Automatic and Interactive Retinal Vessel Segmentation

Zygmunt L. Szpak, Jules R. Tapamo

School of Computer Science, University of Kwazulu-Natal, Durban, Republic of South Africa

ABSTRACT

Retinal fundus images are used in the diagnosis and treatment of many eye conditions, such as diabetic retinopathy and glaucoma. During a clinical examination, an ophthalmologist is able to determine the onset of disease by taking certain features of the retinal vessels of the fundus into account. Often the ophthalmologist will need to select what parts of a retinal fundus image constitute vessels, so that certain statistics such as the thickness of the vessels can be calculated. Labelling all the vessels however, is a tedious and time consuming process. We make use of an edge detection approach, followed by connected component analysis and a fast Shi-Karl level-set technique, to extract the vessels from the image. Our work focuses on using image processing techniques in order to develop a computer program that can automatically and interactively detect and segment blood vessels in these images, thereby saving the ophthalmologist considerable time. We create our tool as a series of plug-ins for ImageJ, which is a public domain, Java-based image processing program developed at the National Institutes of Health. By doing so, we facilitate the use of our tool as part of a bigger system. Furthermore, ophthalmologists can easily modify the proposed vessel segmentation of our system using ImageJ, and have an ever growing library of image processing plug-ins at their disposal, to annotate, enhance and measure images.

KEYWORDS: Retinal Vessels; Interactive Segmentation; Active Contour Implementation; ImageJ

1 INTRODUCTION

Diseases that affect the optic nerve and that can cause blindness, such as glaucoma and diabetic retinopathy, can often be prevented if they are diagnosed at an early stage. It has been shown that a screening process, using retinal fundus images, may aid in reducing blindness by 50% for diabetic retinopathy cases [1].

During the screening process, the ophthalmologists will focus on features of the retinal vessel. Features such as vessel thickness, shape, color and pathologies that appear around the vessels, all aid in the diagnosis of the disease. Computer aided screening systems have been developed to track the changes of the vessel structure of a patient, and to detect pathologies [2]. Such screening systems often rely on the segmentation of retinal vessels, and ophthalmologists are forced to spend a lot of time manually segmenting the vessels in the retinal fundus images.

Automating the segmentation of the vessels in the retinal images is an ongoing field of research, and many image processing techniques have been proposed. The bulk of previous work can be roughly categorized into multi-scale analysis [3], matched filters [4], mathematical morphology schemes [5], adaptive thresholds [6], tracking based approaches [7] and deformable models [8]. A leading method has not emerged, and an ongoing effort is being made to improve the results.

In our work, we focus on segmenting the retinal vessels automatically using image gradients, together with connected component analysis and post-processing, to suppress false edges. Additionally, we extract the contour of the segmented vessels using the Shi-Karl fast level-set method [9], and create a custom selection marquee in ImageJ [10], to allow the user to modify the segmentation.

The rest of our paper is organized as follows. In the next section we review related work on retinal vessel segmentation. In section 3, we explain the processing steps of our retinal vessel segmentation system. The implementation details, and comparisons of our segmentation results, to several other methods using the DRIVE [22] database of retinal images, are presented in section 4. Finally in section 5, we conclude our paper with a discussion of limitations and future work.

2 RELATED WORK

Much has been published on the topic of automatic vessel segmentation, and as we have already mentioned, researchers have applied a wide variety of techniques to extract vessels. A good review of vessel segmentation methods can be found in [11] and [12].

Vessels exhibit a lower reflectance than other features, which results in their appearance as dark structures in retinal images [13]. This contrast allows vessels to be segmented using an edge detection technique. This observation was exploited in the work of Malladi et al.[14], who proposed to extract vessels by halting active contours at positions where the edge

Email: Zygmunt L. Szpak `zygmunt.szpak@cs.ukzn.ac.za`, Jules R. Tapamo `tapamoj@ukzn.ac.za`

response is strong.

A similar approach was taken by Caselles et al.[15], who made use of a geodesic active contour model.

McInerney and Terzopoulos incorporated the Laplacian operator into their parametric active contour model called T-Snake [16], and used it to discover boundaries for tissue extraction in medical images.

Lu and Eiho [17] used edge detection, followed by finding branches and tracing contours, to extract coronary arterial boundaries from X-Ray Angiography images.

In terms of user intervention, Quek et al.[18] proposed a model to interpret Neurovascular X-Ray angiography images interactively. Their so called attentionally-based interactive model(AIM), links the user with the computer as integral partners, and facilitates varying degrees of human intervention in the process of segmenting vessels, and other structures. Their system also employs several edge detection algorithms, such as Canny and Sobel, for the extraction of vessel boundaries.

Another semi-automatic vessel segmentation tool was developed by Shen and Johnson [19]. In their system, the user supplies a initial starting point on a vessel, and thereafter edge detection and contour tracing methods are applied to find region boundaries.

There are two main problems with using edge magnitude to segment retinal vessels. Firstly, when the contrast between the vessel and other structures in the eye is poor, the edge response will be low. Secondly, noise in the image will result in false edge responses.

To minimize the impact of noise on our edge responses, we make use of a Laplacian of Gaussians as our edge detection mechanism.

Vessels that have poor contrast are usually very thin and short, and as yet no leading method has emerged that can detect those very thin vessels with a high degree of accuracy, without many false positives. Instead of attempting to detect these thin vessels automatically, we prefer to allow the end-user to manually segment them using ImageJ. Although ophthalmologists still have to segment thin vessels manually, not much time is needed because those vessels are short and thin.

3 RETINAL VESSEL SEGMENTATION

Our retinal vessel segmentation algorithm can be viewed as a sequence of steps, which starts with image enhancement to minimize noise and increase contrast in the image; followed by edge detection and automatic thresholding, and ends with connected component manipulation and contour extraction.

In the following sections we explain each processing step in more detail.

3.1 Extracting the Green Channel

In color retinal fundus images, the green channel usually contains the best contrast between the vessel structures and the background [23]. The first step of our segmentation method, is to extract the green channel from the RGB image, since the better the contrast between the vessel structures and the background, the stronger the edge response. Refer to Figure 1.

3.2 Sharpening the Image

The second step consists of an unsharp mask filter, which is the process of subtracting a blurred version of an image from the image itself. The process can be expressed as

$$\widehat{f}(x,y) = f(x,y) - (\sum_{i=-k}^{i=k}\sum_{j=-k}^{j=k} f(x+i,y+j)g(i,j)), \tag{1}$$

where $\widehat{f}(x,y)$ denotes the sharpened image obtained by unsharp masking, $f(x,y)$ is the original image, k denotes the size of the convolution kernel, and $g(x,y)$ is a gaussian in two dimensions defined as

$$g(x,y) = \frac{1}{2\pi\sigma^2}e^{\frac{-x^2-y^2}{2\sigma^2}}. \tag{2}$$

The filter enhances the appearance of vessels, and results in a stronger edge response.

3.3 Edge Detection

A Laplacian of Gaussians filter was convolved with the image to detect edges. A standard laplacian highlights gray-level discontinuities in an image, and deemphasizes regions with slow varying gray levels, however it is also very sensitive to noise. The Laplacian of Gaussians filter reduces the response to noise by smoothing the laplacian convolution mask. The 2D Laplacian of Gaussians function centered on zero, and with a standard deviation of σ has the form:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}[1 - \frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3}$$

The edge response image $\widehat{f}(x,y)$ is then computed with

$$\widehat{f}(x,y) = \sum_{i=-k}^{i=k}\sum_{j=-k}^{j=k} f(x+i,y+j)LoG(i,j), \tag{4}$$

where $f(x,y)$ is the original image, and k is the size of the convolution kernel.

3.4 Thresholding the Edge Response

After the Laplacian of Gaussians operator is applied on the image, the resulting image consists of edge magnitudes. A threshold is needed to separate strong edge responses, which ideally represent vessels, from weak edge responses, which represent noise or background.

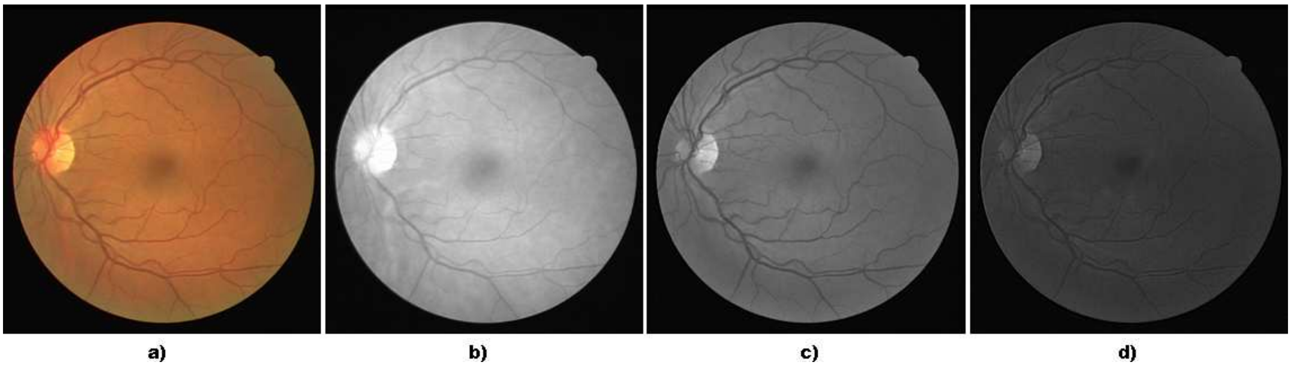


Figure 1: Comparison of the contrast between red, green and blue channels of a color retinal fundus image. **a)** The original color retinal fundus image. **b)** The red channel of the image. **c)** The green channel of the image. **d)** The blue channel of the image.

This threshold creates a binary image, and the process can be expressed as

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) \text{ is } \geq t; \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $f(x,y)$ represents the original image, and $g(x,y)$ is the resulting binary image. We tested several techniques that attempt to determine the best threshold t automatically. The techniques included Otsu [25], entropy of the histogram [24], and a mixture of gaussians approach. The mixture of gaussians method performed the best across our test images.

The mixture of gaussians threshold algorithm assumes that the foreground and the background can be modelled by two gaussian distributions. In other words, the algorithm assumes that the *normalized* histogram $H(i)$, is a mixture of two gaussian distributions, defined as

$$H(i) = \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}(\frac{i-\mu_1}{\sigma_1})^2} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{1}{2}(\frac{i-\mu_2}{\sigma_2})^2}, \tag{6}$$

where i represents the pixel grey-level intensity, and $P_1 + P_2 = 1$.

The aim of the algorithm is to choose a threshold intensity t, to minimize the probability of wrongly classifying a foreground pixel as background, or vice versa.

There are several different techniques of solving this problem, which are discussed in [20]. One commonly used approach called the minimum error method, proposed by Kittler and Illingworth in [21], is to optimize a criterion function $J(t)$, which is related to the average pixel classification error rate. It is defined as

$$\begin{aligned} J(t) &= 1 + 2[P_1(t)\ln\sigma_1(t) + P_2(t)\ln\sigma_2(t)] \quad (7) \\ &- 2[P_1(t)\ln P_1(t) + P_2(t)\ln P_2(2)]. \end{aligned}$$

The function $J(t)$ is evaluated for all grey-level intensities, and the parameters P_1, P_2, σ_1 and σ_2 are calculated from the two portions of the histogram, that are separated by the threshold t. The threshold t that minimized $J(t)$, is then used to binarize the image.

The value $P_1(t)$ can be calculated with

$$P_1(t) = \sum_{i=0}^{t} H(i); \tag{8}$$

and the value $P_2(t)$ can be calculated with

$$P_1(t) = \sum_{i=t+1}^{G} H(i); \tag{9}$$

for a threshold $t \in \{0, 1, 2, \ldots G\}$, where G is the maximum number of grey-levels.

3.5 Connected Components Analysis

After producing the binary image $g(x,y)$, using the threshold determined with the mixture of gaussians algorithm, there is no guarantee that the binary image consists of vessels only (refer to Figure 2). This is because spurious noise in the image may have produced strong edge responses, which would have been labelled incorrectly as vessels.

The foreground and background of the binary image $g(x,y)$, can be viewed as a set of classes of connected and similar pixels. By considering the 4-connected discrete neighborhood of a pixel $g(x,y)$ as the pixels

$$g(x-1,y),\ g(x+1,y),\ g(x,y-1), \text{ and } g(x,y+1),$$

we can use this 4-connectivity to group the binary image $g(x,y)$ into classes c_0, c_2,c_3,...,c_{k-1}, where k is the number of connected components. The resulting image C can be defined as:

$$C = \bigcup_{i=0}^{k-1} c_i, \tag{10}$$

$$c_i = \{(x,y,i),\ 0 \leq y \leq m-1 \text{ and } 0 \leq x \leq n-1\}.$$

The letters m and n, are the number of rows and columns of the image respectively.

Often the spurious noise pixels are not connected to vessels, and form small connected components. By removing connected components of size $< \epsilon$, we can remove a lot of pixels that are not vessels.

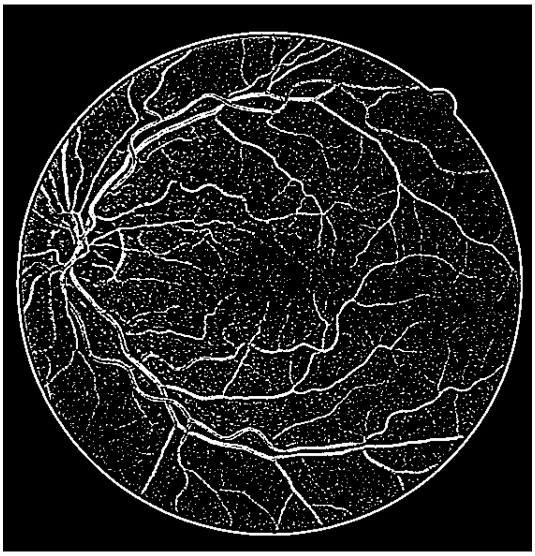


Figure 2: The result of thresholding on the response of a laplacian of gaussian edge detector, using a threshold T, that was determined automatically by a mixture of gaussians algorithm.

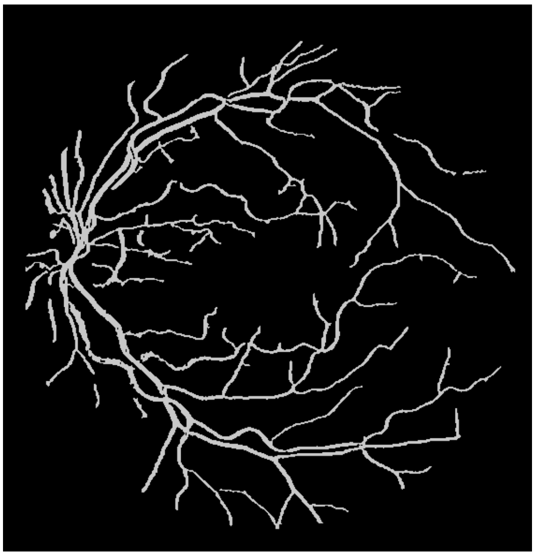


Figure 3: A binary image representing the vessels in a retinal fundus image. Using connected component manipulation, components of size < 20 were removed; holes were filled, and the large outer ring was removed.

It is also possible to remove the large ring component that is formed around the retinal fundus image automatically, by removing the first large component found after scanning the image from left to right in a horizontal direction at row $m/2$.

Additionally, the vessel segmentation can be enhanced by filling any holes that appear in the connected components. Figure 2 shows a binary image before connected component manipulation, and Figure 3 after connected component manipulation.

3.6 Contour Extraction

Once we have the clean binary image of vessels, the last step in our vessel segmentation extracts the contour of the vessels from the binary image, and overlays it over the original retinal fundus image as a custom selection marquee. This means that the end-user can interactively adjust the final segmentation, by adding or removing from the selection.

To extract the vessel contours from the binary image, we make use of the Shi-Karl [9] fast level-set framework, and propagate a region based active contour that flows over pixels of intensity 0, and stops at pixels with intensity 255.

3.6.1 Shi-Karl Fast Level-Set Method

The level-set method is a numerical technique, for tracking a propagating interface which changes topology over time. It is often used in image segmentation [26], and has also been used to segment vessels [27]. A segmentation is achieved by placing a closed curve on an image, and by evolving the curve according to internal, external and user defined forces. Snakes [28] are used in a similar way to segment an image, but in the level-set method, the curve can split and change topology whereas snakes cannot. These deformable models are also sometimes called active contours.

The traditional level-set method [29] requires the calculation of partial differential equations, that govern the evolution of the curve. The level-set function is evaluated over the entire image, or over a narrow band [30], to determine where the contour is for each point in time during its evolution. The evaluation of the level-set function is a very time consuming calculation. In [9], Shi and Karl proposed a fast implementation of the level-set method, that does not require the calculation of partial differential equations. Their framework resembles the traditional level-set method, because the curve C is still represented implicitly as the zero level-set of a function ϕ. The function ϕ is defined as the signed distance function, which is positive outside C and negative inside C. From this definition, when a point on C moves inward, its neighboring point that previously was inside the curve C, will lie outside the curve, and so the value of ϕ of that neighboring point will change from negative to positive. Similarly if a point on C moves outwards, the value of ϕ of its outside neighboring point will change from positive to negative. This means that the evolution of the curve C can be controlled without solving partial differential equations, by manipulating the values of ϕ for a list of neighboring points outside C (L_{out}), and a list of neighboring points inside C (L_{in}). Formally, the two lists of neighboring points can be defined by the following two equations:

$$L_{in} = \{x|\phi(x) < 0 \ \ and \ \ \exists\, y \in N_4(x),\ \phi(y) > 0\}; \tag{11}$$

$$L_{out} = \{x|\phi(x) > 0 \ \ and \ \ \exists\, y \in N_4(x),\ \phi(y) < 0\}, \tag{12}$$

where $N_4(x)$ is the discrete 4-connected neighborhood of a pixel x.

To approximate the signed distance function, ϕ is defined as:

$$\phi(x) = \begin{cases} 3, & \text{if } x \text{ is outside } C \text{ and } x \notin L_{Out}; \\ 1, & \text{if } x \in L_{Out}; \\ -1, & \text{if } x \in L_{In}; \\ -3, & \text{if } x \text{ is inside } C \text{ and } x \notin L_{In}. \end{cases} \tag{13}$$

The curve is evolved by switching neighboring pixels between the two lists L_{in} and L_{out}, based on an external speed function F_{ext}, an internal speed function F_{int}, and by updating the level set function ϕ. The external speed is used to attract the curve to the regions of interest, while the internal speed is used to regularize the evolution of the curve so that the curve remains smooth.

In our approach the curve C is evolved according to the external speed F_{ext} only; a positive value of F_{ext} moves a point on C outwards (by switching the point from L_{out} to L_{in}), while a negative value of F_{ext} moves a point on C inwards (by switching the point from L_{in} to L_{out}).

Since we are evolving our curve on a binary image, where all noise has already been removed, there is no need to further smooth the curve according to an internal speed function F_{int}. We refer the interested reader to [31], where the use of the internal speed function F_{int} is explained in more detail.

In general, the external speed is synthesized from the image, and there are many ways to define F_{ext}. For example, the function F_{ext} could be based on the response of an edge detector applied to the image; it could be based on the mean intensities of the regions inside and outside the curve C, or as in our case, it could depend on a range of pixel intensities only.

To extract the contour of the vessels from the binary image, we define the external speed as:

$$F_{ext}(i) = \begin{cases} 1, & \text{if } i = 0; \\ -1, & \text{if } i = 255, \end{cases} \tag{14}$$

where i represents the pixel intensity, for a point on the binary image $g(x, y)$.

The evolution of the curve stops when one of the two following conditions is satisfied: *(a)* The speeds at each neighboring grid point satisfy:

$$F(x) \leq 0, \quad \forall x \in L_{out} \tag{15}$$

$$F(x) \geq 0, \quad \forall x \in L_{in}; \tag{16}$$

(b) a pre-defined maximum number of iterations N_a, is reached. The pre-defined maximum number of iterations needs to be specified for a noisy image, because the curve may fail to converge to the stable state specified in *(a)*.

3.6.2 Curve Evolution Implementation

Curve evolution using the Shi-Karl level-set framework, can be implemented by using a two dimensional array for the level-set function ϕ. The dimension of the array equals the dimension of the input image I (which itself is stored in an array). The interior list L_{in}, and the exterior list L_{out}, are represented using doubly-linked lists for efficient insertion and deletion. Using these data structures, the curve can be evolved according to algorithm 1.

Algorithm 1 Main Curve Evolution Loop

```
Initialize $\phi$, $L_{in}$ and $L_{out}$ to some closed curve according to equation 13.
repeat
    Execute algorithm 2
until equations 15 and 16 hold, or $N_a$ iterations have elapsed.
```

4 EXPERIMENTAL SETUP

Our vessel segmentation system was implemented as a series of plug-ins for ImageJ [10].

ImageJ is a public domain, Java-based image processing program developed at the National Institutes of Health. It is widely used in the medical image processing community, because of its open architecture, that provides extensibility via Java plugins and recordable macros. By developing our retinal vessel segmentation algorithms as plugins for ImageJ, we facilitate the use of our work as a part of a bigger system. Ophthalmologists will not only have at their disposal our retinal vessel segmentation algorithm, but a whole library of plugins and tools to enhance the image, and to measure and annotate vessels and pathologies.

Our retinal vessel segmentation system can be viewed as a series of procedures executed in sequence, whereby each procedure is implemented as a plug-in for ImageJ (Refer to Figure 4). Each plug-in can be run independently from the pull-down menus available in ImageJ. This allows the user to tweak certain parameters for that plug-in. By writing custom macro's, we chained each of these plug-ins together, with empirically chosen parameters, to achieve our automatic retinal vessel segmentation.

The ability to define custom macros that execute arbitrary plug-ins in an arbitrary sequence, results in a very flexible system whereby it is possible to automate additional pre-processing or post-processing, by writing other plug-ins. This means that our work can easily be extended, and used as a part of a larger system.

4.1 Segmentation Results

We consider the binary image that has been processed with connected component manipulation, to be the segmentation result of our system. To test the accuracy of our vessel segmentation, we made use of the DRIVE database [22], which has been established to facilitate comparative studies on the segmentation of blood vessels in retinal images. The database contains gold standard manual segmentations, done by an expert in the field, against which we compared our results. We tested our algorithm on all twenty images, and achieved the following results:

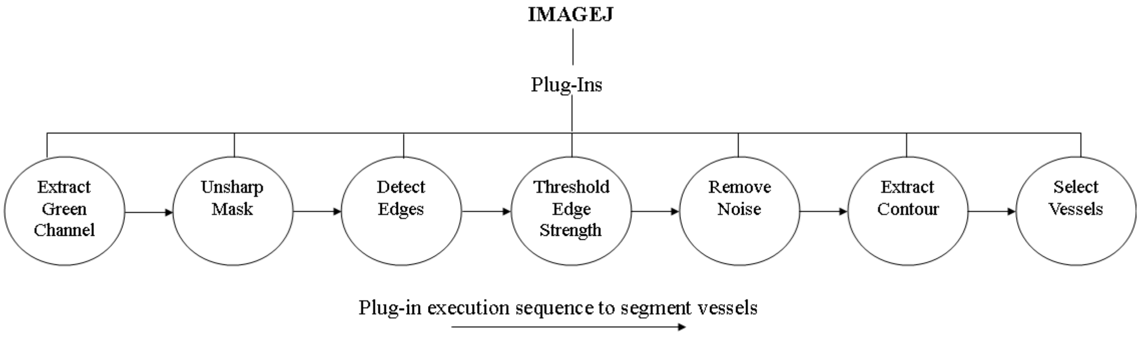


Figure 4: Plug-in execution sequence to segment retinal vessels.

Algorithm 2 Evolution according to F_{ext}

```
for $j = 0$ to $|L_{out}|$ do {$|L_{out}|$ represents the number of elements in $L_{out}$}
    $coord \leftarrow L_{out}(j)$
    $i \leftarrow g(row(coord), column(coord))$
    if $F_{ext}(i) > 0$ then
        Move a point on the curve inward by one pixel {Remove the point from $L_{in}$ and add it to $L_{out}$; update the level-set function $\phi$ accordingly, taking the 4-connected neighborhood of the point into account}.
    end if
end for
Eliminate redundant points in $L_{in}$.
for $j = 0$ to $|L_{in}|$ do {$|L_{in}|$ represents the number of elements in $L_{in}$}
    $coord \leftarrow L_{in}(j)$
    $i \leftarrow g(row(coord), column(coord))$
    if $F_{ext}(i) > 0$ then
        Move a point on the curve outward by one pixel {Remove the point from $L_{out}$ and add it to $L_{in}$; update the level-set function $\phi$ accordingly, taking the 4-connected neighborhood of the point into account}.
    end if
end for
Eliminate redundant points in $L_{out}$.
```

- The highest accuracy of our algorithm was 0.944 with a sensitivity of 0.678 and a specificity of 0.975.
- The lowest accuracy of our algorithm was 0.887 with a sensitivity of 0.755 and a specificity of 0.905.

On average only about 12.3% of pixels in one retinal image in the DRIVE database are vessels [22], so it is possible to achieve an accuracy of approximately 0.87% over the entire test-set, just by naively labelling every pixel in the retinal image as being part of the background. This means that finding a good measure to evaluate the segmentation result is difficult. The current accepted norm is to use sensitivity and specificity as we have done.

Our method compares favorably with other published algorithms displayed in Table 1, achieving the second best segmentation result of the non-supervised methods. The best segmentation result was achieved by a human observer who received limited training from an expert. The second and third best results were achieved by Staal et al [22] and Niemeijer et al [32] respectively. Both used supervised pixel classification methods, which were trained on the DRIVE training set.

Table 1: Vessel Segmentation Results

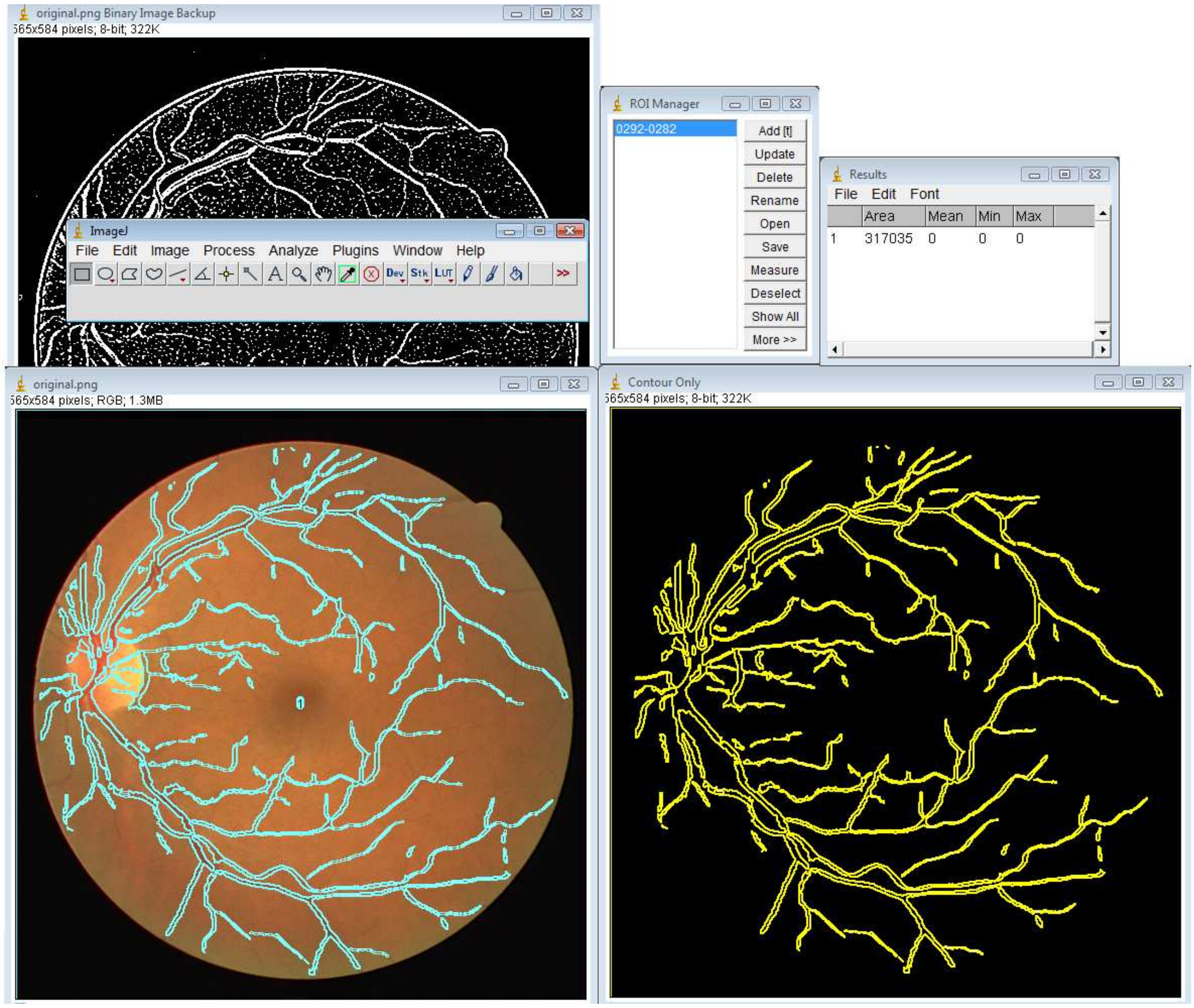
Method	**Average Accuracy**
Human Observer	0.9473
Staal [22]	0.9442
Niemeijer [32]	0.9416
Zana [33]	0.9377
Our Approach	**0.9299**
Jiang [34]	0.9212
Martinez-Perez [35]	0.9181
Chaudhuri [36]	0.8773
All Background	0.8727

Our gradient based approach fails to segment very thin vessels, where the contrast between the thin vessels and other structures is poor. Another drawback of the gradient based approach, is that by itself it cannot differentiate between vessels and pathologies, since certain pathologies have a good contrast against the background of the retinal fundus image. However, since the end-user can modify the proposed vessel segmentation with ease, very little time is spent in removing false positives. An example of our program processing an image is given in Figure 5.

5 CONCLUSION AND FUTURE WORK

We have created a retinal vessel segmentation tool for the ImageJ environment, that can propose a vessel segmentation, which can be manually altered by the end-user. By chaining together a series of plug-ins for ImageJ to achieve our result, our work can easily be extended an used as part of another system. The source code for this project is available by request or at `http://www.cs.ukzn.ac.za/cvdm`.

There are several extension that can be made to our tool. Using plug-ins, it is very easy to integrate other vessel segmentation techniques, such as those based on pixel classification and multi-scale analysis. Our future work will involve investigating methods, that can more accurately segment thin-vessels. We also intend to work more closely with ophthalmolo-


original.png Binary Image Backup
565x584 pixels; 8-bit; 322K
ImageJ
File Edit Image Process Analyze Plugins Window Help
Dev Stk LUT
ROI Manager
0292-0282
Add [t]
Update
Delete
Rename
Open
Save
Measure
Deselect
Show All
More >>
Results
File Edit Font
original.png
565x584 pixels; RGB; 1.3MB
Contour Only
565x584 pixels; 8-bit; 322K

	Area	Mean	Min	Max
1	317035	0	0	0

Figure 5: An example of our program processing a retinal fundus image. The extracted contour is overlayed over the original image.

gists, to evaluate the user-interface and to integrate features based on their feedback.

ACKNOWLEDGMENTS

This research was supported by the PRISM Grant/CSIR.

REFERENCES

[1] Sinthanayothin C., Boyce J., and Williamson C.T., "Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images.", *British Journal of Ophthalmology*, Vol. 83, pp. 902-910,1999.

[2] Jomier J., Wallace D. K., and Aylward S.R., "Quantification of retinopathy of prematurity via vessel segmentation", in *Proc. 6th MICCAI*, 2003, pp. 620-626.

[3] Chwialkowski M.P., Ibrahim Y.M., Hong F.L., and Peshock R.M., "A method for fully automated quantitative analysis of arterial flow using flow-sensitized mr images", *Comp. Med. Imaging and Graphics*, Vol. 20, pp. 365-378, 1996.

[4] Chaudhuri S., Chatterjee S., Katz N., Nelson M., and Goldbaum M., "Detection of blood vessels in retinal images using two-dimensional matched filters", *IEEE Trans. Med. Imag.*, Vol. 8, no. 3, pp. 263-269, September 1989

[5] Zana F., and Klein J.C., "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation", *IEEE Trans. Image Processing*, Vol. 10, no. 7, pp. 1010-1019, 2001.

[6] Hoover A., Kouznetsova V., and Goldbaum M., "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response", *IEEE Trans. Med. Imag.*, Vol. 19, no. 3, pp. 203-210, 2000.

[7] Lecornu L., Roux C., and Jacq J.J., "Extraction of vessel contours in angiograms by simultaneous tracking of the two edges", in *IEEE Conf. Eng. in Medicine and Bio.*, Vol. 1, pp. 678-679, 1994.

[8] McInerney T., and Terzopoulos D., "Deformable models in medical image analysis: A survey", in *IEEE Medical Image Analysis*, Vol. 1, pp. 91-108, 1996.

[9] Shi, Y., Karl, W.: A fast level set method without solving pdes. In: Proceedings of ICASSP 2005, pp. 97–100. Philadelphia, PA, USA, Mar., 2005

[10] Rasband W.S, “ImageJ”, *U. S. National Institutes of Health*, Bethesda, Maryland, USA, `http://rsb.info.nih.gov/ij/` (Accessed 1/3/2008)

[11] Kirbas C., and Quek F. K. H., “ Vessel extraction techniques and algorithms: a survey”, in *Third IEEE Symposium on Bioinformatics and Bioengineering* , pp. 238-245, 2003.

[12] Fritzsche K., Can A., Shen H., Tsai C., Turner J., Tanenbaum H., Stewart C., and Roysam B., “Automated model based segmentation, tracing and analysis of retinal vasculature from digital fundus images.”, in J. S. Suri and S. Laxminarayan, editors, *State-of-The-Art Angiography, Applications and Plaque Imaging Using MR, CT, Ultrasound and X-rays*, pp. 225-298. Academic Press, 2003.

[13] Al-Rawi M., Quteishat M., and Arar M., “An improved matched filter for blood vessel detection of digital retinal images.”, *Computer methods in biology and medicine.*, Vol. 37(2), pp. 230-237,2007.

[14] Malladi R., Sethian J., and Vemuri B., “Shape modeling with front propagation: A level set approach.”, *IEEE Trans. Pattern Analysis Machine Intelligence.*, Vol. 17(2), pp. 158-175, February 1995.

[15] Caselles V., Kimmel R., and Sapiro G., “Geodesic active contours.”, *International Journal of Computer Vision.*, Vol. 22(2), pp. 61-79, 1997.

[16] Mclnerney T., and Terzopoulos D., “Topology adaptive deformable surfaces for medical image volume segmentation.”, *IEEE Trans. Medical Imaging.*, Vol. 18(10), pp. 840-850, 1999.

[17] Lu S., and Eiho S., “Automatic detection of the coronary arterial contours with sub-branches from an x-ray angiogram.”, *IEEE Computers in Cardiology.*, pp. 575-578, 1993.

[18] Quek F., Kirbas C., and Charbel F., “Aim:attentionally-based interaction model for the interpretation of vascular angiograph.”, *IEEE Trans. on Inf. Tech. in Biomed.*, Vol. 3, pp. 139-150, June 1999.

[19] Shen H., and Johnson C.R., “Semi-automatic image segmentation: A bimodal thresholding approach.”, *Technical Report UUCS-94-019*, Un. of Utah, Dept. of

[20] Albregtsen F., “Non-Parametric Histogram Thresholding Methods Error Versus Relative Object Area”, in *Proc. Eigth Scandinavian Conf. Image Analysis*, pp. 273-280, Troms, Norway, 1993.

[21] Kittler J., and Illingworth J., “Minimum error thresholding”, in *Pattern Recognition* Vol. 19(1), pp. 41-47, January/February 1986.

[22] Staal J.J., Abramoff M.D., Niemeijer M., Viergever M.A., and van Ginneken B., “Ridge based vessel segmentation in color images of the retina”, in *IEEE Transactions on Medical Imaging*,Vol. 23 pp. 501-509, 2004. Comp. Science, 1994.

[23] Cornforth D.J., Jelinek H.J., Leandro J.J.G., Soares J.V.B., Cesar Jr. R. M., Cree M.J., Mitchell P., and Bossomaier T., “Development of retinal blood vessel segmentation methodology using wavelet transforms assesment of diabetic retinopathy”, in *The 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 50-60, 2004.

[24] Sahoo P.K., Soltani S., Wong K.C., and Chen Y.C., “A Survey of Thresholding Techniques”, in *Computer Vision, Graphics, and Image Processing*,Vol. 41, pp. 233-260, 1988.

[25] Otsu N., “A threshold selection method from gray level histograms”, in *IEEE Trans. Systems, Man and Cybernetics*,Vol. 9, pp. 62-66, 1979.

[26] Paragios, N., Deriche, R.: Coupled geodesic active regions for image segmentation: A level set approach. In: Proceedings ECCV, pp. 224-240. Dublin, 2000

[27] Nain D., Yezzi A., and Turk G., “Vessel Segmentation Using a Shape Driven Flow”, in *Lecture Notes in Computer Science*, Issu. 3216, pp. 51-59, 2004.

[28] Kass, M., Witkin, A., Terzopoulos, D.: Snakes - active contour models. In: International Journal of Computer Vision 1(4), pp. 321-331. , 1987

[29] Sethian, J.,: Level Set Methods and Fast Marching Methods. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, (1999).

[30] Adalsteinsson, D., Sethian, J.: A fast level set method for propagating interfaces. In: Journal Computational Physics, 118(2), pp. 269-277, 1995

[31] Shi Y., and Karl W. C., “A fast implementation of the level set method without solving partial differential equations”, *Technical Report No. ECE-2005-02*, Department of Electrical and Computer Engineering, Boston University, 2005

[32] Niemeijer M., Staal J.J., van Ginneken B., and Loog M., “Comparitive study of retinal vessel segmentation methods on a new publicly available database”, in *SPIE Medical Imaging*, Editor(s) Fitzpatrick J.M. and Sonka M., Vol. 5370, pp. 648-656, 2004.

[33] Zana F., and Klein J., “Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation”, in *IEEE Transactions on Image Processing* ,Vol. 10, pp. 1010-1019, 2001.

[34] Jiang X., and Mojon D., “Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*,Vol. 25, pp. 131-137, 2003.

[35] Martinez-Perez M., Hughes A., Stanton A., Thom S., Bharath A., and Parker K., “Scale-space analysis for the characterisation of retinal blood vessels”, in *Medical Image Computing and Computer-Assisted Intervention - MICCAI99*, Editor(s) Taylor C. and Colchester A., pp. 90-97, 1999.

[36] Chaudhuri S., Chatterjee S., Katz N., Nelson M., and Goldbaum M., “Detection of blood vessels in retinal images using two-dimensional matched filters”, in *IEEE Transactions on Medical Imaging*, Vol. 8, pp. 263-269, 1989.