. Running several kernels of varied width over the image and again taking the optimal response has potential. Implementing the piecewise Gaussian algorithm of Li et. al. is an additional option.

Parameters of the level set such as number of steps, neighborhood size, and the choice of averaging function should be manipulated to determine which combination yields the optimal input to the fast marching algorithm.

5.2 Fast Marching with Level Sets

Depending on the effectiveness of the starting points for the fast marching algorithm some refinement to the algorithm may be necessary. This may include choosing to pass on only points that are part of long segments, as these are more likely to be part of blood vessels.

We must also implement the function that solves equation 1 and plug it into the framework. Once this is done, testing will be done in order to tweak the parameters of the algorithm.

5.3 Post Processing

The majority of the post processing remains to be completed. The necessary extent of the work to be done can only be determined once the other image processing is completed. It is likely that this will include length filtering, such as that done by [1].

6 Summary and Conclusion

Our current work is producing promising results. We have generated relatively clean filtered images which resemble paper cut-outs. These images appear well-suited as starting points for our level set algorithm which relies on continuous surfaces and clean edges. We are pursuing a number of alternatives to generate starting points for the tracking algorithm, and we are

currently implementing the interface propagation algorithm. Depending on the success of our current strategy, we are optimistic that we will be able to achieve our quantitative goals.

References

[1] T. Chanwimaluang and G. Fan. An efficient blood vessel detection algorithm for retinal images using local entropy thresholding. In *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, 2003.

[2] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Tran. on Medical Imaging*, 8(3):263–269, September 1989.

[3] A. Hoover, V. Kouznetsova, and M. Goldmaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. In *AMIA Annual Symposium*, 1998.

[4] A. Hunter, J. Lowell, D. Steel, A. Basu, and R. Ryder. Non-linear filtering for vascular segmentation and detection of venous beading. Technical report, University of Durham, 2003.

[5] M. Lalonde, L. Gagnon, and M.-C. Boucher. Non-recursive paired tracking for vessel extraction from retinal images. In *Proc. of Conference Vision Interface*, 2000.

[6] J. Lowell, A. Hunter, D. Steel, A. Basu, R. Ryder, and E. Fletcher. Optic nerve head segmentation. Technical report, University of Durham, 2002.

[7] T. Pappas and J. Lim. A new method for estimation of coronary artery dimensions. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36(9):1501–1513, September 1988.

[8] A. Pinz, S. Bernogger, P. Datlinger, and A. Kruger. Mapping the human retina. *IEEE Trans. on Medical Imaging*, 17(4):606–619, August 1998.

[9] J.A. Sethian. *Fast Marching Methods and Level Set Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences.* Cambridge University Press, 1999.

[10] S. Tamura, Y. Okamoto, and K. Yanashima. Zero-crossing interval correction in tracing eye-fundus blood vessels. *Pattern Recognition*, 21(3):227–233, 1988.

[11] S. Tamura, K. Tanaka, S. Ohmori, K. Okazaki, A. Okada, and M. Hoshi. Semiautomatic leakage analyzing system for time series fluorescein ocular fundus angiography. *Pattern Recognition*, 16(2):149–162, 1983.

[12] Y. Tolias and S. Panas. A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *IEEE Trans. on Medical Imaging*, 17(2):263–273, April 1988.